Content index

Request and Response ... 11
Request types ... 12
Jsonrpc ... 12
Xmlrpc ... 12
Simple ... 12
Response types ... 12
Test ... 13
Jsonrpc ... 13
Xmlrpc ... 13
Xml ... 13
Json ... 13
Debug ... 13
Simple ... 14
Errors ... 15
Error codes ... 16
SUCCESS (code: 1000) ... 16
SUCCESS_PENDING (code: 1001) ... 16
UNKNOWN_COMMAND (code: 2000) ... 16
COMMAND_SYNTAX_ERROR (code: 2001) ... 16
COMMAND_USE_ERROR (code: 2002) ... 16
PARAM_MISSING (code: 2003) ... 16
PARAM_VALUE_RANGE (code: 2004) ... 16
PARAM_VALUE_SYNTAX (code: 2005) ... 16
DELETED_COMMAND (code: 2100) ... 17
UNIMPLEMENTED_COMMAND (code: 2101) ... 17
BILLING_FAILURE (code: 2104) ... 17
BILLING_NOT_ALLOWED (code: 2105) ... 17
AUTH_ERROR_USER (code: 2200) ... 17
AUTH_ERROR_OBJECT (code: 2201) ... 17
OBJECT_EXISTS (code: 2302) ... 17
OBJECT_NOT_EXISTS (code: 2303) ... 18
OBJECT_STATUS_PROHIBITS_OP (code: 2304) ... 18
OBJECT_ASSOCIATION_PROHIBITS_OP (code: 2305) ... 19
PARAMETER_VALUE_POLICY_ERROR (code: 2306) ... 19
UNIMPLEMENTED_OBJECT_SERVICE (code: 2307) ... 19
COMMAND_FAILED (code: 2400) ... 20
COMMAND_FAILED_FATAL (code: 2500) ... 20
COMMAND_TIMEOUT (code: 2501) ... 20
COMMAND_CONNECTIONSLIMIT (code: 2502) ... 20
COMMAND_MAINTENANCE_MODE (code: 2503) ... 21
COMMAND_PENDING_PREVIOUS_EXECUTION (code: 3101) ... 21
COMMAND_FTP_MAX_ACCOUNTS (code: 3601) ... 21
COMMAND_EMAIL_REDIRECTION_MAX_ACCOUNTS (code: 3602) ... 21
COMMAND_EMAIL_DISTRIBUTIONLIST_MAX (code: 3603) ... 21
COMMAND_SERVER_KVM_GET_ERROR (code: 3604) ... 21
COMMAND_DEDICATED_SERVER_OS_ERROR (code: 3605) ... 21
COMMAND_LICENSE_MAX_HOSTINGS (code: 3606) ... 21
COMMAND_BILLING_PRODUCT_ERROR (code: 3607) ... 22
COMMAND_BILLING_CREDIT (code: 3608) ... 22
COMMAND_EMAIL_MAX_HOSTINGS (code: 3609) ... 22
COMMAND_DOMAIN_FORBIDDEN_CONTACT_UPDATE (code: 3610) ... 22
COMMAND_DDBB_SQLSERVER_NOT_ENOUGH_SPACE (code: 3612) ... 22
COMMAND_EMAIL_DISTRIBUTION_LIST_EXISTS (code: 3613) ... 22
COMMAND_SMS_NOT_REMAINING_MESSAGES (code: 3614) ... 23
COMMAND_MYSQL_MAX_DATABASES (code: 3615) ... 23
COMMAND_DOMAIN_IP_NOT_RESOLVING_DNS (code: 3616) ... 23
COMMAND_EMAIL_NOT_ENABLED (code: 3617) ... 23
COMMAND_REGENERATE_PASSWORD_SEND_METHOD (code: 3618) ... 23

COMMAND_EMAIL_ACCOUNT_SET_REPEATED_PASSWORD (code: 3619) ... 23
COMMAND_BILLING_DOMAIN_FORBIDDEN_PREDEFINED_CONTACTS (code: 3620) ... 23
COMMAND_DOMAIN_TRANSFER_ALREADY_APPROVED (code: 3630) ... 23
COMMAND_DOMAIN_CERT_ALREADY_ASSOCIATED (code: 3631) ... 23
COMMAND_BILLING_PRODUCT_TEMPORARILY_DISABLED (code: 3621) ... 24
COMMAND_NFS_VOLUME_DOES_NOT_EXIST (code: 3640) ... 24
COMMAND_PARAM_IS_NOT_A_VALID_CERTIFICATE (code: 3641) ... 24
COMMAND_DOMAIN_SERVER_ZONE_NOT_FOUND (code: 3645) ... 24
COMMAND_ADD_ZONES_LIMIT (code: 3646) ... 24
Commands ... 25
Billing commands ... 26
Billing_Transfer_Domain ... 26
Billing_Transfer_GetRequirements ... 26
Billing_Renew_Multihosting ... 26
Billing_Renew_Domain ... 27
Billing_Renew_Sms ... 27
Billing_Renew_Vps ... 27
Billing_Renew_Hosting ... 28
Billing_Buy_Multihosting ... 28
Billing_Buy_Domain ... 28
Billing_Buy_Sms ... 29
Billing_Buy_Vps_Unmanaged_Windows ... 29
Billing_Buy_Vps_Unmanaged_Linux ... 30
Billing_Buy_Vps_Managed_Windows ... 30
Billing_Buy_Vps_Managed_Linux ... 31
Billing_Buy_DomainUsingContactsIds ... 31
Billing_Buy_Hosting ... 32
Billing_Price_Multihosting ... 32
Billing_Price_Domain ... 32
Billing_Price_Sms ... 33
Billing_Price_Vps_Unmanaged_Windows ... 33
Billing_Price_Vps_Unmanaged_Linux ... 34
Billing_Price_Vps_Managed_Windows ... 34
Billing_Price_Vps_Managed_Linux ... 34
Billing_Price_Hosting ... 35
Billing_Autorenew_SetOff ... 35
Billing_Autorenew_SetOn ... 36
Billing_Autorenew_SetNoRenewable ... 36
Billing_Autorenew_GetAll ... 36
Billing_GetConfigurableVpsOptions ... 37
Billing_Help_DomainExtraData ... 37
Billing_Period_Multihosting ... 37
Billing_Period_Domain ... 38
Billing_Period_Vps ... 38
Billing_Period_Hosting ... 38
Billing_NoRenewable_SetRenewable ... 39
Billing_NoRenewable_GetAll ... 39
Dedicated commands ... 40
Dedicated_Services_ChangeStatus ... 40
Dedicated_Services_GetAll ... 40
Dedicated_ForVps_GetServerInfo ... 40
Dedicated_ForVps_Vps_Delete ... 41
Dedicated_ForVps_Vps_Update ... 41
Dedicated_ForVps_Vps_Activate ... 41
Dedicated_ForVps_Vps_Deactivate ... 42
Dedicated_ForVps_Vps_Create_Unmanaged ... 42
Dedicated_ForVps_Vps_Create_Managed ... 42
Dedicated_ForVps_Vps_GetTypes ... 43
Dedicated_ForVps_Vps_GetAll ... 43

Dedicated_ForVps_GetTicketStatus ... 43
Dedicated_Nfs_Get ... 44
Dedicated_MailReport_Get ... 44
Dedicated_MailReport_SetOff ... 44
Dedicated_MailReport_Set ... 45
Dedicated_GetDataTransferInfo ... 45
Dedicated_GetUsedSpace ... 45
Dedicated_GetServerInfo ... 46
Dedicated_GetBandwidthConsumption ... 46
Dedicated_HostingSetup_Delete ... 46
Dedicated_HostingSetup_Get ... 47
Dedicated_HostingSetup_Update ... 47
Dedicated_HostingSetup_GetCustomSetups ... 47
Dedicated_HostingSetup_GetDhSetups ... 47
Dedicated_HostingSetup_GetOptions ... 48
Dedicated_HostingSetup_Create ... 48
Dedicated_HostingSetup_GetAll ... 48
Dedicated_GetSpaceAndTransferReportPerHosting ... 49
Dedicated_Monitorization_Get ... 49
Dedicated_Monitorization_Activate ... 49
Dedicated_Monitorization_Deactivate ... 49
Dedicated_GetTransferValue ... 50
Dedicated_Kvm_Get ... 50
Dedicated_Kvm_Unmount ... 50
Dedicated_Kvm_Mount ... 51
Dedicated_Control_GetCpuLoadByHosting ... 51
Dedicated_Control_GetMysqlProcesses ... 51
Dedicated_Control_GetWebserverStatus ... 52
Dedicated_Control_GetSystemStatus ... 52
Dedicated_Control_KillMysqlProcess ... 52
Dedicated_Control_GetApacheProcesses ... 53
Dedicated_Control_GetSystemLoad ... 53
Dedicated_PowerStatus_Restart ... 53
Dedicated_GetLicenses ... 53
Dedicated_Hosting_Delete ... 54
Dedicated_Hosting_ChangeSetup ... 54
Dedicated_Hosting_Activate ... 54
Dedicated_Hosting_GetSetupName ... 55
Dedicated_Hosting_Deactivate ... 55
Dedicated_Hosting_GetSpace ... 55
Dedicated_Hosting_Create ... 56
Dedicated_Hosting_GetAll ... 56
Dedicated_GetMonitorizedPorts ... 56
Domain commands ... 58
Domain_CheckForTransfer ... 58
Domain_Transfer_SetAuthcode ... 58
Domain_Transfer_Restart ... 58
Domain_Transfer_Cancel ... 58
Domain_Transfer_GetAuthcode ... 59
Domain_Transfer_GetStatus ... 59
Domain_Test ... 59
Domain_CheckForRegister ... 59
Domain_Zone_GetTypeURL301 ... 60
Domain_Zone_UpdateTypeA ... 60
Domain_Zone_Delete ... 60
Domain_Zone_AddTypeSRV ... 61
Domain_Zone_AddTypeFrame ... 61
Domain_Zone_AddTypeAAAA ... 61
Domain_Zone_DeleteTypeA ... 62

Domain_Zone_DeleteTypeSRV ... 62
Domain_Zone_DeleteTypeURL301 ... 62
Domain_Zone_DeleteTypeFrame ... 63
Domain_Zone_AddTypeURL301 ... 63
Domain_Zone_GetTypeFrame ... 63
Domain_Zone_DeleteTypeURL ... 64
Domain_Zone_DeleteTypeMX ... 64
Domain_Zone_AddTypeURL ... 64
Domain_Zone_DeleteTypeTXT ... 65
Domain_Zone_AddTypeMXDetailedEmail ... 65
Domain_Zone_DeleteTypeAname ... 65
Domain_Zone_AddTypeMXS ... 65
Domain_Zone_AddTypeTXT ... 66
Domain_Zone_GetTypeURL ... 66
Domain_Zone_UpdateTypeAname ... 66
Domain_Zone_DeleteTypeCname ... 67
Domain_Zone_GetTypeAname ... 67
Domain_Zone_AddTypeMX ... 67
Domain_Zone_GetTypeSRV ... 68
Domain_Zone_UrlShortener_Delete ... 68
Domain_Zone_UrlShortener_Activate ... 68
Domain_Zone_UrlShortener_Deactivate ... 69
Domain_Zone_UrlShortener_Create ... 69
Domain_Zone_UrlShortener_GetAll ... 69
Domain_Zone_UpdateTypeAAAA ... 69
Domain_Zone_AddTypeCname ... 70
Domain_Zone_GetTypeA ... 70
Domain_Zone_GetTypeTXT ... 70
Domain_Zone_DeleteTypeAAAA ... 71
Domain_Zone_GetTypeAAAA ... 71
Domain_Zone_AddTypeAname ... 71
Domain_Zone_GetTypeMX ... 71
Domain_Zone_DeleteTypeSPF ... 72
Domain_Zone_GetTypeSPF ... 72
Domain_Zone_DeleteAll ... 72
Domain_Zone_GetTypeCname ... 73
Domain_Zone_AddTypeA ... 73
Domain_Zone_GetAll ... 73
Domain_GetRegistrationDate ... 74
Domain_Status_Get ... 74
Domain_Status_IsLocked ... 74
Domain_Status_Unlock ... 74
Domain_Status_Lock ... 75
Domain_Redirection_Delete ... 75
Domain_Redirection_Get ... 75
Domain_Redirection_SetStd ... 76
Domain_Redirection_SetFrameSimple ... 76
Domain_Redirection_SetFrame ... 76
Domain_Redirection_Set301 ... 77
Domain_Redirection_GetAll ... 77
Domain_PanelIndependentAccess_SetPassword ... 77
Domain_Contacts_Get ... 77
Domain_Contacts_GetRegistrantVerificationInfo ... 78
Domain_Contacts_SetAll ... 78
Domain_Contacts_Set ... 78
Domain_Contacts_GetAll ... 79
Domain_GetRenovationDate ... 79
Domain_GetExpirationDate ... 79
Domain_Dnss_Get ... 79

Domain_Dnss_Set ... 80
Domain_UrlShortener_Delete ... 80
Domain_UrlShortener_Activate ... 80
Domain_UrlShortener_Deactivate ... 80
Domain_UrlShortener_Create ... 81
Domain_UrlShortener_GetAll ... 81
Domain_GetAuthcode ... 81
Domain_GetInfoDomain ... 82
Domain_WhoisPrivacy_Get ... 82
Domain_WhoisPrivacy_Set ... 82
Domain_Parking_Get ... 82
Domain_Parking_Set ... 83
Domain_NameServer_Delete ... 83
Domain_NameServer_Register ... 83
Domain_NameServer_Modify ... 84
Domain_NameServer_GetAll ... 84
DomainVerification commands ... 85
DomainVerification_GetPending ... 85
Hosting commands ... 86
Hosting_Test ... 86
Hosting_Ddbb_MariaDb_Delete ... 86
Hosting_Ddbb_MariaDb_SetPassword ... 86
Hosting_Ddbb_MariaDb_Create ... 86
Hosting_Ddbb_MariaDb_GetAll ... 87
Hosting_Ddbb_Postgre_Delete ... 87
Hosting_Ddbb_Postgre_SetPassword ... 87
Hosting_Ddbb_Postgre_Create ... 88
Hosting_Ddbb_Postgre_GetAll ... 88
Hosting_Ddbb_Mysql_SetHostAcl ... 88
Hosting_Ddbb_Mysql_Delete ... 89
Hosting_Ddbb_Mysql_GetAclConfiguration ... 89
Hosting_Ddbb_Mysql_CreateWithIpAcl ... 89
Hosting_Ddbb_Mysql_SetPassword ... 90
Hosting_Ddbb_Mysql_BackupTask_Delete ... 90
Hosting_Ddbb_Mysql_BackupTask_Set ... 90
Hosting_Ddbb_Mysql_BackupTask_GetAll ... 91
Hosting_Ddbb_Mysql_GetInfo ... 91
Hosting_Ddbb_Mysql_SetIpAcl ... 91
Hosting_Ddbb_Mysql_Backup_Get ... 92
Hosting_Ddbb_Mysql_Backup_Restore ... 92
Hosting_Ddbb_Mysql_Create ... 92
Hosting_Ddbb_Mysql_GetAll ... 93
Hosting_Ddbb_Access_GetAllMdb ... 93
Hosting_Ddbb_Sqlserver_Delete ... 93
Hosting_Ddbb_Sqlserver_GetSpaceInfo ... 94
Hosting_Ddbb_Sqlserver_SetPassword ... 94
Hosting_Ddbb_Sqlserver_SetTransactionsSize ... 94
Hosting_Ddbb_Sqlserver_GetGlobalSpaceInfo ... 94
Hosting_Ddbb_Sqlserver_SetDataSize ... 95
Hosting_Ddbb_Sqlserver_Create ... 95
Hosting_Ddbb_Sqlserver_GetAll ... 96
Hosting_Ddbb_MongoDb_Delete ... 96
Hosting_Ddbb_MongoDb_SetPassword ... 96
Hosting_Ddbb_MongoDb_GetAll ... 97
Hosting_Ddbb_GetAll ... 97
Hosting_Import_ListAll ... 97
Hosting_Import_List ... 97
Hosting_Import_ApplyForEmailAccountImap ... 98
Hosting_Import_ApplyForFtp ... 98

Hosting_Import_ApplyForDdbb 99
Hosting_Import_ApplyForEmailAccountPop 100
Hosting_Dsn_Mysql_Delete 100
Hosting_Dsn_Mysql_Create 101
Hosting_Dsn_Mysql_GetAll 101
Hosting_Dsn_Access_Delete 101
Hosting_Dsn_Access_Create 102
Hosting_Dsn_Access_GetAll 102
Hosting_Dsn_Sqlserver_Delete 102
Hosting_Dsn_Sqlserver_Create 103
Hosting_Dsn_Sqlserver_GetAll 103
Hosting_Https_Get 103
Hosting_Https_Activate 103
Hosting_Https_Deactivate 104
Hosting_Subdomain_Delete 104
Hosting_Subdomain_Create 104
Hosting_Subdomain_GetAll 105
Hosting_Ssh_Get 105
Hosting_Ssh_Activate 105
Hosting_Ssh_Deactivate 105
Hosting_IsActive 106
Hosting_PanelIndependentAccess_SetPassword 106
Hosting_GetInfo 106
Hosting_GetMailInfo 106
Hosting_Backup_Send_Database_Tables 107
Hosting_Backup_Send_Database 107
Hosting_Backup_Send_Resource 108
Hosting_Backup_Database_GetTables 108
Hosting_Backup_Restore_Database_Tables 108
Hosting_Backup_Restore_EmailAccounts 109
Hosting_Backup_Restore_Database 109
Hosting_Backup_Restore_Resource 109
Hosting_Backup_GetEmailAccounts 110
Hosting_Backup_GetDatabases 110
Hosting_Backup_GetResources 110
Hosting_Backup_GetStatus 111
Hosting_UrlShortener_Delete 111
Hosting_UrlShortener_Activate 111
Hosting_UrlShortener_Deactivate 111
Hosting_UrlShortener_Create 112
Hosting_UrlShortener_GetAll 112
Hosting_Email_Autoresponse_Delete 112
Hosting_Email_Autoresponse_Get 113
Hosting_Email_Autoresponse_Activate 113
Hosting_Email_Autoresponse_Deactivate 113
Hosting_Email_Autoresponse_Create 114
Hosting_Email_Redirection_Delete 114
Hosting_Email_Redirection_DeactivateCatchAll 114
Hosting_Email_Redirection_Get 115
Hosting_Email_Redirection_Update 115
Hosting_Email_Redirection_ActivateCatchAll 115
Hosting_Email_Redirection_Create 115
Hosting_Email_Redirection_GetAll 116
Hosting_Email_UsedSpace_Get 116
Hosting_Email_UsedSpace_GetPerAlias 116
Hosting_Email_AutomaticDeletion_CreateRule 117
Hosting_Email_AutomaticDeletion_DeleteRule 117
Hosting_Email_Filter_CreateForSpam 117
Hosting_Email_Filter_Delete 118

Hosting_Email_Filter_GetGeolocalization ... 118
Hosting_Email_Filter_SetGeolocalization ... 118
Hosting_Email_Filter_Create ... 118
Hosting_Email_Filter_GetAll ... 119
Hosting_Email_Dkim_Get ... 119
Hosting_Email_Dkim_Activate ... 119
Hosting_Email_Dkim_IsActive ... 120
Hosting_Email_Dkim_Deactivate ... 120
Hosting_Email_Antispam_AddDomainToBlacklist ... 120
Hosting_Email_Antispam_RemoveDomainFromBlacklist ... 120
Hosting_Email_Antispam_Activate ... 121
Hosting_Email_Antispam_SetConfig ... 121
Hosting_Email_Antispam_Deactivate ... 121
Hosting_Email_Antispam_GetScore ... 121
Hosting_Email_Antispam_RemoveDomainFromWhitelist ... 122
Hosting_Email_Antispam_GetBlacklist ... 122
Hosting_Email_Antispam_GetConfig ... 122
Hosting_Email_Antispam_AddDomainToWhitelist ... 123
Hosting_Email_Antispam_SetScore ... 123
Hosting_Email_Antispam_GetWhitelist ... 123
Hosting_Email_Account_Delete ... 124
Hosting_Email_Account_Get ... 124
Hosting_Email_Account_SetCopyTo ... 124
Hosting_Email_Account_SetPassword ... 125
Hosting_Email_Account_Empty ... 125
Hosting_Email_Account_Activate ... 125
Hosting_Email_Account_GetCopyTo ... 126
Hosting_Email_Account_RemoveCopyTo ... 126
Hosting_Email_Account_Deactivate ... 126
Hosting_Email_Account_Create ... 126
Hosting_Email_Account_GetAll ... 127
Hosting_Email_Antivirus_GetAttachedFilesFilters ... 127
Hosting_Email_Antivirus_SetConfig ... 127
Hosting_Email_Antivirus_GetConfig ... 128
Hosting_Email_Antivirus_UpdateAttachedFileFilter ... 128
Hosting_Ftp_Get ... 128
Hosting_Ftp_GetAccountsLimit ... 129
Hosting_Ftp_Activate ... 129
Hosting_Ftp_Deactivate ... 129
Hosting_Ftp_Account_Delete ... 129
Hosting_Ftp_Account_SetPassword ... 130
Hosting_Ftp_Account_CmdStatus_GetSetPassword ... 130
Hosting_Ftp_Account_CmdStatus_ListAll ... 130
Hosting_Ftp_Account_CmdStatus_GetDelete ... 131
Hosting_Ftp_Account_CmdStatus_GetCreate ... 131
Hosting_Ftp_Account_Create ... 131
Hosting_Ftp_Account_GetAll ... 132
Hosting_Certificate_Uninstall ... 132
Hosting_Certificate_ActivateLetsEncrypt ... 132
Hosting_Certificate_DeactivateLetsEncrypt ... 133
Hosting_Certificate_Install ... 133
Hosting_Php_PhpExecutionOnHtmlIsActive ... 133
Hosting_Php_ActivatePhpExecutionOnHtml ... 133
Hosting_Php_DeactivatePhpExecutionOnHtml ... 134
Hosting_Php_SetVersion ... 134
Hosting_Php_OverwritePhpIniWithDefaults ... 134
Hosting_Php_OverwritePhpIni ... 134
Hosting_GetDaysWithBackups ... 135
Hosting_TlsMail_Get ... 135

Hosting_TlsMail_Activate ... 135
Hosting_TlsMail_Deactivate ... 135
Housing commands ... 137
Housing_GetDataTransferInfo ... 137
Housing_GetServerInfo ... 137
Housing_GetBandwidthConsumption ... 137
Housing_Monitorization_Get ... 137
Housing_Monitorization_Activate ... 138
Housing_Monitorization_Deactivate ... 138
Housing_GetTransferValue ... 138
Housing_Kvm_Get ... 139
Housing_Kvm_Unmount ... 139
Housing_Kvm_Mount ... 139
Housing_PowerStatus_Restart ... 139
Import commands ... 141
Import_GetStatus ... 141
Multihosting commands ... 142
Multihosting_MailReport_Get ... 142
Multihosting_MailReport_SetOff ... 142
Multihosting_MailReport_Set ... 142
Multihosting_GetEmailAccountsNumber ... 142
Multihosting_GetUsedSpace ... 143
Multihosting_HostingSetup_Delete ... 143
Multihosting_HostingSetup_Get ... 143
Multihosting_HostingSetup_Update ... 144
Multihosting_HostingSetup_GetOptions ... 144
Multihosting_HostingSetup_Create ... 144
Multihosting_HostingSetup_GetAll ... 145
Multihosting_GetInfo ... 145
Multihosting_GetUsedTransfer ... 145
Multihosting_GetLicenses ... 146
Multihosting_Hosting_Delete ... 146
Multihosting_Hosting_ChangeSetup ... 146
Multihosting_Hosting_Activate ... 147
Multihosting_Hosting_Deactivate ... 147
Multihosting_Hosting_GetSpace ... 147
Multihosting_Hosting_Create ... 147
Multihosting_Hosting_GetAll ... 148
Payment commands ... 149
Payment_Method_GetAll ... 149
Payment_Credit_List ... 149
Payment_Credit_GetDetailed ... 149
Payment_Credit_GetTotal ... 149
Reseller commands ... 151
Reseller_AssociateServices ... 151
Reseller_DissociateServices ... 151
Services commands ... 152
Services_GetUnmanagedCloudServers ... 152
Services_GetUnmanagedVps ... 152
Services_GetUnmanagedDedicatedServers ... 152
Services_GetManagedCloudServers ... 152
Services_GetSms ... 153
Services_GetDomainTransfers ... 153
Services_GetHousingServers ... 153
Services_GetMultihostings ... 154
Services_GetVps ... 154
Services_GetDomains ... 154
Services_GetManagedVps ... 154
Services_GetCloudServers ... 155

Services_GetAllManagedServersNames ... 155
Services_GetManagedDedicatedServers ... 155
Services_GetDedicatedServers ... 156
Services_GetHostings ... 156
Sms commands ... 157
Sms_Send_Bulk_Long_Gsm7 ... 157
Sms_Send_Bulk_Long_Unicode ... 157
Sms_Send_Bulk_Limited_Gsm7 ... 157
Sms_Send_Bulk_Limited_Unicode ... 158
Sms_Phonebook_UpdateContact ... 158
Sms_Phonebook_DeleteContact ... 159
Sms_Phonebook_GetAllContacts ... 159
Sms_Phonebook_CreateContact ... 159
Sms_PanelIndependentAccess_SetPassword ... 160
Sms_DeliveryList_Delete ... 160
Sms_DeliveryList_Get ... 160
Sms_DeliveryList_GetPhones ... 160
Sms_DeliveryList_Update ... 161
Sms_DeliveryList_Create ... 161
Sms_DeliveryList_GetAll ... 161
Sms_Programmed_Cancel ... 162
Sms_GetCredit ... 162
Sms_History_GetSent ... 162
Sms_History_GetBulkSent ... 162
Sms_History_GetProgrammed ... 163
Sms_SetUp ... 163
Sms_GetSender ... 163
System commands ... 165
System_GetApiVersion ... 165
System_GetRequestTypes ... 165
System_TestSimulation ... 165
System_ListMethods ... 165
System_GetResponseTypes ... 165
System_MethodSignature ... 166
System_MethodHelp ... 166
User commands ... 167
User_GetServices ... 167
User_GetInvoicedAmount ... 167
User_GetContactVerificationStatus ... 167
User_Secondary_Delete ... 168
User_Secondary_Update ... 168
User_Secondary_Create ... 168
User_Secondary_GetAll ... 169
User_GetInfo ... 169
User_GetNotifications ... 169
User_SendContactVerification ... 169
User_GetLastLoginDate ... 170
User_Data_GetValidLegalForms ... 170
User_Default_GetTechContact ... 170
User_Default_GetAdminContact ... 171
User_Default_GetDnss ... 171
Vps commands ... 172
Vps_Services_ChangeStatus ... 172
Vps_Services_GetAll ... 172
Vps_Nfs_Get ... 172
Vps_MailReport_Get ... 173
Vps_MailReport_SetOff ... 173
Vps_MailReport_Set ... 173
Vps_GetDataTransferInfo ... 173

Vps_GetUsedSpace ... 174
Vps_GetServerInfo ... 174
Vps_GetBandwidthConsumption ... 174
Vps_HostingSetup_Delete ... 175
Vps_HostingSetup_Get ... 175
Vps_HostingSetup_Update ... 175
Vps_HostingSetup_GetOptions ... 176
Vps_HostingSetup_Create ... 176
Vps_HostingSetup_GetAll ... 176
Vps_GetSpaceAndTransferReportPerHosting ... 177
Vps_Monitorization_Get ... 177
Vps_Monitorization_Activate ... 177
Vps_Monitorization_Deactivate ... 177
Vps_GetAllHostingsFullReport ... 178
Vps_Control_GetCpuLoadByHosting ... 178
Vps_Control_GetMysqlProcesses ... 178
Vps_Control_GetSystemStatus ... 179
Vps_Control_KillMysqlProcess ... 179
Vps_Control_GetApacheProcesses ... 179
Vps_Control_GetSystemLoad ... 180
Vps_PowerStatus_Restart ... 180
Vps_PowerStatus_Get ... 180
Vps_PowerStatus_TurnOn ... 180
Vps_PowerStatus_Suspend ... 181
Vps_PowerStatus_Reactivate ... 181
Vps_PowerStatus_TurnOff ... 181
Vps_PowerStatus_ElectricRestart ... 181
Vps_PowerStatus_Unpause ... 182
Vps_PowerStatus_Pause ... 182
Vps_GetLicenses ... 182
Vps_Hosting_Delete ... 183
Vps_Hosting_ChangeSetup ... 183
Vps_Hosting_Activate ... 183
Vps_Hosting_GetSetupName ... 183
Vps_Hosting_Deactivate ... 184
Vps_Hosting_Create ... 184
Vps_Hosting_GetAll ... 184
Vps_GetMonitorizedPorts ... 185

Request and Response

Request types

Jsonrpc

Json Remote Procedure Call request.

Special HTTP request that sends JSON encoded arguments and expects a JSON encoded response.

Characteristics and requeriments:
- Request data sent through HTTP POST query .
- Content-type indicated as "application/json" on HTTP headers.
- User login data must be sent at "Authorization: Basic" on HTTP headers .
- Parameters can be sent in any order. Keys are case sensitive.
- Only used with Jsonrpc response type.
- Simulation mode only available on web server at https://dinahosting.com/api/.

Xmlrpc

Standard Xml Remote Procedure Call request.

Special HTTP request that sends XML encoded arguments and expects a XML encoded response.

Characteristics and requeriments:
- Request data sent through HTTP POST query .
- Content-type indicated as "application/xml" on HTTP headers.
- User login data must be sent at "Authorization: Basic" on HTTP headers .
- Parameters must arrive ordered, as described in method signature.
- Only used with Xmlrpc response type.
- Simulation mode only available on web server at https://dinahosting.com/api/ .

Simple

Simple HTTP REQUEST .

Standard HTTP request that sends key=value pairs in a URL encoded request.

Characteristics and requeriments:
- Request data sent through a standard HTTP GET or POST query .
- User login authentication data might be sent at "Authorization: Basic" on HTTP headers or in request parameters AUTH_USER and AUTH_PWD.
- Parameters can be sent in any order. Keys are case sensitive.
- Can be used with any responseType.
- Can force client-side activation of simulation mode. Just add a boolean parameter SIMULATE=true to the request.

Example: https://dinahosting.com/special/api.php?AUTH_USER=yourUser&AUTH_PWD=yourPass&command=System_GetRequestTypes&responseType=Xml&SIMULATE=false

Response types

Test

Testing response. Returns valid PHP code.
Harder to parse than Json or XML.

Used with standard (Simple) HTTP requests.

Jsonrpc

JavaScript Object Notation Remote procedure call response.

Same as standard Json .

Only for JsonRpc requests.

Xmlrpc

Extensible markup language Remote procedure call response.

Standard XML response built with Zend XmlRpc server.
Easy to parse with Zend XmlRpc client, but heavier communication.

Recommended only for XmlRpc requests.

Xml

Extensible markup language response.

Multi-plataform language. Lighter than our XmlRpc.

Used with standard (Simple) HTTP requests.

Json

JavaScript Object Notation response.

Lightweight, multi-plataform, easy parseable response type.

Used with standard (Simple) HTTP requests.

Debug

Debuging response.

Used with standard (Simple) HTTP requests.

Simple

Simple key value plain text response.

Lightest response, but it isn't standard.
Parsing may be troublesome if response data is complex.
Useful for testing purposes.

Used with standard (Simple) HTTP requests.

Errors

Error codes

SUCCESS (code: 1000)

Command execution ended OK and returned data if any.

Error message: *Success.*
Can be throwed by commands: All existing commands might return this error.

SUCCESS_PENDING (code: 1001)

Command execution ended OK but additional background action is pending.

Error message: *Success. Action is pending.*
Can be throwed by commands: None

UNKNOWN_COMMAND (code: 2000)

The user requested a command that doesn't exists.

Error message: *Unknown command.*
Can be throwed by commands: Thrown if requested command doesn't exists.

COMMAND_SYNTAX_ERROR (code: 2001)

There is at least one error at command arguments.
A parameter may be missing or has got a bad format.
See aditional information at errors structure.

Error message: *Command syntax error.*
Can be throwed by commands: All existing commands might return this error.

COMMAND_USE_ERROR (code: 2002)

The user doesn't indicated a command on request, requested an invalid response type, or tried to use a command for something that isn't meant to.

Error message: *Command use error.*
Can be throwed by commands: All existing commands might return this error.

PARAM_MISSING (code: 2003)

The command requires a parameter that is missing on the request.

Error message: *Required param "%s" is missing.*
Can be throwed by commands: All existing commands might return this error.

PARAM_VALUE_RANGE (code: 2004)

A command parameter value lenght isn't valid. The value lenght might be too long or too short.

Error message: *Param "%s" length is not valid.*
Can be throwed by commands: All existing commands might return this error.

PARAM_VALUE_SYNTAX (code: 2005)

A command parameter value format isn't valid.

Error message: *Param "%s" value syntax is not valid.*
Can be throwed by commands: All existing commands might return this error.

DELETED_COMMAND (code: 2100)

The user requested a command that existed but it isn't available anymore,
as it was deprecated and deleted.

Error message: *Command was deprecated and itsn't available anymore.*
Can be throwed by commands: None

UNIMPLEMENTED_COMMAND (code: 2101)

The user requested a command that exists but it isn't implemented,
or tried to do simulation or MethodHelp over a command that doesn't implements it.

Error message: *Action %s currently cant be performed on this command.*
Can be throwed by commands: All existing commands might return this error.

BILLING_FAILURE (code: 2104)

The billing process failed. Check you've got enought cash and retry opperation.

Error message: *Billing error.*
Can be throwed by commands: Billing_Transfer_Domain, Billing_Renew_Multihosting, Billing_Renew_Domain, Billing_Renew_Sms, Billing_Renew_Hosting, Billing_Buy_Multihosting, Billing_Buy_Domain, Billing_Buy_Sms, Billing_Buy_DomainUsingContactsIds, Billing_Buy_Hosting, Billing_Autorenew_SetOn

BILLING_NOT_ALLOWED (code: 2105)

User is not authorized to perform any billing operations.

Error message: *Billing operations are not allowed.*
Can be throwed by commands: None

AUTH_ERROR_USER (code: 2200)

The user authentication failed. Check that username and password sent on request are valid.
Keep in mind that some request types require sending auth info on HTTP headers.

Error message: *Authentication error.*
Can be throwed by commands: All existing commands might return this error.

AUTH_ERROR_OBJECT (code: 2201)

The authenticated user is not authorized to execute this action on the required object (domain, hosting, ...).
This could happen if you aren't the owner of the object, or the object internal status doesn't allow this action (it isn't active ...).

Error message: *Authorization error.*
Can be throwed by commands: All existing commands might return this error.

OBJECT_EXISTS (code: 2302)

The request is valid but it can be only completed if the requested object (zone, hosting, email account...) doesn't exists.

Error message: *Param "%s" value already exists.*
Can be throwed by commands: Domain_Zone_AddTypeSRV, Domain_Zone_AddTypeFrame, Domain_Zone_AddTypeAAAA, Domain_Zone_AddTypeURL301, Domain_Zone_AddTypeURL, Domain_Zone_AddTypeMXDetailedEmail, Domain_Zone_AddTypeMXS, Domain_Zone_AddTypeTXT, Domain_Zone_AddTypeMX, Domain_Zone_UrlShortener_Activate, Domain_Zone_AddTypeCname, Domain_Zone_AddTypeAname, Domain_Zone_AddTypeA, Billing_Transfer_Domain, Billing_Buy_Multihosting, Billing_Buy_Domain, Billing_Buy_Sms, Billing_Buy_DomainUsingContactsIds, Billing_Buy_Hosting, Dedicated_HostingSetup_Create, Dedicated_Hosting_Create, Vps_HostingSetup_Create, Vps_Hosting_Create, Multihosting_HostingSetup_Create, Multihosting_Hosting_Create, Hosting_Ddbb_MariaDb_Create, Hosting_Ddbb_Postgre_Create, Hosting_Ddbb_Mysql_CreateWithIpAcl, Hosting_Ddbb_Mysql_Create, Hosting_Ddbb_Sqlserver_Create, Hosting_Import_ApplyForEmailAccountImap, Hosting_Import_ApplyForFtp, Hosting_Import_ApplyForDdbb, Hosting_Import_ApplyForEmailAccountPop, Hosting_Dsn_Mysql_Create, Hosting_Dsn_Access_Create, Hosting_Dsn_Sqlserver_Create, Hosting_Subdomain_Create, Hosting_Email_Autoresponse_Create, Hosting_Email_Redirection_Create, Hosting_Email_Antispam_AddDomainToBlacklist, Hosting_Email_Antispam_AddDomainToWhitelist, Hosting_Email_Account_Create

OBJECT_NOT_EXISTS (code: 2303)

The request is valid but it can be only completed if the requested object (zone, hosting, email account ...) already exists.

Error message: *Param "%s" value doesn't exist.*
Can be throwed by commands: Domain_Zone_UpdateTypeA, Domain_Zone_Delete, Domain_Zone_DeleteTypeA, Domain_Zone_DeleteTypeSRV, Domain_Zone_DeleteTypeMX, Domain_Zone_DeleteTypeTXT, Domain_Zone_AddTypeMXDetailedEmail, Domain_Zone_DeleteTypeAname, Domain_Zone_UpdateTypeAname, Domain_Zone_DeleteTypeCname, Domain_Zone_AddTypeMX, Domain_Zone_UpdateTypeAAAA, Domain_Zone_AddTypeCname, Domain_Zone_DeleteTypeAAAA, Domain_Zone_DeleteTypeSPF, Billing_Transfer_Domain, Billing_Renew_Domain, Billing_Renew_Hosting, Billing_Autorenew_SetOff, Billing_Autorenew_SetOn, Billing_Autorenew_SetNoRenewable, Billing_NoRenewable_SetRenewable, Sms_Programmed_Cancel, Dedicated_Services_ChangeStatus, Dedicated_HostingSetup_Get, Dedicated_HostingSetup_Update, Dedicated_Control_KillMysqlProcess, Dedicated_Hosting_Delete, Dedicated_Hosting_ChangeSetup, Dedicated_Hosting_Activate, Dedicated_Hosting_GetSetupName, Dedicated_Hosting_Deactivate, Dedicated_Hosting_GetSpace, Dedicated_Hosting_Create, Vps_Services_ChangeStatus, Vps_HostingSetup_Get, Vps_HostingSetup_Update, Vps_Control_KillMysqlProcess, Vps_Hosting_Delete, Vps_Hosting_ChangeSetup, Vps_Hosting_Activate, Vps_Hosting_GetSetupName, Vps_Hosting_Deactivate, Vps_Hosting_Create, Multihosting_HostingSetup_Get, Multihosting_HostingSetup_Update, Multihosting_Hosting_Delete, Multihosting_Hosting_ChangeSetup, Multihosting_Hosting_Activate, Multihosting_Hosting_Deactivate, Multihosting_Hosting_GetSpace, Hosting_Import_ApplyForEmailAccountImap, Hosting_Import_ApplyForFtp, Hosting_Import_ApplyForDdbb, Hosting_Import_ApplyForEmailAccountPop, Hosting_Subdomain_Delete, Hosting_Subdomain_Create, Hosting_Email_Autoresponse_Delete, Hosting_Email_Autoresponse_Get, Hosting_Email_Autoresponse_Activate, Hosting_Email_Autoresponse_Deactivate, Hosting_Email_Autoresponse_Create, Hosting_Email_Redirection_Delete, Hosting_Email_Redirection_Update, Hosting_Email_Antispam_RemoveDomainFromBlacklist, Hosting_Email_Antispam_RemoveDomainFromWhitelist, Hosting_Email_Account_Delete, Hosting_Email_Account_SetCopyTo, Hosting_Email_Account_SetPassword, Hosting_Email_Account_Empty, Hosting_Email_Account_Activate, Hosting_Email_Account_GetCopyTo, Hosting_Email_Account_RemoveCopyTo, Hosting_Email_Account_Deactivate, Hosting_Ftp_Account_Delete, Hosting_Ftp_Account_SetPassword

OBJECT_STATUS_PROHIBITS_OP (code: 2304)

The request is valid but it can be only completed if the requested object status meets some condition.
Some condition must be archieved before executing the command again.
Please, read the command documentation for additional information about command limitations and requeriments.

Error message: *Param "%s" current value status prohibits operation.*

Can be throwed by commands: Domain_Transfer_SetAuthcode, Domain_Transfer_Restart, Domain_Transfer_Cancel, Domain_Transfer_GetAuthcode, Domain_Transfer_GetStatus, Domain_Zone_AddTypeMXDetailedEmail, Domain_Zone_AddTypeMX, Domain_Zone_UrlShortener_Delete, Domain_Zone_UrlShortener_Activate, Domain_Zone_UrlShortener_Deactivate, Domain_Zone_UrlShortener_Create, Domain_Zone_UrlShortener_GetAll, Domain_UrlShortener_Delete, Domain_UrlShortener_Activate, Domain_UrlShortener_Deactivate, Domain_UrlShortener_Create, Domain_UrlShortener_GetAll, Domain_WhoisPrivacy_Set, Billing_Autorenew_SetOn, Hosting_Subdomain_Delete, Hosting_Subdomain_Create, Hosting_Subdomain_GetAll, Hosting_UrlShortener_Delete, Hosting_UrlShortener_Activate, Hosting_UrlShortener_Deactivate, Hosting_UrlShortener_Create, Hosting_UrlShortener_GetAll, Hosting_Ftp_Account_Create, Hosting_TlsMail_Get, Hosting_TlsMail_Activate, Hosting_TlsMail_Deactivate

OBJECT_ASSOCIATION_PROHIBITS_OP (code: 2305)

The requets is valid but cant be run because the object is currently associated to another object.
Association must be broken before running this command.

Error message: *Param "%s" current value is associated to another object. Association must be broken.*
Can be throwed by commands: Domain_Zone_Delete, Domain_Zone_DeleteTypeA, Domain_Zone_DeleteTypeCname, Domain_Zone_DeleteTypeAAAA

PARAMETER_VALUE_POLICY_ERROR (code: 2306)

A internal or external parameter value is not valid. It's registry implementation dependant.
This is relatively common if domain contacts are missing.

Error message: *Some argument value is not currently valid*
Can be throwed by commands: Domain_Dnss_Set, Billing_Autorenew_SetOn, Hosting_Import_ApplyForFtp

UNIMPLEMENTED_OBJECT_SERVICE (code: 2307)

The request is valid but requested object won't admit it, neither now nor later.
Please, read the command documentation for additional information about command limitations and requeriments.

Error message: *Command cannot be run on requested object.*
Can be throwed by commands: Domain_CheckForTransfer, Domain_Transfer_SetAuthcode, Domain_Transfer_GetAuthcode, Domain_CheckForRegister, Domain_NameServer_GetAll, Billing_Renew_Domain, Billing_Autorenew_SetOn, Dedicated_ForVps_Vps_Delete, Dedicated_ForVps_Vps_Update, Dedicated_ForVps_Vps_Activate, Dedicated_ForVps_Vps_Deactivate, Dedicated_ForVps_Vps_Create_Unmanaged, Dedicated_ForVps_Vps_Create_Managed, Dedicated_ForVps_Vps_GetTypes, Dedicated_ForVps_Vps_GetAll, Dedicated_GetMonitorizedPorts, Vps_GetMonitorizedPorts, Hosting_Ddbb_MariaDb_Delete, Hosting_Ddbb_MariaDb_SetPassword, Hosting_Ddbb_MariaDb_Create, Hosting_Ddbb_MariaDb_GetAll, Hosting_Ddbb_Postgre_Delete, Hosting_Ddbb_Postgre_SetPassword, Hosting_Ddbb_Postgre_Create, Hosting_Ddbb_Postgre_GetAll, Hosting_Ddbb_Mysql_SetHostAcl, Hosting_Ddbb_Mysql_Delete, Hosting_Ddbb_Mysql_GetAclConfiguration, Hosting_Ddbb_Mysql_CreateWithIpAcl, Hosting_Ddbb_Mysql_SetPassword, Hosting_Ddbb_Mysql_BackupTask_Delete, Hosting_Ddbb_Mysql_BackupTask_Set, Hosting_Ddbb_Mysql_BackupTask_GetAll, Hosting_Ddbb_Mysql_GetInfo, Hosting_Ddbb_Mysql_SetIpAcl, Hosting_Ddbb_Mysql_Backup_Get, Hosting_Ddbb_Mysql_Backup_Restore, Hosting_Ddbb_Mysql_Create, Hosting_Ddbb_Mysql_GetAll, Hosting_Ddbb_Sqlserver_Delete, Hosting_Ddbb_Sqlserver_GetSpaceInfo, Hosting_Ddbb_Sqlserver_SetPassword, Hosting_Ddbb_Sqlserver_SetTransactionsSize, Hosting_Ddbb_Sqlserver_GetGlobalSpaceInfo, Hosting_Ddbb_Sqlserver_SetDataSize, Hosting_Ddbb_Sqlserver_Create, Hosting_Ddbb_Sqlserver_GetAll, Hosting_Ddbb_MongoDb_Delete, Hosting_Ddbb_MongoDb_SetPassword, Hosting_Ddbb_MongoDb_GetAll, Hosting_Dsn_Mysql_Delete, Hosting_Dsn_Mysql_Create, Hosting_Dsn_Mysql_GetAll, Hosting_Dsn_Access_Delete, Hosting_Dsn_Access_Create, Hosting_Dsn_Access_GetAll, Hosting_Dsn_Sqlserver_Delete,

Hosting_Dsn_Sqlserver_Create, Hosting_Dsn_Sqlserver_GetAll, Hosting_Ssh_Get, Hosting_Ssh_Activate, Hosting_Ssh_Deactivate, Hosting_GetMailInfo, Hosting_Email_Autoresponse_Delete, Hosting_Email_Autoresponse_Get, Hosting_Email_Autoresponse_Activate, Hosting_Email_Autoresponse_Deactivate, Hosting_Email_Autoresponse_Create, Hosting_Email_Redirection_Delete, Hosting_Email_Redirection_DeactivateCatchAll, Hosting_Email_Redirection_Get, Hosting_Email_Redirection_Update, Hosting_Email_Redirection_ActivateCatchAll, Hosting_Email_Redirection_Create, Hosting_Email_Redirection_GetAll, Hosting_Email_Filter_CreateForSpam, Hosting_Email_Filter_Delete, Hosting_Email_Filter_GetGeolocalization, Hosting_Email_Filter_SetGeolocalization, Hosting_Email_Filter_Create, Hosting_Email_Filter_GetAll, Hosting_Email_Antispam_AddDomainToBlacklist, Hosting_Email_Antispam_RemoveDomainFromBlacklist, Hosting_Email_Antispam_Activate, Hosting_Email_Antispam_SetConfig, Hosting_Email_Antispam_Deactivate, Hosting_Email_Antispam_GetScore, Hosting_Email_Antispam_RemoveDomainFromWhitelist, Hosting_Email_Antispam_GetBlacklist, Hosting_Email_Antispam_GetConfig, Hosting_Email_Antispam_AddDomainToWhitelist, Hosting_Email_Antispam_SetScore, Hosting_Email_Antispam_GetWhitelist, Hosting_Email_Account_Delete, Hosting_Email_Account_Get, Hosting_Email_Account_SetCopyTo, Hosting_Email_Account_SetPassword, Hosting_Email_Account_Empty, Hosting_Email_Account_Activate, Hosting_Email_Account_GetCopyTo, Hosting_Email_Account_RemoveCopyTo, Hosting_Email_Account_Deactivate, Hosting_Email_Account_Create, Hosting_Email_Account_GetAll, Hosting_Email_Antivirus_GetAttachedFilesFilters, Hosting_Email_Antivirus_SetConfig, Hosting_Email_Antivirus_GetConfig, Hosting_Email_Antivirus_UpdateAttachedFileFilter, Hosting_Ftp_Get, Hosting_Ftp_GetAccountsLimit, Hosting_Ftp_Activate, Hosting_Ftp_Deactivate, Hosting_Ftp_Account_Delete, Hosting_Ftp_Account_SetPassword, Hosting_Ftp_Account_Create, Hosting_Ftp_Account_GetAll, Hosting_Php_PhpExecutionOnHtmlIsActive, Hosting_Php_ActivatePhpExecutionOnHtml, Hosting_Php_DeactivatePhpExecutionOnHtml, Hosting_Php_SetVersion, Hosting_Php_OverwritePhpIniWithDefaults, Hosting_Php_OverwritePhpIni, Hosting_TlsMail_Get, Hosting_TlsMail_Activate, Hosting_TlsMail_Deactivate

COMMAND_FAILED (code: 2400)

Command failed because of an internal error. Command request is valid but we couldn't finish it.
You can try again later o contact with support for additional details.

Error message: *Command failed.*
Can be throwed by commands: All existing commands might return this error.

COMMAND_FAILED_FATAL (code: 2500)

Command execution ended with an internal fatal error.
The execution might end OK but something wrent wrong. Or maybe command execution doesn't started at all.
We'll look at the problem and fix it ASAP.

Error message: *Command failed, internal error.*
Can be throwed by commands: All existing commands might return this error.

COMMAND_TIMEOUT (code: 2501)

We received a timeout during execution of command.
Command execution could end ok but we didn't get the response at time.

Error message: *Timeout on execution, command might be complete.*
Can be throwed by commands: All existing commands might return this error.

COMMAND_CONNECTIONSLIMIT (code: 2502)

User tried to complete too many command executions in a period of time and/or got banned.
Please, just wait and try again later.

Error message: *Too many connections, command won't execute. Try again later.*
Can be throwed by commands: None

COMMAND_MAINTENANCE_MODE (code: 2503)

Server is in maintenance mode.

Error message: *Maintenance mode. Try again later.*
Can be throwed by commands: None

COMMAND_PENDING_PREVIOUS_EXECUTION (code: 3101)

There is a existing execution for this delayed command.

Error message: *There is a previous pending execution.*
Can be throwed by commands: Hosting_Ftp_Account_Delete, Hosting_Ftp_Account_SetPassword, Hosting_Ftp_Account_Create

COMMAND_FTP_MAX_ACCOUNTS (code: 3601)

Hosting already reached maximum FTP accounts number.

Error message: *Reached max number of FTP accounts.*
Can be throwed by commands: None

COMMAND_EMAIL_REDIRECTION_MAX_ACCOUNTS (code: 3602)

Hosting already reached maximum email redirections number.

Error message: *Reached max number of email redirections.*
Can be throwed by commands: Hosting_Email_Redirection_Create

COMMAND_EMAIL_DISTRIBUTIONLIST_MAX (code: 3603)

Hosting already reached maximum distribution list number.

Error message: *Reached max number of email distribution lists.*
Can be throwed by commands: None

COMMAND_SERVER_KVM_GET_ERROR (code: 3604)

Server is missing KVM information.

Error message: *Missing KVM information.*
Can be throwed by commands: Dedicated_Kvm_Get, Housing_Kvm_Get

COMMAND_DEDICATED_SERVER_OS_ERROR (code: 3605)

Server Operating System can't run this command.

Error message: *Server O.S. is not compatible with this command.*
Can be throwed by commands: Dedicated_Control_KillMysqlProcess, Dedicated_Control_GetApacheProcesses, Vps_Control_KillMysqlProcess

COMMAND_LICENSE_MAX_HOSTINGS (code: 3606)

Server has reached the maximum number of hostings.
This limit is defined by the current server license.

Error message: *Reached maximum number of hostings - license limit.*
Can be throwed by commands: Dedicated_Hosting_Create, Vps_Hosting_Create, Multihosting_Hosting_Create

COMMAND_BILLING_PRODUCT_ERROR (code: 3607)

The requested product is not valid and billing process is aborted.
Remember that not all domain TLDs share same registration periods.

Error message: *Invalid product.*
Can be throwed by commands: Billing_Transfer_Domain, Billing_Renew_Multihosting, Billing_Renew_Domain, Billing_Renew_Sms, Billing_Renew_Vps, Billing_Renew_Hosting, Billing_Buy_Multihosting, Billing_Buy_Domain, Billing_Buy_Sms, Billing_Buy_Vps_Unmanaged_Windows, Billing_Buy_Vps_Unmanaged_Linux, Billing_Buy_Vps_Managed_Windows, Billing_Buy_Vps_Managed_Linux, Billing_Buy_DomainUsingContactsIds, Billing_Buy_Hosting, Billing_Price_Multihosting, Billing_Price_Domain, Billing_Price_Sms, Billing_Price_Vps_Unmanaged_Windows, Billing_Price_Vps_Unmanaged_Linux, Billing_Price_Vps_Managed_Windows, Billing_Price_Vps_Managed_Linux, Billing_Price_Hosting, Billing_Help_DomainExtraData, Billing_Period_Multihosting, Billing_Period_Domain, Billing_Period_Vps, Billing_Period_Hosting

COMMAND_BILLING_CREDIT (code: 3608)

The user has not enought credit for completing the billing process.

Error message: *Not enought credit.*
Can be throwed by commands: Billing_Transfer_Domain, Billing_Renew_Multihosting, Billing_Renew_Domain, Billing_Renew_Sms, Billing_Renew_Hosting, Billing_Buy_Multihosting, Billing_Buy_Domain, Billing_Buy_Sms, Billing_Buy_DomainUsingContactsIds, Billing_Buy_Hosting

COMMAND_EMAIL_MAX_HOSTINGS (code: 3609)

Server cannot create more hostings currently on this server.
It has reached the email accounts limit. Please, delete some and try again.

Error message: *Reached maximum number of email accounts on server.*
Can be throwed by commands: Hosting_Email_Account_Create

COMMAND_DOMAIN_FORBIDDEN_CONTACT_UPDATE (code: 3610)

Domain contacts update can't be run because of registry policy blocks certaing contactType data change.
Example: .es registrant and .eu registrant contacts update are forbidden. These must be removed from request.

Error message: *Requested contact change is forbidden by registry policy.*
Can be throwed by commands: Domain_Contacts_SetAll, Domain_Contacts_Set

COMMAND_DDBB_SQLSERVER_NOT_ENOUGH_SPACE (code: 3612)

There is not enough space for the requested SqlServer operation.

Error message: *Not enough space. Check params "%s".*
Can be throwed by commands: Hosting_Ddbb_Sqlserver_Create

COMMAND_EMAIL_DISTRIBUTION_LIST_EXISTS (code: 3613)

Can't create the account. It exists a distribution list with the same email.

Error message: *It exists a distribution list with the same email.*
Can be throwed by commands: None

COMMAND_SMS_NOT_REMAINING_MESSAGES (code: 3614)

The user has not remaining messages for completing the sending process.

Error message: *Not remaining messages.*
Can be throwed by commands: None

COMMAND_MYSQL_MAX_DATABASES (code: 3615)

Server cannot create more MySQL databases currently on this server.

Error message: *Reached max number of MySQL databases.*
Can be throwed by commands: None

COMMAND_DOMAIN_IP_NOT_RESOLVING_DNS (code: 3616)

Some IP doesn't resolve a DNS.

Error message: *Some IP doesn't resolve a DNS.*
Can be throwed by commands: None

COMMAND_EMAIL_NOT_ENABLED (code: 3617)

Email accounts are not allowed on this hosting.

Error message: *Email accounts are not allowed on this hosting.*
Can be throwed by commands: None

COMMAND_REGENERATE_PASSWORD_SEND_METHOD (code: 3618)

Send method is not valid for this contact.

Error message: *Send method is not valid for this contact.*
Can be throwed by commands: None

COMMAND_EMAIL_ACCOUNT_SET_REPEATED_PASSWORD (code: 3619)

Password can not match any of the last 5 used in this account..

Error message: *Password can not match any of the last 5 used in this account.*
Can be throwed by commands: Hosting_Email_Account_SetPassword

COMMAND_BILLING_DOMAIN_FORBIDDEN_PREDEFINED_CONTACTS (code: 3

Some registries have special requirements that must be fit by user default domain registration contacts.

Error message: *Invalid predefined contacts for this TLD .%s*
Can be throwed by commands: Billing_Buy_Domain, Billing_Buy_DomainUsingContactsIds

COMMAND_DOMAIN_TRANSFER_ALREADY_APPROVED (code: 3630)

Domain transfer was already approved.

Error message: *Domain transfer was already approved.*
Can be throwed by commands: Domain_Transfer_Restart, Domain_Transfer_Cancel

COMMAND_DOMAIN_CERT_ALREADY_ASSOCIATED (code: 3631)

A cert has already been associated with this domain.

Error message: *A cert has already been associated with this domain.*
Can be throwed by commands: None

COMMAND_BILLING_PRODUCT_TEMPORARILY_DISABLED (code: 3621)

Product reference temporarily disabled by promotion or other reasons

Error message: *Product reference temporarily disabled by promotion or other reasons.*
Can be throwed by commands: None

COMMAND_NFS_VOLUME_DOES_NOT_EXIST (code: 3640)

Server does not have a nfs volume

Error message: *Server does not have a nfs volume*
Can be throwed by commands: None

COMMAND_PARAM_IS_NOT_A_VALID_CERTIFICATE (code: 3641)

(Missing documentation).
Error message: *(Missing response message, the error code might be on implementation).*
Can be throwed by commands: None

COMMAND_DOMAIN_SERVER_ZONE_NOT_FOUND (code: 3645)

Domain's zone for server not found

Error message: *Domain's zone for server not found*
Can be throwed by commands: None

COMMAND_ADD_ZONES_LIMIT (code: 3646)

No more zones can be added to the domain.

Error message: *Reached max number of Zones DNS*
Can be throwed by commands: Domain_Zone_AddTypeSRV, Domain_Zone_AddTypeFrame, Domain_Zone_AddTypeAAAA, Domain_Zone_AddTypeURL301, Domain_Zone_AddTypeURL, Domain_Zone_AddTypeMXDetailedEmail, Domain_Zone_AddTypeMXS, Domain_Zone_AddTypeTXT, Domain_Zone_AddTypeMX, Domain_Zone_AddTypeCname, Domain_Zone_AddTypeAname, Domain_Zone_AddTypeA

Commands

Billing commands

Billing_Transfer_Domain

Description:
Buy and transfer to Dinahosting an existing domain name.
Some domains require an auth code to complete the transfer. There is a command for this: Api_Command_Domain_Transfer_SetAuthcode.
1.0 -> 1.1 Improved simulation mode.
1.1 -> 1.2: COMMAND_BILLING_CREDIT error.
1.2 -> 1.3: OBJECT_EXISTS and OBJECT_NOT_EXISTS errors.
1.3 -> 1.4: New TLDs are supported.
1.4 -> 1.5: New contact keys (RegistrantIdentification, AdminIdentification, TechIdentification, BillingIdentification) - required for esNic domain validation.
EsNic domain validation for technical contact: must be a natural person
Version: 1.5

Parameters:
domain (string): Domain name. The parameter is required.
period (int): Registration period (years). Default: 1 year. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 10 inclusive.
contacts (struct): Array containing contacts (see Domain_Contacts_GetAll). Optional, but many of the tlds require registrant information (see Billing_Transfer_GetRequirements). The parameter is optional. Parameter value must be a valid contacts.
dnss (array): Array containing DNS strings (see Domain_Dnss_Get). Optional. The parameter is optional. Parameter value must be a valid array of 2-8 unique DNS.
whoisProtection (bool): Default is false. Set true to buy domain whois protection. Is activated inmediatly after transfer this domain. The parameter is optional. Parameter value must be a boolean (true or false).
paymentMethodType (string): Default is saldo. Set it to "bono" to use that payment Method. The parameter is optional. Accepted parameter values: "saldo","bono".

Response:
type: void

Billing_Transfer_GetRequirements

Description:
Gets transfer requirements.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: struct
Example:
'None'

Billing_Renew_Multihosting

Description:
Manually renew an existing multihosting. Payment is done at runtime.
1.0 -> 1.1: Improved simulation mode.

1.1 -> 1.2: COMMAND_BILLING_CREDIT error.
Version: 1.2

Parameters:
serverName (string): Server name. The parameter is required.
period (int): Renew period (years). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 5 inclusive.

Response:
type: void

Billing_Renew_Domain

Description:
Manually renew an existing domain name. Payment is done at runtime.
Domain must be currently registered at Dinahosting.
1.0 -> 1.1: Improved simulation mode.
1.1 -> 1.2: COMMAND_BILLING_CREDIT error.
1.2 -> 1.3: OBJECT_NOT_EXISTS error.
1.3 -> 1.4: New TLDs are supported.
Version: 1.4

Parameters:
domain (string): Domain name. The parameter is required.
period (int): Renew period (years). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 10 inclusive.
whoisProtection (bool): Default is false. Set true to also renovate an already active whois protection for this domain. The parameter is optional. Parameter value must be a boolean (true or false).
paymentMethodType (string): Default is saldo. Set it to "bono" to use that payment Method. The parameter is optional. Accepted parameter values: "saldo","bono".

Response:
type: void

Billing_Renew_Sms

Description:
Manually renew an existing SMS account. Payment is done at runtime.
1.0 -> 1.1: Improved simulation mode.
1.1 -> 1.2: COMMAND_BILLING_CREDIT error.
Version: 1.2

Parameters:
account (string): Sms account number. The parameter is required.
credit (int): Aditional credit. The parameter is required. Parameter value must be a valid integer. Accepted parameter values: "50","100","1000","5000","20000".

Response:
type: void

Billing_Renew_Vps

Description:

Renews a VPS.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
period (int): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".

Response:
type: boolean
Example:
'None'

Billing_Renew_Hosting

Description:
Manually renew an existing hosting. Payment is done at runtime.
Hosting must exist in a Dinahosting server.
1.0 -> 1.1: Improved simulation mode.
1.1 -> 1.2: COMMAND_BILLING_CREDIT error.
Version: 1.2

Parameters:
hosting (string): Hosting name. The parameter is required.
period (int): Contratation period (months). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 120 inclusive.

Response:
type: void

Billing_Buy_Multihosting

Description:
Buy a new multihosting.
1.2 -> 1.3: Improved simulation mode.
1.3 -> 1.4: COMMAND_BILLING_CREDIT error.
Version: 1.4

Parameters:
period (int): Contratation period (years). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 5 inclusive.
serverOs (string): Host server operating system. The parameter is required. Accepted parameter values: "LINUX","WINDOWS".

Response:
type: void

Billing_Buy_Domain

Description:
Buy and register a unused domain name.
1.0 -> 1.1: Improved simulation mode.
1.1 -> 1.2: COMMAND_BILLING_CREDIT error.

1.2 -> 1.3: New TLDs are supported.
1.3 -> 1.4: New contact keys (RegistrantIdentification, AdminIdentification, TechIdentification, BillingIdentification) - required for esNic domain validation.
1.4 -> 1.5: New optional contact keys (RegistrantFax, AdminFax, TechFax, BillingFax)
1.5 -> 1.6: New parameter extraData.
1.6 -> 1.7: New validation for esNic registration and new error code: COMMAND_BILLING_DOMAIN_FORBIDDEN_PREDEFINED_CONTACTS
Version: 1.7

Parameters:
domain (string): Domain name. The parameter is required.
period (int): Registration period (years). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 10 inclusive.
contacts (struct): Array containing contacts (see Domain_Contacts_GetAll). Omit OrganizationName and LegalForm parameters in cases where contact is not an organization. The parameter is required. Parameter value must be a valid contacts.
dnss (array): Array containing at least 2 DNS strings. The parameter is optional. Parameter value must be a valid array of 2-8 unique DNS.
whoisProtection (bool): Default is false. Set true to buy domain whois protection. Is activated inmediatly after register this domain. The parameter is optional. Parameter value must be a boolean (true or false).
showRegistrantInfoOnWhois (bool): Default is false. Set true to show registrant info on whois. The parameter is optional. Parameter value must be a boolean (true or false).
showAdminContactInfoOnWhois (bool): Default is false. Set true to show admin contact info on whois. The parameter is optional. Parameter value must be a boolean (true or false).
showTechContactInfoOnWhois (bool): Default is false. Set true to show tech contact info on whois. The parameter is optional. Parameter value must be a boolean (true or false).
paymentMethodType (string): Default is saldo. Set it to "bono" to use that payment Method. The parameter is optional. Accepted parameter values: "saldo","bono".
extraData (struct): Hash array containing additional data. The parameter is optional.

Response:
type: void

Billing_Buy_Sms

Description:
Buy a new SMS account.
1.0 -> 1.1: Improved simulation mode.
1.1 -> 1.2: COMMAND_BILLING_CREDIT error.
Version: 1.2

Parameters:
account (string): Sms account number. The parameter is required.
credit (int): Initial credit. The parameter is required. Parameter value must be a valid integer. Accepted parameter values: "50","100","1000","5000","20000".

Response:
type: void

Billing_Buy_Vps_Unmanaged_Windows

Description:
Buys an unmanaged windows VPS.
Version: 1.0

Parameters:
type (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".
period (int): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".
vcpusCount (int): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".
diskType (string): Disk type. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.
diskImage (string): Disk image. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk images. The parameter is required.

Response:
type: boolean
Example:
'None'

Billing_Buy_Vps_Unmanaged_Linux

Description:
Buys an unmanaged linux VPS.
Version: 1.0

Parameters:
type (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".
period (int): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".
vcpusCount (int): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".
diskType (string): Disk type. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.
diskImage (string): Disk image. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk images. The parameter is required.

Response:
type: boolean
Example:
'None'

Billing_Buy_Vps_Managed_Windows

Description:
Buys a managed windows VPS.
Version: 1.0

Parameters:
type (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".
period (int): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".
vcpusCount (int): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".
diskType (string): Disk type. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.

Response:
type: boolean
Example:

'None'

Billing_Buy_Vps_Managed_Linux

Description:
Buys a managed linux VPS.
Version: 1.0

Parameters:
type (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".
period (int): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".
vcpusCount (int): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".
diskType (string): Disk type. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.
diskImage (string): Disk image. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk images. The parameter is required.

Response:
type: boolean
Example:
'None'

Billing_Buy_DomainUsingContactsIds

Description:
Buy and register a unused domain name.
1.0 -> 1.1: Improved simulation mode.
1.1 -> 1.2: COMMAND_BILLING_CREDIT error.
1.2 -> 1.3: New TLDs are supported.
1.3 -> 1.4: EsNic domain validation for technical contact: must be a natural person
1.4 -> 1.5: New parameter extraData.
1.5 -> 1.6: New validation for esNic registration and new error code: COMMAND_BILLING_DOMAIN_FORBIDDEN_PREDEFINED_CONTACTS
Version: 1.6

Parameters:
domain (string): Domain name. The parameter is required.
period (int): Registration period (years). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 10 inclusive.
contacts (struct): Array containing contacts ids. The parameter is required. Parameter value must be valid contacts ids.
dnss (array): Array containing at least 2 DNS strings. The parameter is optional. Parameter value must be a valid array of 2-8 unique DNS.
whoisProtection (bool): Default is false. Set true to buy domain whois protection. Is activated inmediatly after register this domain. The parameter is optional. Parameter value must be a boolean (true or false).
showRegistrantInfoOnWhois (bool): Default is false. Set true to show registrant info on whois. The parameter is optional. Parameter value must be a boolean (true or false).
showAdminContactInfoOnWhois (bool): Default is false. Set true to show admin contact info on whois. The parameter is optional. Parameter value must be a boolean (true or false).
showTechContactInfoOnWhois (bool): Default is false. Set true to show tech contact info on whois. The parameter is optional. Parameter value must be a boolean (true or false).
paymentMethodType (string): Default is saldo. Set it to "bono" to use that payment Method. The parameter is optional. Accepted parameter values: "saldo","bono".
extraData (struct): Hash array containing additional data. The parameter is optional.

Response:
type: void

Billing_Buy_Hosting

Description:
Buy a new hosting.
1.2 -> 1.3: Improved simulation mode. Deprecated hostingType 810.
1.3 -> 1.4: COMMAND_BILLING_CREDIT error.
Version: 1.4

Parameters:
hosting (string): Hosting name. The parameter is required.
period (int): Contratation period (months). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 120 inclusive.
hostingType (int): Hosting type. 1 = Basic, 2 = Advanced, 3 = Advanced Plus, 10 = Email, 419 = Wordpress, 420 = Wordpress Optimized, 421 = Wordpress Pro, 422 = Wordpress Pro Optimized, 425 = Prestashop, 426 = Prestashop Pro, 428 = WooCommerce, 429 = WooCommerce Pro, 431 = MediaWiki, 470 = Joomla, 480 = Moodle, 490 = Drupal. The parameter is required. Accepted parameter values: "1","2","3","10","419","420","421","422","425","426","428","429","431","470","480","490".
serverOs (string): Host server operating system. The parameter is required. Accepted parameter values: "LINUX","WINDOWS".

Response:
type: void

Billing_Price_Multihosting

Description:
Get price for a multihosting (new or renew).
Version: 1.2

Parameters:
period (int): Contratation period (years). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 5 inclusive.
serverOs (string): Host server operating system. The parameter is required. Accepted parameter values: "LINUX","WINDOWS".
action (string): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew".
currency (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

Response:
type: float
Example:
'None'

Billing_Price_Domain

Description:
Get price for a domain name (register, transfer or renew).
1.0 -> 1.1: New TLDs are supported.
Version: 1.1

Parameters:
tld (string): Domain name TLD. Example: com.es (as is). The parameter is required.
period (int): Registration period (years). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 10 inclusive.
action (string): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew","transfer".
whoisProtection (bool): Default is false. Set true to include domain whois protection price. The parameter is optional. Parameter value must be a boolean (true or false).
currency (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

Response:
type: float
Example:
'None'

Billing_Price_Sms

Description:
Get price for a SMS account (new or renew).
Version: 1.0

Parameters:
credit (int): Initial credit. The parameter is required. Parameter value must be a valid integer. Accepted parameter values: "50","100","1000","5000","20000".
action (string): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew".
currency (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

Response:
type: float
Example:
'None'

Billing_Price_Vps_Unmanaged_Windows

Description:
Gets the price of an unmanaged windows VPS.
Version: 1.0

Parameters:
type (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".
period (int): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".
vcpusCount (int): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".
diskType (string): Disk type. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.
diskImage (string): Disk image. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk images. The parameter is required.
action (string): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew".
currency (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values:

"USD","EUR".

Response:
type: float
Example:
'None'

Billing_Price_Vps_Unmanaged_Linux

Description:
Gets the price of an unmanaged linux VPS.
Version: 1.0

Parameters:
type (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".
period (int): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".
vcpusCount (int): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".
diskType (string): Disk type. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.
action (string): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew".
currency (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

Response:
type: float
Example:
'None'

Billing_Price_Vps_Managed_Windows

Description:
Gets the price of a managed windows VPS.
Version: 1.0

Parameters:
type (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".
period (int): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".
vcpusCount (int): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".
diskType (string): Disk type. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.
action (string): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew".
currency (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

Response:
type: float
Example:
'None'

Billing_Price_Vps_Managed_Linux

Description:
Gets the price of a managed linux VPS.
Version: 1.0

Parameters:
type (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".
period (int): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".
vcpusCount (int): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".
diskType (string): Disk type. Use the command 'Billing_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.
action (string): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew".
currency (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

Response:
type: float
Example:
'None'

Billing_Price_Hosting

Description:
Get price for a hosting (new or renew).
Version: 1.2

Parameters:
period (int): Contratation period (months). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 120 inclusive.
hostingType (int): Hosting type. 1 = Basic, 2 = Advanced, 3 = Advanced Plus, 10 = Email, 419 = Wordpress, 420 = Wordpress Optimized, 421 = Wordpress Pro, 422 = Wordpress Pro Optimized, 425 = Prestashop, 426 = Prestashop Pro, 428 = WooCommerce, 429 = WooCommerce Pro, 431 = MediaWiki, 470 = Joomla, 480 = Moodle, 490 = Drupal. The parameter is required. Accepted parameter values: "1","2","3","10","419","420","421","422","425","426","428","429","431","470","480","490".
serverOs (string): Host server operating system. The parameter is required. Accepted parameter values: "LINUX","WINDOWS".
action (string): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew".
currency (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

Response:
type: float
Example:
'None'

Billing_Autorenew_SetOff

Description:
Set autorenovation OFF for a service. Service won't autorenew automatically at expiration date.
Version: 1.02

Parameters:
id (int): Service ID (as returned by Billing_Autorenew_GetAll). The parameter is required. Parameter value must be a valid integer.

Response:
type: void

Billing_Autorenew_SetOn

Description:
Set autorenovation ON for a service. Service will renew automatically at expiration date.
No payment is done until expiration date.
1.0 -> 1.1: Added some validation error codes for invalid paymentMethodId .
Version: 1.1

Parameters:
id (int): Service ID (as returned by Billing_Autorenew_GetAll). The parameter is required. Parameter value must be a valid integer.
paymentMethodId (int): Payment method ID (as returned by Payment_Method_GetAll). The parameter is required. Parameter value must be a valid integer.
period (int): Renovation period (years or months, see periodUnit parameter). Optional. The parameter is optional. Parameter value must be a valid integer. Parameter value must be between values 1 and 120 inclusive.
periodUnit (string): Period unit. The parameter is required. Accepted parameter values: "YEAR","MONTH".

Response:
type: void

Billing_Autorenew_SetNoRenewable

Description:
Set service as no-renewable.
Service won't autorenew automatically at expiration date, service expiration won't be reported at expiration date, service won't be elligible por renew.
Version: 1.00

Parameters:
id (int): Service ID (as returned by Billing_Autorenew_GetAll). The parameter is required. Parameter value must be a valid integer.

Response:
type: void

Billing_Autorenew_GetAll

Description:
Returns renewable services with autorenovation info.
1.0 -> 1.1 Return only renewable services.
1.1 -> 1.2 Added domain parameter to filter results
Version: 1.2

Parameters:
currency (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".
showRenovationPeriods (bool): True to add possible renovation periods (months) for each service. The parameter is optional. Parameter value must be a boolean (true or false).
domain (string): Domain to filter results The parameter is optional.

Response:
type: array
Example:
(Example too long, see web.)

Billing_GetConfigurableVpsOptions

Description:
Gets configurable options for buying a VPS.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: struct
Example:
'None'

Billing_Help_DomainExtraData

Description:
Get extraData structure information.
Version: 1.0

Parameters:
action (string): Action that might need extraData. The parameter is required. Accepted parameter values: "create","transfer".
tld (string): Domain name TLD. Example: com.es (as is). The parameter is optional.

Response:
type: struct
Example:
'None'

Billing_Period_Multihosting

Description:
Get valid contratation periods (years) for a multihosting.
Version: 1.0

Parameters:
action (string): Action done over multihosting. Different actions may have different periods. The parameter is required. Accepted parameter values: "create","renew".

Response:
type: struct

Example:
array('mh' => array(1,2,3,4,5),
)

Billing_Period_Domain

Description:
Get valid contratation periods (years) for each domain TLD and action.
1.1 -> 1.2: New TLDs are supported.
Version: 1.2

Parameters:
action (string): Action done over domain. Different actions may have different periods. The parameter is required. Accepted parameter values: "create","renew","transfer".
tld (string): Domain name TLD. Example: com.es (as is). The parameter is optional.

Response:
type: struct
Example:
(Example too long, see web.)

Billing_Period_Vps

Description:
Gets the selectable billing periods for VPS.
Version: 1.0

Parameters:
action (string): Action done over VPS. Different actions may have different periods. The parameter is required. Accepted parameter values: "create","renew".

Response:
type: struct
Example:
'None'

Billing_Period_Hosting

Description:
Get valid contratation periods (years) for each hosting type.
Version: 1.0

Parameters:
action (string): Action done over hosting. Different actions may have different periods. The parameter is required. Accepted parameter values: "create","renew".

Response:
type: struct
Example:
array('1' => array(1,2,3,5,10),
'2' => array(1,2,3,5,10),
'3' => array(1,2,3,5,10),
'419' => array(1,2,3,5,10),

'425' => array(1,2,3,5,10),
'431' => array(1,2,3,5,10),
'436' => array(1,2,3,5,10),
'470' => array(1,2,3,5,10),
'498' => array(1,2,3,5,10),
)

Billing_NoRenewable_SetRenewable

Description:
Set service as renowable. Service expiration will be reported at expiration date, service will be elligible por renew.
Service can be autorenewed but you must set autorenew on (see Billing_Autorenew_SetOn).
Version: 1.0

Parameters:
id (int): Service ID (as returned by Billing_NoRenewable_GetAll). The parameter is required. Parameter value must be a valid integer.

Response:
type: void

Billing_NoRenewable_GetAll

Description:
Returns no-renewable services list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
'None'

Dedicated commands

Dedicated_Services_ChangeStatus

Description:
Change server service status.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
serviceName (string): Service name. The parameter is required. Accepted parameter values: "web","mysql","pop3","smtp","ftp","dinascanner".
serviceAction (string): Action over service. The parameter is required. Accepted parameter values: "start","stop","activate","deactivate","restart".

Response:
type: void

Dedicated_Services_GetAll

Description:
Returns services status.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('iis' => 'started',
'pop3' => 'started',
'smtp' => 'started',
'ftp' => 'started',
'mysql' => 'started',
'web' => 'started',
)

Dedicated_ForVps_GetServerInfo

Description:
Returns the information of a dedicated server.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct

Example:
'None'

Dedicated_ForVps_Vps_Delete

Description:
Generates and returns a ticket for deleting a VPS in a dedicated server.
The ticket status can be consulted with the command \'Dedicated_ForVps_GetTicketStatus\'.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
vpsHostname (string): VPS hostname. The parameter is required.

Response:
type: void

Dedicated_ForVps_Vps_Update

Description:
Generates and returns a ticket for updating a VPS in a dedicated server. The ticket status can be consulted with the command \'Dedicated_ForVps_GetTicketStatus\'.
Please note this command has a delayed execution and it'll return SUCCESS_PENDING code on success and a ticket id (integer) on the data field.Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
vpsHostname (string): VPS hostname. The parameter is required.
alias (string): VPS alias. The parameter is optional.
description (string): VPS description. The parameter is optional.
vcpusCount (int): Number of vcpus. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 4 inclusive.
ram (int): RAM (MB). Use the command 'Dedicated_ForVps_GetServerInfo' to check the available RAM. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 512.
diskSpace (int): Disk space (GB). Use the command 'Dedicated_ForVps_GetServerInfo' to check the available space. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 10.
transferLimit (int): Transfer limit (GB). Set empty for unlimited. Use the command 'Dedicated_ForVps_GetServerInfo' to check the avaliable transfer limit. The parameter is optional.

Response:
type: int
Example:
'None'

Dedicated_ForVps_Vps_Activate

Description:
Activates a VPS in a dedicated server.
Version: 1.0

Parameters:

serverName (string): Server name. The parameter is required.
vpsHostname (string): VPS hostname. The parameter is required.

Response:
type: void

Dedicated_ForVps_Vps_Deactivate

Description:
Desactivates a VPS in a dedicated server.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
vpsHostname (string): VPS hostname. The parameter is required.

Response:
type: void

Dedicated_ForVps_Vps_Create_Unmanaged

Description:
Generates and returns a ticket for creating an unmanaged VPS in a dedicated server. The ticket status can be consulted with the command \'Dedicated_ForVps_GetTicketStatus\'.
Please note this command has a delayed execution and it'll return SUCCESS_PENDING code on success and a ticket id (integer) on the data field.Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
alias (string): VPS alias. The parameter is optional.
description (string): VPS description. The parameter is optional.
diskImage (string): Image disk. Use the command 'Dedicated_ForVps_Vps_GetTypes' to get a list of available images. The parameter is required.
vcpusCount (int): Number of vcpus. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 4 inclusive.
ram (int): RAM (MB). Use the command 'Dedicated_ForVps_GetServerInfo' to check the available RAM. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 512.
diskSpace (int): Disk space (GB). Use the command 'Dedicated_ForVps_GetServerInfo' to check the available space. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 10.
transferLimit (int): Transfer limit (GB). Set empty for unlimited. Use the command 'Dedicated_ForVps_GetServerInfo' to check the avaliable transfer limit. The parameter is optional.

Response:
type: int
Example:
'None'

Dedicated_ForVps_Vps_Create_Managed

Description:
Generates and returns a ticket for creating a managed VPS in a dedicated server. The ticket status can be consulted with the command \'Dedicated_ForVps_GetTicketStatus\'.

Please note this command has a delayed execution and it'll return SUCCESS_PENDING code on success and a ticket id (integer) on the data field.Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
alias (string): VPS alias. The parameter is optional.
description (string): VPS description. The parameter is optional.
diskImage (string): Image disk. Use the command 'Dedicated_ForVps_Vps_GetTypes' to get a list of available images. The parameter is required.
vcpusCount (int): Number of vcpus. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 4 inclusive.
ram (int): RAM (MB). Use the command 'Dedicated_ForVps_GetServerInfo' to check the available RAM. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 512.
diskSpace (int): Disk space (GB). Use the command 'Dedicated_ForVps_GetServerInfo' to check the available space. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 10.
transferLimit (int): Transfer limit (GB). Set empty for unlimited. Use the command 'Dedicated_ForVps_GetServerInfo' to check the avaliable transfer limit. The parameter is optional.

Response:
type: int
Example:
'None'

Dedicated_ForVps_Vps_GetTypes

Description:
Returns the types of VPS, based on the type of administration and disk image.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
'None'

Dedicated_ForVps_Vps_GetAll

Description:
It returns the VPS list of the dedicated server.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
'None'

Dedicated_ForVps_GetTicketStatus

Description:
Returns the status information about creating, updating and modifying VPS in a dedicated server.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
ticketId (int): Ticket ID. The parameter is required. Parameter value must be a valid integer.

Response:
type: array
Example:
'None'

Dedicated_Nfs_Get

Description:
Returns NFS server name
Version: 1.0

Parameters:
serverName (string): Server name The parameter is required.

Response:
type: struct
Example:
'None'

Dedicated_MailReport_Get

Description:
Returns consumption email report status.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('periodicity' => 'never',
'emailTo' => '',
)

Dedicated_MailReport_SetOff

Description:
Deactivate consumption email report.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Dedicated_MailReport_Set

Description:
Set consumption email report.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
emailTo (string): Destination emails separated with commas. The parameter is required. Parameter value must be a valid email address.
periodicity (string): Report periodicity. The parameter is required. Accepted parameter values: "daily","weekly","monthly".

Response:
type: void

Dedicated_GetDataTransferInfo

Description:
Returns data transfer summary.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_GetUsedSpace

Description:
Returns used space.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('total_space' => 12,
'space_limit' => 0,
'homes_space' => 0,

'web_space' => 12.287109375,
'webmail_space' => 0,
'bds_space' => 0.095703125,
'mail_space' => 0,
'ftp_space' => 0,
'others_space' => 0,
)

Dedicated_GetServerInfo

Description:
Returns dedicated server info.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_GetBandwidthConsumption

Description:
Returns bandwidth consumption summary.
Only for non-managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
month (int): Month number. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 12 inclusive.
year (int): Year. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 2002 and 2012 inclusive.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_HostingSetup_Delete

Description:
Deletes a existing user custom hosting setup.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

Response:
type: void

Dedicated_HostingSetup_Get

Description:
Returns an user custom hosting setup.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_HostingSetup_Update

Description:
Updates an existing user custom hosting setup.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.
options (struct): New setup options (as returned by Dedicated_HostingSetup_Get). The parameter is required.

Response:
type: void

Dedicated_HostingSetup_GetCustomSetups

Description:
Returns all user custom hosting setups.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_HostingSetup_GetDhSetups

Description:
Returns all DH default hosting setups.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_HostingSetup_GetOptions

Description:
Returns all available hosting setup options.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_HostingSetup_Create

Description:
Creates a new user custom hosting setup.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.
options (struct): Setup options (as returned by Dedicated_HostingSetup_Get). The parameter is required.

Response:
type: void

Dedicated_HostingSetup_GetAll

Description:
Returns all available hosting setups.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_GetSpaceAndTransferReportPerHosting

Description:
Returns a summary of the space and transfer data per hosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
'None'

Dedicated_Monitorization_Get

Description:
Returns monitorization configuration.
Only for non-managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('emails' => array(''),
'activated' => false,
)

Dedicated_Monitorization_Activate

Description:
Activate monitorization.
Only for non-managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
emails (array): Array of emails. The parameter is required.

Response:
type: void

Dedicated_Monitorization_Deactivate

Description:
Deactivate monitorization.
Only for non-managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Dedicated_GetTransferValue

Description:
Returns max data transfer value (MBs).
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: int
Example:
1500

Dedicated_Kvm_Get

Description:
Returns KVM configuration.
Only for non-managed dedicated servers.
1.0 -> 1.1: new keys: user, password
Version: 1.1

Parameters:
serverName (string): Server name The parameter is required.

Response:
type: struct
Example:
array('user' => '6pri',
'password' => 'yeqryr8',
'ip' => '10.10.10.10',
)

Dedicated_Kvm_Unmount

Description:
Unmount KVM.
Only for non-managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name The parameter is required.
allowed_ip (string): IP that will be granted access. If it's missing, we'll use the requesting client IP. The parameter is optional. Parameter value must be a valid IP v4 address.

Response:
type: void

Dedicated_Kvm_Mount

Description:
Mount KVM and get public IP.
Only for non-managed dedicated servers.
1.0 -> 1.1: returns mounted IP
1.1 -> 1.2: new parameter: allowed_ip
Version: 1.2

Parameters:
serverName (string): Server name The parameter is required.
allowed_ip (string): IP that will be granted access. If it's missing, we'll use the requesting client IP. The parameter is optional. Parameter value must be a valid IP v4 address.

Response:
type: string
Example:
'None'

Dedicated_Control_GetCpuLoadByHosting

Description:
Returns cpu load by hosting information.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: array
Example:
array(array('hosting' => 'others',
'used_cpu' => 100,
'percent_used_cpu' => 100,
))

Dedicated_Control_GetMysqlProcesses

Description:
Returns mysql active processes information.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: array
Example:
array(array('info' => 'show',
'id' => 6,
'host' => 'gdc.dinaserver.com:37883',
'ddbb' => 'NULL',
'user' => 'root',
'command' => 'Query',
'time' => 'NULL',
'status' => 'NULL',
))

Dedicated_Control_GetWebserverStatus

Description:
Returns webserver information.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_Control_GetSystemStatus

Description:
Returns sysyem information.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_Control_KillMysqlProcess

Description:
Kill mysql active processes .
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
pid (int): Process id. The parameter is required. Parameter value must be a valid integer.

Response:
type: void

Dedicated_Control_GetApacheProcesses

Description:
Returns apache active processes information.
Only for linux managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)

Dedicated_Control_GetSystemLoad

Description:
Returns cpu load by hosting information.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Dedicated_PowerStatus_Restart

Description:
Restart server.
Only for non-managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Dedicated_GetLicenses

Description:
Returns licenses information.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('totalLicenses' => 25,
'availableLicenses' => 19,
'usedLicenses' => 6,
)

Dedicated_Hosting_Delete

Description:
Deletes an existing hosting.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: void

Dedicated_Hosting_ChangeSetup

Description:
Changes setup of a existing hosting.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.
setupName (string): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

Response:
type: void

Dedicated_Hosting_Activate

Description:
Activates an existing but inactive hosting.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: void

Dedicated_Hosting_GetSetupName

Description:
Returns hosting setup name.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: string
Example:
'None'

Dedicated_Hosting_Deactivate

Description:
Deactivates an existing and active hosting.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: void

Dedicated_Hosting_GetSpace

Description:
Returns used hosting space.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: struct
Example:
array('total_space' => 2,

'space_limit' => 2000,
'homes_space' => 0,
'web_space' => 1.537109375,
'webmail_space' => 0,
'bds_space' => 0.0478515625,
'mail_space' => 0,
'ftp_space' => 0,
'others_space' => 0,
)

Dedicated_Hosting_Create

Description:
Creates a new hosting.
Only for managed dedicated servers.
1.0 -> 1.1: Added an existing setup name validation.
Version: 1.1

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.
setupName (string): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.
setZones (bool): True to set new zones, false to leave them unchanged. Default: true. The parameter is required. Parameter value must be a boolean (true or false).

Response:
type: void

Dedicated_Hosting_GetAll

Description:
Returns existing hostings on dedicated.
Only for managed dedicated servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
onlyActiveHostings (bool): True to add filter: return only active hostings. False or empty, execute without filter. The parameter is optional. Parameter value must be a boolean (true or false).

Response:
type: array
Example:
array('probando2.shiai.es','probandohosting.com','aaaaaaaa.es','bbbbb.com','154fdsa1f.com','pruebahostingdw27.com')

Dedicated_GetMonitorizedPorts

Description:
Returns monitorized ports with their status.
Only for unmanaged servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)

Domain commands

Domain_CheckForTransfer

Description:
Check if a domain name is available for transfer to Dinahosting.
Returns true if domain is registered but not in Dinahosting.
This doesn't guarantee a successful transfer.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: bool
Example:
false

Domain_Transfer_SetAuthcode

Description:
Set new authcode for incoming domain transfer.
Changes won't take effect until a new transfer petition takes place.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
newAuthcode (string): New authcode. The parameter is required.

Response:
type: void

Domain_Transfer_Restart

Description:
Restarts a domain transfer.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: void

Domain_Transfer_Cancel

Description:
Cancels a domain transfer.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: void

Domain_Transfer_GetAuthcode

Description:
Returns authcode used for incoming domain transfer.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: string
Example:
'None'

Domain_Transfer_GetStatus

Description:
Returns status of incoming domain transfer.
On transfer codes: 'WAIT_INIT','WAIT_APROB','WAIT_AUTH','WAIT_TRANSFER'
Error codes: 'KO_AUTH','FAILED','BLOCKED','REJECTED','CANCELED','DELETED','UNKNOWN'
OK codes: 'ACTIVE'
1.0 -> 1.1 Transfer codes and simulation mode.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: string
Example:
'None'

Domain_Test

Description:
Domain command Testing
Version: 1.0

Parameters:
domain (string): Domain name, for testing purposes. The parameter is required.
optional (string): An optional parameter. The parameter is optional.

Response:
type: struct
Example:
'None'

Domain_CheckForRegister

Description:
Check if a domain name is available for register on Dinahosting.
Returns true if domain is not registered and domain is not currently in activation process on Dinahosting.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: bool
Example:
false

Domain_Zone_GetTypeURL301

Description:
Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
array(array('destinationUrl' => 'http://aaa.es',
'hostname' => 'ns',
))

Domain_Zone_UpdateTypeA

Description:
Update an existing type A zone.
This command doesn't work with .tel domains.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
ip (string): New IP v4 Address. The parameter is required. Parameter length must be between 0 and 15 characters.Parameter value must be a valid IP v4 address.
oldIp (string): Old IP v4 Address. The parameter is optional.

Response:
type: void

Domain_Zone_Delete

Description:
Deletes a domain zone (any type except A, AAAA and SRV).

Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Zone_AddTypeSRV

Description:
Adds a new type SRV zone (service zone).
This command doesn't work with .tel domains.
1.0 -> 1.1: Improved parameter validation and documentation.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
service (string): Service. The parameter is required. Accepted parameter values: "http","ftp","scp","https","tftpd","smtp","imap","pop3","imaps","pop3s","named","ldap","ssh","dhcp","radius","sip","sipfederationtls","xmpp-server","xmpp-client","jabber","autodiscover","sipinternaltls".
protocol (string): Protocol. The parameter is required. Accepted parameter values: "tcp","udp","tls".
priority (int): Priority (integer value). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 100 inclusive.
weight (int): Weight (integer value) used when same priority. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 100 inclusive.
port (int): Destination port number (integer value). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 65535 inclusive.
destination (string): Destination domain . The parameter is required.

Response:
type: void

Domain_Zone_AddTypeFrame

Description:
Adds a new type FRAME zone (frame redirection).
This command doesn't work with .tel domains.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Origin hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
destinationUrl (string): Destination URL. The parameter is required. Parameter value must be a valid URL.

Response:
type: void

Domain_Zone_AddTypeAAAA

Description:
Adds a new type AAAA zone.
This command doesn't work with .tel domains.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
ip (string): IP v6 Address. The parameter is required. Parameter length must be between 0 and 39 characters.Parameter value must be a valid IP v6 address.

Response:
type: void

Domain_Zone_DeleteTypeA

Description:
Deletes a domain A type zone.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
ip (string): IP v4 Address. The parameter is required. Parameter length must be between 0 and 15 characters.Parameter value must be a valid IP v4 address.

Response:
type: void

Domain_Zone_DeleteTypeSRV

Description:
Deletes a domain SRV type zone.
1.0 -> 1.1: Major bugfix.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
service (string): Service. The parameter is required. Accepted parameter values: "http","ftp","scp","https","tftpd","smtp","imap","pop3","imaps","pop3s","named","ldap","ssh","dhcp","radius","sip","sipfederationtls","xmpp-server","xmpp-client","jabber","autodiscover","sipinternaltls".
protocol (string): Protocol. The parameter is required. Accepted parameter values: "tcp","udp","tls".
priority (int): Priority (integer value). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 100 inclusive.
weight (int): Weight (integer value) used when same priority. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 100 inclusive.
port (int): Destination port number (integer value). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 65535 inclusive.
destination (string): Destination domain . The parameter is required.

Response:
type: void

Domain_Zone_DeleteTypeURL301

Description:
Deletes a domain URL 301 type zone.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. Set empty for deleting an entire domain redirect. The parameter is optional. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Zone_DeleteTypeFrame

Description:
Deletes a domain FRAME type zone.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. Set empty for deleting an entire domain redirect. The parameter is optional. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Zone_AddTypeURL301

Description:
Adds a new type URL zone (standard redirection).
This command doesn't work with .tel domains.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Origin hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
destinationUrl (string): Destination URL. The parameter is required. Parameter value must be a valid URL.

Response:
type: void

Domain_Zone_GetTypeFrame

Description:
Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
array(array('destinationUrl' => 'http://www.dinahosting.com',
'hostname' => '@',
),array('destinationUrl' => 'http://www.dinahosting.com',
'hostname' => 'www',
))

Domain_Zone_DeleteTypeURL

Description:
Deletes a domain URL type zone.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. Set empty for deleting an entire domain redirect. The parameter is optional. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Zone_DeleteTypeMX

Description:
Deletes a domain MX type zone.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
host (string): Host. The parameter is required.
address (string): Address. The parameter is required. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Zone_AddTypeURL

Description:
Adds a new type URL zone (standard redirection).
This command doesn't work with .tel domains.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Origin hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
destinationUrl (string): Destination URL. The parameter is required. Parameter value must be a valid URL.

Response:

type: void

Domain_Zone_DeleteTypeTXT

Description:
Deletes a domain TXT type zone.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
value (string): Value (optional). For specifying what to delete if there are several zones under the same hostname. The parameter is optional.

Response:
type: void

Domain_Zone_AddTypeMXDetailedEmail

Description:
Adds a new type MX zone.
This command doesn't work with .tel domains.
The MX zone requires that a A zone already exists.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
mxNumber (int): Mx number. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 7 inclusive.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Zone_DeleteTypeAname

Description:
Deletes a domain ANAME type zone.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: void

Domain_Zone_AddTypeMXS

Description:
Adds a new type MX simple zone.

This command doesn't work with .tel domains.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
ip (string): IP v4 Address. The parameter is required. Parameter length must be between 0 and 15 characters.Parameter value must be a valid IP v4 address.

Response:
type: void

Domain_Zone_AddTypeTXT

Description:
Adds a new type TXT zone .
This command doesn't work with .tel domains.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Origin hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
text (string): Text. The parameter is required.

Response:
type: void

Domain_Zone_GetTypeURL

Description:
Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
array(array('destinationUrl' => 'http://www.example.com/test',
'hostname' => 'test',
))

Domain_Zone_UpdateTypeAname

Description:
Update an existing type ANAME zone.
Version: 1.0

Parameters:

domain (string): Domain name. The parameter is required.
destinationHostname (string): Destination hostname. The parameter is required.
oldDestinationHostname (string): Old destination hostname. The parameter is required.

Response:
type: void

Domain_Zone_DeleteTypeCname

Description:
Deletes a domain CNAME type zone.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Zone_GetTypeAname

Description:
Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
array(array('destinationHostname' => 'google.com',
'hostname' => '@',
))

Domain_Zone_AddTypeMX

Description:
Adds a new type MX zone.
This command doesn't work with .tel domains.
The MX zone requires that a A zone already exists.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
host (string): Host (empty for default host: @). Required for ip values. The parameter is optional. Parameter length must be between 0 and 64 characters.
priority (int): Priority. The parameter is required. Parameter value must be a valid integer. Accepted parameter values: "1","5","10","15","20","25","30","40","50".
value (string): Value (servername or ip). The parameter is required.

Response:
type: void

Domain_Zone_GetTypeSRV

Description:
Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
array(array('service' => 'http',
'protocol' => 'tcp',
'priority' => 10,
'weight' => 70,
'port' => 8080,
'destination' => 'comooor.es',
),array('service' => 'http',
'protocol' => 'tcp',
'priority' => 10,
'weight' => 30,
'port' => 80,
'destination' => 'comor.es',
))

Domain_Zone_UrlShortener_Delete

Description:
Deletes a shortened URL on a subdomain
Version: 1.0

Parameters:
domain (string): Domain. The parameter is required.
keyword (string): Keyword of the shortened URL to be deleted. The parameter is required.

Response:
type: void

Domain_Zone_UrlShortener_Activate

Description:
Activates shortener on a subdomain.
Version: 1.0

Parameters:
domain (string): Domain. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Zone_UrlShortener_Deactivate

Description:
Deactivates shortener on a subdomain.
Use with caution: it deletes all the shortened URLs created under the subdomain.
Version: 1.0

Parameters:
domain (string): Domain. The parameter is required.

Response:
type: void

Domain_Zone_UrlShortener_Create

Description:
Creates a new shortened URL on a subdomain.
Version: 1.0

Parameters:
domain (string): Domain. The parameter is required.
url (string): URL to be shorten. The parameter is required. Parameter value must be a valid URL.
title (string): URL title. The parameter is optional.

Response:
type: void

Domain_Zone_UrlShortener_GetAll

Description:
Returns shortened URLs on a subdomain.
Version: 1.0

Parameters:
domain (string): Domain. The parameter is required.

Response:
type: struct
Example:
'None'

Domain_Zone_UpdateTypeAAAA

Description:
Update an existing type AAAA zone.
This command doesn't work with .tel domains.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
ip (string): New IP v6 Address. The parameter is required. Parameter length must be between 0 and 39 characters.Parameter value must be a valid IP v6 address.
oldIp (string): Old IP v6 Address. The parameter is optional.

Response:
type: void

Domain_Zone_AddTypeCname

Description:
Adds a new type CNAME zone.
This command doesn't work with .tel domains.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Origin hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
destinationHostname (string): Destination hostname. The parameter is required.

Response:
type: void

Domain_Zone_GetTypeA

Description:
Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)

Domain_Zone_GetTypeTXT

Description:
Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:

array(array('host' => 'key_9344._domainkey',
'text' => 'Address',
))

Domain_Zone_DeleteTypeAAAA

Description:
Deletes a domain AAAA type zone.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
ip (string): IP v6 Address. The parameter is required. Parameter length must be between 0 and 39 characters.Parameter value must be a valid IP v6 address.

Response:
type: void

Domain_Zone_GetTypeAAAA

Description:
Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
array(array('ip' => '2001:470:0:76:0:0:0:2',
'hostname' => 'trescuatro',
))

Domain_Zone_AddTypeAname

Description:
Adds a new type ANAME zone to the root domain.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
destinationHostname (string): Destination hostname. The parameter is required.

Response:
type: void

Domain_Zone_GetTypeMX

Description:

Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
array(array('host' => 'SOA',
'type' => 'MX',
'address' => 'mx4.mepegotolafiesta.com.',
))

Domain_Zone_DeleteTypeSPF

Description:
Deletes a domain SPF type zone.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
value (string): Value (optional). For specifying what to delete if there are several zones under the same hostname. The parameter is optional.

Response:
type: void

Domain_Zone_GetTypeSPF

Description:
Returns domain zones.
This command has pending some modifications: Make response data format more friendly.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
array(array('address' => 'no|||',
))

Domain_Zone_DeleteAll

Description:
Deletes domain zones of a given type.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
type (string): Zone type. The parameter is required. Accepted parameter values: "A","AAAA","CNAME","frame","URL","URL_301","TXT","SRV","SPF","MX".

Response:
type: void

Domain_Zone_GetTypeCname

Description:
Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
array(array('destinationHostname' => 'google.com',
'hostname' => 'preproduccion',
))

Domain_Zone_AddTypeA

Description:
Adds a new type A zone.
This command doesn't work with .tel domains.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.
ip (string): IP v4 Address. The parameter is required. Parameter length must be between 0 and 15 characters.Parameter value must be a valid IP v4 address.

Response:
type: void

Domain_Zone_GetAll

Description:
Returns domain zones.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
orderBy (string): Ordenation types: no ordenation, by type, by hostname. The parameter is optional. Accepted parameter values: "none","type","hostname".

Response:

type: array
Example:
(Example too long, see web.)

Domain_GetRegistrationDate

Description:
Returns registration date of domain.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: string
Example:
'2003-02-10'

Domain_Status_Get

Description:
Returns status of domain.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: string
Example:
'clientDeleteProhibited, clientTransferProhibited'

Domain_Status_IsLocked

Description:
This commands checks domain status and returns boolean true if domain is currently locked.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: bool
Example:
true

Domain_Status_Unlock

Description:
Unlocks domain status.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: void

Domain_Status_Lock

Description:
Locks domain status.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: void

Domain_Redirection_Delete

Description:
Delete URL redirection over whole domain.
1.0 -> 1.1: Optionally now can remove any subdomain redirections.
1.1-> 1.2: Optionally now can remove specific subdomain redirections.
Version: 1.2

Parameters:
domain (string): Domain name. The parameter is required.
removeAll (bool): Set true for deleting any subdomain redirection zones. The parameter is optional. Parameter value must be a boolean (true or false).
subdomain (?string): Subdomain (restricted: www, mail, ftp, mysql, sql, autoconfig, autodiscover, *, @). The parameter is optional.

Response:
type: void

Domain_Redirection_Get

Description:
Returns current URL redirection on whole domain, if any.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: struct
Example:
array('Type' => 'FRAME',
'Url' => 'http://www.dinahosting.com',
'Title' => '',
'Description' => '',

'Keywords' => '',
)

Domain_Redirection_SetStd

Description:
Sets Standard URL Redirecion on whole domain or domain zone.
1.0 -> 1.1: Optionally now can add a subdomain redirection.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
url (string): URL redirection destination. The parameter is required. Parameter value must be a valid URL.
host (string): Subdomain hostname. The parameter is optional. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Redirection_SetFrameSimple

Description:
Sets Frame Redirecion on whole domain or domain zone.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
url (string): URL redirection destination. The parameter is required. Parameter value must be a valid URL.
host (string): Subdomain hostname. The parameter is optional. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Redirection_SetFrame

Description:
Sets Frame URL Redirecion on whole domain.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
url (string): URL redirection framed destination. The parameter is required. Parameter value must be a valid URL.
title (string): Frame title. The parameter is optional. Parameter length must be between 0 and 255 characters.
description (string): Frame description meta. The parameter is optional. Parameter length must be between 0 and 255 characters.
keywords (string): Frame keywords meta. The parameter is optional. Parameter length must be between 0 and 255 characters.
favicon (string): Frame favicon URL. The parameter is optional. Parameter length must be between 0 and 255 characters.
viewport (string): Frame viewport meta. The parameter is optional. Parameter length must be between 0

and 255 characters.

Response:
type: void

Domain_Redirection_Set301

Description:
Sets 301 URL Redirecion on whole domain or domain zone.
1.0 -> 1.1: Optionally now can add a subdomain redirection.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
url (string): URL redirection destination. The parameter is required. Parameter value must be a valid URL.
host (string): Subdomain hostname. The parameter is optional. Parameter length must be between 0 and 64 characters.

Response:
type: void

Domain_Redirection_GetAll

Description:
Returns all URL redirections on whole domain and domain zone redirections, if any.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)

Domain_PanelIndependentAccess_SetPassword

Description:
Set a new password for domain panel independent access.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
newPassword (string): New password. The parameter is required.

Response:
type: void

Domain_Contacts_Get

Description:
Returns array with contacts of domain. Only returns contacts of one type (example: registrant).

1.0 -> 1.1: Added Fax key.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
contactType (string): Contact type (registrant, admin, billing or technical). The parameter is required. Accepted parameter values: "registrant","admin","billing","tech".

Response:
type: struct
Example:
(Example too long, see web.)

Domain_Contacts_GetRegistrantVerificationInfo

Description:
Gets registrant verification info.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: struct
Example:
'None'

Domain_Contacts_SetAll

Description:
Sets new contacts of domain.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
contacts (struct): Array containing contacts (see Domain_Contacts_GetAll). The parameter is required. Parameter value must be a valid contacts.

Response:
type: void

Domain_Contacts_Set

Description:
Sets new contacts of domain. Only updates one type of contacts (example: billing).
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.
contactType (string): Contact type (registrant, admin, billing or technical). The parameter is required. Accepted parameter values: "registrant","admin","billing","tech".
contacts (struct): Array containing contacts (see Domain_Contacts_Get for format). The parameter is

required. Parameter value must be a valid contacts.

Response:
type: void

Domain_Contacts_GetAll

Description:
Returns array with contacts of domain.
1.0 -> 1.1: Added Fax key.
Version: 1.1

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Domain_GetRenovationDate

Description:
Returns expected domain deactivation date.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: string
Example:
'2013-07-06'

Domain_GetExpirationDate

Description:
Returns registry expiration date of domain.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: string
Example:
'2014-02-10'

Domain_Dnss_Get

Description:
Returns array with DNS of domain.

Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: struct
Example:
array('Dns1' => 'ns2.gestiondecuenta.com',
'Dns2' => 'ns.gestiondecuenta.com',
)

Domain_Dnss_Set

Description:
Sets new DNS of domain.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
dnss (array): Array containing DNS strings. The parameter is required. Parameter value must be a valid array of 2-8 unique DNS.

Response:
type: void

Domain_UrlShortener_Delete

Description:
Deletes a shortened URL on a domain
Version: 1.0

Parameters:
domain (string): Domain. The parameter is required.
keyword (string): Keyword of the shortened URL to be deleted. The parameter is required.

Response:
type: void

Domain_UrlShortener_Activate

Description:
Activates shortener on a domain.
Version: 1.0

Parameters:
domain (string): Domain. The parameter is required.

Response:
type: void

Domain_UrlShortener_Deactivate

Description:
Deactivates shortener on a domain.
Use with caution: it deletes all the shortened URLs created under the domain.
Version: 1.0

Parameters:
domain (string): Domain. The parameter is required.

Response:
type: void

Domain_UrlShortener_Create

Description:
Creates a new shortened URL on a domain.
Use with caution: shortened URLs on a domain make traffic being redirected to the shortener.
Version: 1.0

Parameters:
domain (string): Domain. The parameter is required.
url (string): URL to be shorten. The parameter is required. Parameter value must be a valid URL.
title (string): URL title. The parameter is optional.

Response:
type: void

Domain_UrlShortener_GetAll

Description:
Returns shortened URLs on a domain.
Version: 1.0

Parameters:
domain (string): Domain. The parameter is required.

Response:
type: struct
Example:
'None'

Domain_GetAuthcode

Description:
Returns authcode of domain for outgoing transfer.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:

type: string
Example:
'BqDCrmqJ788}`~'*

Domain_GetInfoDomain

Description:
Returns registry info of domain.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Domain_WhoisPrivacy_Get

Description:
Returns domain whois privacy configuration.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: struct
Example:
array('activated' => true,
'redirection' => 'informedk_org562@privacidadwhois.com',
)

Domain_WhoisPrivacy_Set

Description:
Configures domain whois privacy.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
redirectionEmail (string): Email direction where received emails are sent to. The parameter is optional. Parameter value must be a valid email address.
enable (boolean): Enables or disables whois protection. The parameter is optional. Parameter value must be a boolean (true or false).

Response:
type: void

Domain_Parking_Get

Description:
Returns parking info.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: struct
Example:
array('Title' => 'Mi página en parking',
'Description' => 'Estas es la web de dinahosting',
)

Domain_Parking_Set

Description:
Sets domain on parking.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
title (string): Title for parking webpage. The parameter is optional. Parameter length must be between 0 and 254 characters.
description (string): Description for parking webpage. The parameter is optional.

Response:
type: void

Domain_NameServer_Delete

Description:
Delete a DNS server over domain.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required.

Response:
type: void

Domain_NameServer_Register

Description:
Register a DNS server over domain.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required.
ip (string): IP Address. The parameter is required. Parameter value must be a valid IP address (v4 or v6).

Keep in mind that v6 is currently experimental, might not work.

Response:
type: void

Domain_NameServer_Modify

Description:
Modify existing DNS server over domain.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.
hostname (string): Hostname. The parameter is required.
ip (string): IP Address. The parameter is required. Parameter value must be a valid IP address (v4 or v6). Keep in mind that v6 is currently experimental, might not work.

Response:
type: void

Domain_NameServer_GetAll

Description:
Returns array with registered DNS server names and IPs over domain.
Version: 1.0

Parameters:
domain (string): Domain name. The parameter is required.

Response:
type: array
Example:
array(array('name' => 'ns2.afiliacionweb.com',
'ip' => '85.112.29.233',
),array('name' => 'ns1.afiliacionweb.com',
'ip' => '85.112.29.232',
))

DomainVerification commands

DomainVerification_GetPending

Description:
Returns all pending domain verifications.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

Hosting commands

Hosting_Test

Description:
Hosting command Testing
Version: 1.0

Parameters:
hosting (string): Domain name, for testing purposes. The parameter is required.
optional (string): Just an optional parameter The parameter is optional.

Response:
type: struct
Example:
'None'

Hosting_Ddbb_MariaDb_Delete

Description:
Delete selected MariaDb Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
userName (string): User name. The parameter is required. Parameter length must be between 1 and 16 characters.

Response:
type: void

Hosting_Ddbb_MariaDb_SetPassword

Description:
Set new password for selected MariaDb Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
userName (string): User name. The parameter is required. Parameter length must be between 1 and 16 characters.
userPassword (string): New password. The parameter is required.

Response:
type: void

Hosting_Ddbb_MariaDb_Create

Description:
Creates a new MariaDb DataBase.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
userName (string): User name. The parameter is required. Parameter length must be between 1 and 16 characters.
userPassword (string): User password. The parameter is required. Parameter length must be between 6 and 71 characters.

Response:
type: void

Hosting_Ddbb_MariaDb_GetAll

Description:
Returns all MariaDb databases.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
'None'

Hosting_Ddbb_Postgre_Delete

Description:
Delete selected Postgre Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

Response:
type: void

Hosting_Ddbb_Postgre_SetPassword

Description:
Set new password for selected Postgre Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
userName (string): User name. The parameter is required. Parameter length must be between 1 and 16 characters.
userPassword (string): New password. The parameter is required.

Response:
type: void

Hosting_Ddbb_Postgre_Create

Description:
Creates a new Postgre DataBase.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
userName (string): User name. The parameter is required. Parameter length must be between 1 and 16 characters.
userPassword (string): User password. The parameter is required. Parameter length must be between 6 and 71 characters.

Response:
type: void

Hosting_Ddbb_Postgre_GetAll

Description:
Returns all Postgre databases.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
array(array('server' => 'pgsql01.dinaserver.com',
'account' => 'userpostgresiphone',
'name' => 'postgreiphone',
'autoinstall' => false,
'date' => '2011-01-31 14:25:08',
))

Hosting_Ddbb_Mysql_SetHostAcl

Description:
Sets selected Mysql Database hostname ACL configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
hostname (string): Accepted hostname or "any" string (without quotes). The parameter is required.

Response:
type: void

Hosting_Ddbb_Mysql_Delete

Description:
Delete selected Mysql Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

Response:
type: void

Hosting_Ddbb_Mysql_GetAclConfiguration

Description:
Returns selected Mysql Database ACL configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

Response:
type: struct
Example:
array('host' => 'ANY',
)

Hosting_Ddbb_Mysql_CreateWithIpAcl

Description:
Creates a new Mysql DataBase with IP ACL.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
userName (string): User name. The parameter is required. Parameter length must be between 1 and 16 characters.
userPassword (string): User password. The parameter is required. Parameter length must be between 6 and 16 characters.
accessHostIp (string): Accepted IP v4 . The parameter is required. Parameter value must be a valid IP v4 address.

Response:
type: void

Hosting_Ddbb_Mysql_SetPassword

Description:
Set new password for selected Mysql Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
password (string): New password. The parameter is required.

Response:
type: void

Hosting_Ddbb_Mysql_BackupTask_Delete

Description:
Deletes an existing Backup Task for a selected Mysql dataBase.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

Response:
type: void

Hosting_Ddbb_Mysql_BackupTask_Set

Description:
Sets the Backup Task for a selected Mysql dataBase.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
taskPeriod (string): Task Period (##d for days, ##h for hours or ##m for minutes). The parameter is required. Parameter value must be a valid task period (##d ##m ##h).

Response:
type: void

Hosting_Ddbb_Mysql_BackupTask_GetAll

Description:
Returns all Mysql backup tasks on this hosting.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
array('oscommerce_47aa503883f33' => '1m',
)

Hosting_Ddbb_Mysql_GetInfo

Description:
Returns selected Mysql Database information.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

Response:
type: struct
Example:
array('server' => 'hl77.dinaserver.com',
'name' => 'test2BBDD',
'account' => 'test2',
'date' => '2009-08-26 22:14:15',
'autoinstall' => false,
)

Hosting_Ddbb_Mysql_SetIpAcl

Description:
Sets selected Mysql Database hostname ACL configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
ip (string): Accepted IP v4 . The parameter is required. Parameter value must be a valid IP v4 address.

Response:
type: void

Hosting_Ddbb_Mysql_Backup_Get

Description:
Return an existing Backup Task for a selected Mysql dataBase.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

Response:
type: struct
Example:
(Example too long, see web.)

Hosting_Ddbb_Mysql_Backup_Restore

Description:
Restore an existing Backup Task for a selected Mysql dataBase.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
file (string): Backup file The parameter is required.

Response:
type: void

Hosting_Ddbb_Mysql_Create

Description:
Creates a new Mysql DataBase.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database

name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
userName (string): User name. The parameter is required. Parameter length must be between 6 and 16 characters.
userPassword (string): User password. The parameter is required. Parameter length must be between 1 and 41 characters.
accessHost (string): Access host. Default: any. The parameter is required. Accepted parameter values: "any","localhost".

Response:
type: void

Hosting_Ddbb_Mysql_GetAll

Description:
Returns all Mysql databases.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)

Hosting_Ddbb_Access_GetAllMdb

Description:
Returns all Access .mdb databases .
Warning: This won't work with accdb databases (access 2007+).
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
array('1ruebaRomina.mdb','1ruebaRomina2.mdb','1ruebaRuben.mdb','1ruebaRuben2.mdb')

Hosting_Ddbb_Sqlserver_Delete

Description:
Delete selected Sqlserver Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

Response:
type: void

Hosting_Ddbb_Sqlserver_GetSpaceInfo

Description:
Returns a sqlserver databases space report.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Hosting_Ddbb_Sqlserver_SetPassword

Description:
Set new password for selected Sqlserver Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
password (string): New password. The parameter is required.

Response:
type: void

Hosting_Ddbb_Sqlserver_SetTransactionsSize

Description:
Set transaction size for selected Sqlserver Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
transactionsSize (int): Maximum data volume transmitted (MB). The parameter is required. Parameter value must be greater than 0.

Response:
type: void

Hosting_Ddbb_Sqlserver_GetGlobalSpaceInfo

Description:
It gets the space info of SQL Server formatted in MB: space limit and available space.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
array('limit' => 200,
'used' => 0,
'reserved' => 0,
'available' => 200,
)

Hosting_Ddbb_Sqlserver_SetDataSize

Description:
Set data size for selected Sqlserver Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
dataSize (int): Maximum data volume stored (MB). The parameter is required. Parameter value must be greater than 2.

Response:
type: void

Hosting_Ddbb_Sqlserver_Create

Description:
Creates a new Sqlserver DataBase.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
userName (string): User name. The parameter is required. Parameter length must be between 1 and 16 characters.
userPassword (string): User password. The parameter is required. Parameter length must be between 6 and 71 characters.Parameter value must be an alphanumeric string.
dataSize (int): Maximum data volume stored (MB). The parameter is required. Parameter value must be greater than 4.
transactionsSize (int): Maximum data volume transmitted (MB). The parameter is required. Parameter value must be greater than 0.
accessFrom (string): Access from: Any location (default), Localhost and specific IP (XX.XX.XX.XX), Only localhost (localhost). The parameter is optional.

Response:
type: void

Hosting_Ddbb_Sqlserver_GetAll

Description:
Returns all SqlServer databases.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
array(array('name' => 'unabbdd',
'account' => 'unusuario',
'password' => '?????????',
'date' => '2011-07-13 17:41:03',
'data_space' => 3,
'transaction_space' => 1,
))

Hosting_Ddbb_MongoDb_Delete

Description:
Delete selected MongoDb Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

Response:
type: void

Hosting_Ddbb_MongoDb_SetPassword

Description:
Set new password for selected MongoDb Database.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
userName (string): User name. The parameter is required. Parameter length must be between 1 and 16 characters.
userPassword (string): New password. The parameter is required.

Response:
type: void

Hosting_Ddbb_MongoDb_GetAll

Description:
Returns all MongoDb databases.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
'None'

Hosting_Ddbb_GetAll

Description:
Returns all databases.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
array(array('name' => 'kbsoporte',
'account' => 'kbsoporte',
'date' => '2012-11-05 20:59:35',
'autoinstall' => false,
'type' => 'mysql',
'access_host' => 'any',
))

Hosting_Import_ListAll

Description:
List all imports
Version: 1.0

Parameters:
hosting (string): Destination hosting name. The parameter is required.

Response:
type: array
Example:
'None'

Hosting_Import_List

Description:
List unfinished (PENDING or STARTED) imports.
Version: 1.0

Parameters:
hosting (string): Destination hosting name. The parameter is required.

Response:
type: array
Example:
'None'

Hosting_Import_ApplyForEmailAccountImap

Description:
Apply for an email import through IMAP to your hosting.
1.0 -> 1.1: Removed unused parameter.
1.1 -> 1.2: Delegated all operations to async ticket execution (incl. mail account creation).
Please note this command has a delayed execution and it'll return SUCCESS_PENDING code on success and a ticket id (integer) on the data field.Version: 1.2

Parameters:
hosting (string): Destination hosting name. The parameter is required.
originEmailAddress (string): Origin full email address. The parameter is required. Parameter value must be a valid email address.
originPassword (string): Origin user password. The parameter is required. Parameter length must be between 1 and 255 characters.
originServer (string): Origin server hostname or IP with (or without) port. Default port: 143. The parameter is required.
create (bool): Create destination email account. The parameter is optional. Parameter value must be a boolean (true or false).
destinationUser (string): Destination new email account alias (without hostname). If it's empty, we'll use originEmailAddress account alias as the destination name. The parameter is optional. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
destinationPassword (string): Only used if create is true. Destination user password. If it's empty, we'll use originPassword as the destination password on creation. The parameter is optional. Parameter length must be between 0 and 16 characters.
notificationMail (string): Email address for a notification. The parameter is optional. Parameter value must be a valid email address.

Response:
type: int
Example:
'None'

Hosting_Import_ApplyForFtp

Description:
Imports files from an external FTP server into a Dinahosting hosting account.
Notes:
- The source FTP server must be hosted on a different server than the destination hosting account.
- The source FTP account must have permission to read all files and directories to be imported.
- The destination FTP account must have permission to write to its assigned home directory.
- If some files or directories cannot be accessed due to permission restrictions, they may be skipped or the

import may fail.
- Symbolic links and server-specific filesystem features may not be preserved during the import.
- When originDirectory is not specified, the import starts from the source FTP account's default home directory.
1.0 -> 1.1: Removed unused parameter.
1.1 -> 1.2: Parameters "create" and "destinationPassword" are now deprecated.
Please note this command has a delayed execution and it'll return SUCCESS_PENDING code on success and a ticket id (integer) on the data field.Version: 1.2

Parameters:
hosting (string): Name of the destination Dinahosting hosting account where the files will be imported. The parameter is required.
originUser (string): Username of the source FTP account. This user must have read access to all files and directories to be imported. The parameter is required. Parameter length must be between 1 and 35 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
originPassword (string): Password of the source FTP account. The parameter is required. Parameter length must be between 3 and 35 characters.
originServer (string): Hostname of the source FTP server. The connection uses port 21. The source server must be different from the server where the destination hosting is located. The parameter is required.
originDirectory (string): Source directory to import files from. If omitted, the FTP account's default home directory is used. All files and subdirectories must be accessible by originUser. The parameter is optional. Parameter length must be between 1 and 255 characters. It should be a valid unix path.
create (bool): Deprecated parameter. The parameter is optional. Parameter value must be a boolean (true or false).
destinationUser (string): Destination FTP username. Files are imported into this user's home directory (www for the main account). If empty, originUser is used and the account is created automatically if needed. The destination user must have write permission in its assigned home directory. The parameter is optional. Parameter length must be between 1 and 35 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
destinationPassword (string): Deprecated parameter. The parameter is optional.
notificationMail (string): Email address that will receive the import status notification. The parameter is optional. Parameter value must be a valid email address.

Response:
type: int
Example:
'None'

Hosting_Import_ApplyForDdbb

Description:
Apply for a database import to your hosting.
1.0 -> 1.1: Removed unused parameter.
Please note this command has a delayed execution and it'll return SUCCESS_PENDING code on success and a ticket id (integer) on the data field.Version: 1.1

Parameters:
hosting (string): Destination hosting name. The parameter is required.
originDdbbName (string): Origin database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
originUser (string): Origin user name. The parameter is required. Parameter length must be between 1 and 16 characters.
originPassword (string): Origin user password. The parameter is required. Parameter length must be between 1 and 255 characters.

originServer (string): Origin server hostname or IP with (or without) port. Default port: 3306. The parameter is required.
create (bool): Create destination DDBB. The parameter is optional. Parameter value must be a boolean (true or false).
destinationDdbbName (string): Destination new database name. If it's empty, we'll use originDdbbName as the destination database name. The parameter is optional. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
destinationUser (string): Destination new user name. If it's empty, we'll use originUser as the destination user name on creation. The parameter is optional. Parameter length must be between 0 and 16 characters.
destinationPassword (string): Only used if create is true. Destination user password. If it's empty, we'll use originPassword as the destination password on creation. The parameter is optional. Parameter length must be between 0 and 16 characters.
notificationMail (string): Email address for a notification. The parameter is optional. Parameter value must be a valid email address.

Response:
type: int
Example:
'None'

Hosting_Import_ApplyForEmailAccountPop

Description:
Apply for an email import through POP to your hosting.
1.0 -> 1.1: Removed unused parameter.
1.1 -> 1.2: Delegated all operations to async ticket execution (incl. mail account creation).
Please note this command has a delayed execution and it'll return SUCCESS_PENDING code on success and a ticket id (integer) on the data field.Version: 1.2

Parameters:
hosting (string): Destination hosting name. The parameter is required.
originEmailAddress (string): Origin full email address. The parameter is required. Parameter value must be a valid email address.
originPassword (string): Origin user password. The parameter is required. Parameter length must be between 1 and 255 characters.
originServer (string): Origin server hostname or IP with (or without) port. Default port: 110. The parameter is required.
create (bool): Create destination account. The parameter is optional. Parameter value must be a boolean (true or false).
destinationUser (string): Destination new email account alias (without hostname). If it's empty, we'll use originEmailAddress account alias as the destination name. The parameter is optional. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
destinationPassword (string): Only used if create is true. Destination user password. If it's empty, we'll use originPassword as the destination password on creation. The parameter is optional. Parameter length must be between 0 and 16 characters.
destinationFolder (string): Destination folder outside inbox. The parameter is optional. Parameter length must be between 0 and 255 characters.
notificationMail (string): Email address for a notification. The parameter is optional. Parameter value must be a valid email address.

Response:
type: int
Example:
'None'

Hosting_Dsn_Mysql_Delete

Description:
Delete selected Mysql Database Source Name.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
dsnName (string): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.

Response:
type: void

Hosting_Dsn_Mysql_Create

Description:
Creates a new Mysql Data Source Name.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
dsnName (string): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.
dsnDescription (string): Database description. The parameter is required.

Response:
type: void

Hosting_Dsn_Mysql_GetAll

Description:
Returns all Mysql database source names.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)

Hosting_Dsn_Access_Delete

Description:
Delete selected Access Database Source Name.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
dsnName (string): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.

Response:
type: void

Hosting_Dsn_Access_Create

Description:
Creates a new Access Data Source Name.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
dsnName (string): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.
dsnDescription (string): Database description. The parameter is required.

Response:
type: void

Hosting_Dsn_Access_GetAll

Description:
Returns all MS Access database source names.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)

Hosting_Dsn_Sqlserver_Delete

Description:
Delete selected Sqlserver Database Source Name.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
dsnName (string): DSN name. The parameter is required. Parameter value must be a valid database server

name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.

Response:
type: void

Hosting_Dsn_Sqlserver_Create

Description:
Creates a new Sqlserver Data Source Name.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ddbbName (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.
dsnName (string): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.
dsnDescription (string): Database description. The parameter is required.

Response:
type: void

Hosting_Dsn_Sqlserver_GetAll

Description:
Returns all SqlServer database source names.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)

Hosting_Https_Get

Description:
Get HTTPs access status.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: bool
Example:
false

Hosting_Https_Activate

Description:
Activate HTTPs access.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Https_Deactivate

Description:
Deactivate HTTPs access.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Subdomain_Delete

Description:
Delete a subdomain on hosting.
Version: 1.0

Parameters:
hosting (string): Hosting domain name. The parameter is required.
host (string): Host for subdomain name. The parameter is required. Parameter length must be between 1 and 50 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Subdomain_Create

Description:
Create a subdomain on hosting.
1.0 -> 1.1: Fix on directory path validation.
Version: 1.1

Parameters:
hosting (string): Hosting domain name. The parameter is required.
host (string): Host for subdomain name. The parameter is required. Parameter length must be between 1 and 50 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
directory (string): Subdirectory name for account files. Default value: same as host . The parameter is

optional. Parameter length must be between 1 and 255 characters. It should be a valid unix path.
ftpAccount (string): FTP account name. Default value: same as host . The parameter is optional. Parameter length must be between 1 and 50 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Subdomain_GetAll

Description:
Returns all existing subdomains on hosting.
Version: 1.0

Parameters:
hosting (string): Hosting domain name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)

Hosting_Ssh_Get

Description:
Get SSH access status.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: bool
Example:
false

Hosting_Ssh_Activate

Description:
Activate SSH access.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Ssh_Deactivate

Description:
Deactivate SSH access.

Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_IsActive

Description:
Returns if a hosting is active.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: bool
Example:
'None'

Hosting_PanelIndependentAccess_SetPassword

Description:
Set a new password for hosting panel independent access.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
newPassword (string): New password. The parameter is required.

Response:
type: void

Hosting_GetInfo

Description:
Returns hosting webserver information with traffic and space metrics.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Hosting_GetMailInfo

Description:
Returns hosting mailserver information.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
array('server' => 'vl193.dinaserver.com',
'host' => 'mail.test-edu-personalizacion-avanzada.com',
'TLS_activable' => true,
'TLS_is_active' => false,
)

Hosting_Backup_Send_Database_Tables

Description:
It sends one or several database tables from a backup to a specified email.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
databaseName (string): Database name. The parameter is required.
databaseType (string): Database type. The parameter is required. Accepted parameter values: "sqlServer","otherType".
tables (string): Database tablenames separated by comma. The parameter is required.
email (string): Email to send the backup. The parameter is required. Parameter value must be a valid email address.

Response:
type: struct
Example:
'None'

Hosting_Backup_Send_Database

Description:
It sends a database backup to a specified email.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
databaseName (string): Database name. The parameter is required.
databaseType (string): Database type. The parameter is required. Accepted parameter values: "sqlServer","otherType".
email (string): Email to send the backup. The parameter is required. Parameter value must be a valid email address.

Response:

type: struct
Example:
'None'

Hosting_Backup_Send_Resource

Description:
It sends a resource from a hosting backup to a specified email.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
resourcePath (string): Resource path The parameter is required.
email (string): Email to send the backup. The parameter is required. Parameter value must be a valid email address.

Response:
type: struct
Example:
'None'

Hosting_Backup_Database_GetTables

Description:
It returns the database tables in a hosting backup.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
databaseName (string): Database name. The parameter is required.
databaseType (string): Database type. The parameter is required. Accepted parameter values: "sqlServer","otherType".

Response:
type: struct
Example:
'None'

Hosting_Backup_Restore_Database_Tables

Description:
It restores one or several database tables from a backup.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
databaseName (string): Database name. The parameter is required.

databaseType (string): Database type. The parameter is required. Accepted parameter values: "sqlServer","otherType".
tables (string): Database tablenames separated by comma. The parameter is required.
notificationEmail (string): Email to send the notification of restoring. The parameter is required. Parameter value must be a valid email address.

Response:
type: struct
Example:
'None'

Hosting_Backup_Restore_EmailAccounts

Description:
It restores one or several email accounts from backup.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
emailAccounts (string): Email accounts separated by comma. The parameter is required.
notificationEmail (string): Email to send the notification of restoring. The parameter is required. Parameter value must be a valid email address.

Response:
type: struct
Example:
'None'

Hosting_Backup_Restore_Database

Description:
It restores a database backup.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
databaseName (string): Database name. The parameter is required.
databaseType (string): Database type. The parameter is required. Accepted parameter values: "sqlServer","otherType".
notificationEmail (string): Email to send the notification of restoring. The parameter is required. Parameter value must be a valid email address.

Response:
type: struct
Example:
'None'

Hosting_Backup_Restore_Resource

Description:

It restores a resource from a hosting backup.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
resourcePath (string): Resource path The parameter is required.
notificationEmail (string): Email to send the notification of restoring. The parameter is required. Parameter value must be a valid email address.

Response:
type: struct
Example:
'None'

Hosting_Backup_GetEmailAccounts

Description:
It returns the email accounts in a hosting backup.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

Response:
type: struct
Example:
'None'

Hosting_Backup_GetDatabases

Description:
It returns the databases in a hosting backup.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

Response:
type: struct
Example:
'None'

Hosting_Backup_GetResources

Description:
It returns the resources in a hosting backup.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
date (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
folderPath (string): If empty, it returns the root folder content. The parameter is optional.

Response:
type: struct
Example:
'None'

Hosting_Backup_GetStatus

Description:
It returns the status of the restorings or sendings programmed.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
'None'

Hosting_UrlShortener_Delete

Description:
Deletes a shortened URL on a hosting
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
keyword (string): Keyword of the shortened URL to be deleted. The parameter is required.

Response:
type: void

Hosting_UrlShortener_Activate

Description:
Activates shortener on a hosting.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
folder (string): Hosting folder to store shorten URLs. The parameter is required.

Response:
type: void

Hosting_UrlShortener_Deactivate

Description:
Deactivates shortener on a hosting.
Use with caution: it deletes all the shortened URLs created under the hosting.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_UrlShortener_Create

Description:
Creates a new shortened URL on a hosting.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
url (string): URL to be shorten. The parameter is required. Parameter value must be a valid URL.
title (string): URL title. The parameter is optional.

Response:
type: void

Hosting_UrlShortener_GetAll

Description:
Returns shortened URLs on a hosting.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
'None'

Hosting_Email_Autoresponse_Delete

Description:
Deletes an existing email autoresponse.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Existing email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_Autoresponse_Get

Description:
Returns an existing email autoresponse configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Existing email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: struct
Example:
array('alias' => 'acura',
'date' => '2007-03-14 00:24:51',
'text' => 'Prueba con fechas',
'from' => '2009-04-20',
'to' => '2009-04-27',
'activated' => true,
'excluded-emails' => '',
)

Hosting_Email_Autoresponse_Activate

Description:
Activates an existing email autoresponse.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Existing email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_Autoresponse_Deactivate

Description:
Deactivates an existing email autoresponse.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Existing email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash

chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_Autoresponse_Create

Description:
Creates a new email autoresponse.
1.1 -> 1.2: Parameter "excluded" is optional.
Version: 1.2

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Existing email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
text (string): Autoresponse text. The parameter is required.
excluded (array): Exluded emails. The parameter is optional.
start (string): Start datetime. Format: yyyy-MM-dd hh:mm:ss. If empty, autoresponse starts now. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
end (string): End datetime. Format: yyyy-MM-dd hh:mm:ss. If empty, autoresponse doesn't expire. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
copy (array): Copy emails. The parameter is optional.

Response:
type: void

Hosting_Email_Redirection_Delete

Description:
Delete an existing email redirection.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Existing redirection alias (email account without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_Redirection_DeactivateCatchAll

Description:
Deactivate catch-all email redirection.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:

type: void

Hosting_Email_Redirection_Get

Description:
Returns an email account configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Redirection alias (email account without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: struct
Example:
array('alias' => 'dfsdf',
'catch-all' => false,
'date' => '2009-07-22 11:30:40',
'destination' => array('pla@olaa.com'),
)

Hosting_Email_Redirection_Update

Description:
Update an existing email redirection.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Existing redirection alias (email account without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
destination (array): Array of destination email accounts. The parameter is required.

Response:
type: void

Hosting_Email_Redirection_ActivateCatchAll

Description:
Activate catch-all email redirection.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
destination (array): Array of destination email accounts. The parameter is required.

Response:
type: void

Hosting_Email_Redirection_Create

Description:
Creates a new email redirection on a new alias.
If already exists a redirection alias or an email account over this alias name, this command will fail.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Redirection alias (email account without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
destination (array): Array of destination email accounts. The parameter is required.

Response:
type: void

Hosting_Email_Redirection_GetAll

Description:
Returns all email account redirections.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
array(array('alias' => 'dfsdf',
'catch-all' => false,
'date' => '2009-07-22 11:30:40',
'active' => true,
'destination' => array('pla@olaa.com'),
))

Hosting_Email_UsedSpace_Get

Description:
Get the used space by email accounts.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Email_UsedSpace_GetPerAlias

Description:
Get the used space by an email account.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Account alias (email account without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_AutomaticDeletion_CreateRule

Description:
Creates a new automatic deletion rule for one email account.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Account alias (email account without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
scope (string): All messages or those that have been read. The parameter is required. Accepted parameter values: "all","read".
antique (string): Antique messages. The parameter is required. Accepted parameter values: "week","fortnight","month","quarter-year","half-year","year","two-year","three-year","four-year","five-year".
ifExceedsSize (string): If used space exceeds indicated size. The parameter is required. Accepted parameter values: "any","50MB","100MB","200MB","500MB","1GB".

Response:
type: void

Hosting_Email_AutomaticDeletion_DeleteRule

Description:
Deletes automatic deletion rule for one email account.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
alias (string): Account alias (email account without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_Filter_CreateForSpam

Description:
Create a new email filter rule for spam.
Version: 1.1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
action (string): Action on filtered emails. The parameter is required. Accepted parameter values: "reject","moveToSpamFolder","moveToFolder","redirectToEmailAccount".
actionDestiny (string): Value associated with action. Only required by actions "moveToFolder" and "redirectToEmailAccount". The parameter is optional.

Response:
type: void

Hosting_Email_Filter_Delete

Description:
Delete a existing email filter rule.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
ruleId (int): Rule ID. The parameter is required.

Response:
type: void

Hosting_Email_Filter_GetGeolocalization

Description:
Returns geolocalization status (active true, inactive false).
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: bool
Example:
false

Hosting_Email_Filter_SetGeolocalization

Description:
Changes geolocalization status. If active, country filters (OriginCountryCode) will take effect on received emails.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
active (bool): True active . The parameter is required. Parameter value must be a boolean (true or false).

Response:
type: void

Hosting_Email_Filter_Create

Description:
Create a new email filter rule.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
searchObject (string): String that must match for filter email (do action). The parameter is required.
searchSubject (string): Search subject. The parameter is required. Accepted parameter values: "EmailSubject","EmailFrom","EmailTo","EmailHeaders","OriginCountryCode".
action (string): Action on filtered emails. The parameter is required. Accepted parameter values: "mark","delete".
markLabel (string): Label for filtered emails if action equals 'mark'. The parameter is optional.

Response:
type: void

Hosting_Email_Filter_GetAll

Description:
Return email filter rule list.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
country (string): Country. The parameter is optional.

Response:
type: array
Example:
array(array('action' => 'mark',
'date' => '2009-07-22 09:25:06',
'ruleId' => 258105,
'searchSubject' => 'EmailSubject',
'searchObject' => 'nvbnbv',
'markLabel' => 'vbn',
))

Hosting_Email_Dkim_Get

Description:
Get DKIM TXT zone (hostname and text).
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
array('hostname' => 'default._domainkey',
'text' => 'v=DKIM1; g=; k=rsa; p=',*
)

Hosting_Email_Dkim_Activate

Description:
Activate TLS over mail.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Email_Dkim_IsActive

Description:
Get DKIM activation status.
True means active, false means inactive.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: bool
Example:
false

Hosting_Email_Dkim_Deactivate

Description:
Deactivate TLS over mail.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Email_Antispam_AddDomainToBlacklist

Description:
Add a domain to blacklist.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
domain (string): Domain added to blacklist. The parameter is required.

Response:
type: void

Hosting_Email_Antispam_RemoveDomainFromBlacklist

Description:
Remove a domain from antispam blacklist.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
domain (string): Domain removed from blacklist. The parameter is required.

Response:
type: void

Hosting_Email_Antispam_Activate

Description:
Activate spam assasin.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Email_Antispam_SetConfig

Description:
Updates spam assasin configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
action (string): Action over detected spam emails. The parameter is required. Accepted parameter values: "mark","delete","none".
list (array): Whitelist. The parameter is optional.

Response:
type: void

Hosting_Email_Antispam_Deactivate

Description:
Deactivate spam assasin.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Email_Antispam_GetScore

Description:
Returns spam assasin aggresivity score.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: int
Example:
6

Hosting_Email_Antispam_RemoveDomainFromWhitelist

Description:
Remove a domain from antispam whitelist.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
domain (string): Domain removed from whitelist. The parameter is required.

Response:
type: void

Hosting_Email_Antispam_GetBlacklist

Description:
Returns spam assasin black list configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
array(array('id' => 12743,
'action' => 'reject',
'domain' => 'dadasdadsad.com',
),array('id' => 12745,
'action' => 'reject',
'domain' => 'alaaaaa@funciona.com',
))

Hosting_Email_Antispam_GetConfig

Description:
Returns spam assasin configuration data.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
array('delete_mesage' => false,
'label_subject' => true,
'active' => true,
)

Hosting_Email_Antispam_AddDomainToWhitelist

Description:
Add a domain to whitelist.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
domain (string): Domain added to whitelist. The parameter is required.

Response:
type: void

Hosting_Email_Antispam_SetScore

Description:
Updates spam assasin aggresivity score.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
score (int): New aggresivity score. The parameter is required. Parameter value must be between values 1 and 10 inclusive.

Response:
type: void

Hosting_Email_Antispam_GetWhitelist

Description:
Returns spam assasin white list configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: array
Example:
array(array('id' => 4503,
'action' => 'allow',

'domain' => 'ola@ola.com',
))

Hosting_Email_Account_Delete

Description:
Deletes an existing email account.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_Account_Get

Description:
Returns an existing email account.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: struct
Example:
array('alias' => 'test-linux-personal',
'account' => 'test-linux-personal@test-linux-personal.dinahosting.com',
'password' => '?????????',
'catch-all' => false,
'auto_response' => false,
'date' => '2010-03-17 12:51:02',
'id' => 0,
'disabled' => false,
)

Hosting_Email_Account_SetCopyTo

Description:
Sets copy-to emails for an email account.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String

must start with a alphanumeric char.
destination (array): Array of destination email accounts. The parameter is required.

Response:
type: void

Hosting_Email_Account_SetPassword

Description:
Changes email account password.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
password (string): New account password. The parameter is required. Password length must be between 8 and 35 characters. It can contain only alphanumeric, underscore or dash chars. String must start with a alphanumeric char. String must not contain email alias. String must not contain more than 2 consectuvie repeating characters. String must not contain more than 2 consecutive numbers in any order.

Response:
type: void

Hosting_Email_Account_Empty

Description:
Empties an existing email account.
Warning: This process can't be reversed.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_Account_Activate

Description:
Activates an existing but inactive email account.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_Account_GetCopyTo

Description:
Returns copy-to emails for an email account.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: struct
Example:
array('czxccxz@dinahosting.com','asffsasf@dinahosting.com')

Hosting_Email_Account_RemoveCopyTo

Description:
Remove copy-to for an email account.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_Account_Deactivate

Description:
Deactivates an existing email account.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: void

Hosting_Email_Account_Create

Description:
Creates a new email account.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
password (string): Account password. The parameter is required. Password length must be between 8 and 35 characters. It can contain only alphanumeric, underscore or dash chars. String must start with a alphanumeric char. String must not contain email alias. String must not contain more than 2 consectuvie repeating characters. String must not contain more than 2 consecutive numbers in any order.

Response:
type: void

Hosting_Email_Account_GetAll

Description:
Returns all email accounts.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
array(array('alias' => 'test-linux-personal',
'account' => 'test-linux-personal@test-linux-personal.dinahosting.com',
'catch-all' => false,
'auto_response' => false,
'date' => '2010-03-17 12:51:02',
'id' => 0,
'disabled' => false,
))

Hosting_Email_Antivirus_GetAttachedFilesFilters

Description:
Returns antivirus file filters configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Hosting_Email_Antivirus_SetConfig

Description:
Updates antivirus configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
action (string): Action over infected emails. The parameter is required. Accepted parameter values: "mark","delete","none".

Response:
type: void

Hosting_Email_Antivirus_GetConfig

Description:
Returns antivirus configuration data.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
array('delete_mesage' => true,
'label_subject' => false,
'active' => true,
)

Hosting_Email_Antivirus_UpdateAttachedFileFilter

Description:
Update antivirus file filters configuration.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
tag (string): Tag. The parameter is required.
action (string): Action on filtered emails. The parameter is required. Accepted parameter values: "allow","deny","scan".

Response:
type: void

Hosting_Ftp_Get

Description:
Get FTP access status.
Version: 1.1

Parameters:

hosting (string): Hosting name. The parameter is required.

Response:
type: bool
Example:
true

Hosting_Ftp_GetAccountsLimit

Description:
Get FTP account number limit for this hosting.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
array('moreAccountsAllowed' => false,
'accountLimitNumber' => 0,
)

Hosting_Ftp_Activate

Description:
Activate FTP access.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Ftp_Deactivate

Description:
Deactivate FTP access.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Ftp_Account_Delete

Description:
Deletes a FTP account.
1.0 -> 1.1: Minor improvements.

1.1 -> 1.2: Turned into a delayed execution command.
Please note this command has a delayed execution and it'll return SUCCESS_PENDING code on success and a ticket id (integer) on the data field.Version: 1.2

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): FTP account name. The parameter is required. Parameter length must be between 1 and 35 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

Response:
type: int
Example:
'None'

Hosting_Ftp_Account_SetPassword

Description:
Changes FTP account password.
1.0 -> 1.1: Minor improvements.
1.1 -> 1.2: Improved password validation.
1.2 -> 1.3: Turned into a delayed execution command.
Please note this command has a delayed execution and it'll return SUCCESS_PENDING code on success and a ticket id (integer) on the data field.Version: 1.3

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): FTP account name. The parameter is required. Parameter length must be between 1 and 35 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
password (string): New account password. The parameter is required. Parameter length must be between 6 and 71 characters.Parameter length must be at least 6 characters. It can contain only alphanumeric characters. String must contain at least one number and one character. Unsafe patterns as 111111 or 123456 aren't allowed.

Response:
type: int
Example:
'None'

Hosting_Ftp_Account_CmdStatus_GetSetPassword

Description:
Get delayed Hosting_Ftp_Account_SetPassword execution information.
Version: 1.0

Parameters:
hosting (string): Destination hosting name. The parameter is required.
orderId (int): Command order id. The parameter is required. Parameter value must be a valid integer.

Response:
type: array
Example:
'None'

Hosting_Ftp_Account_CmdStatus_ListAll

Description:
List all imports
Version: 1.0

Parameters:
hosting (string): Destination hosting name. The parameter is required.

Response:
type: array
Example:
'None'

Hosting_Ftp_Account_CmdStatus_GetDelete

Description:
Get delayed Hosting_Ftp_Account_Delete execution information.
Version: 1.0

Parameters:
hosting (string): Destination hosting name. The parameter is required.
orderId (int): Command order id. The parameter is required. Parameter value must be a valid integer.

Response:
type: array
Example:
'None'

Hosting_Ftp_Account_CmdStatus_GetCreate

Description:
Get delayed Hosting_Ftp_Account_Create execution information.
Version: 1.0

Parameters:
hosting (string): Destination hosting name. The parameter is required.
orderId (int): Command order id. The parameter is required. Parameter value must be a valid integer.

Response:
type: array
Example:
'None'

Hosting_Ftp_Account_Create

Description:
Creates a new FTP account.
1.0 -> 1.1: Minor improvements.
1.1 -> 1.2: Folder parameter is now optional (may be left empty or missing).
1.2 -> 1.3: Turned into a delayed execution command.
Please note this command has a delayed execution and it'll return SUCCESS_PENDING code on success and a ticket id (integer) on the data field.Version: 1.3

Parameters:
hosting (string): Hosting name. The parameter is required.
account (string): FTP account name. The parameter is required. Parameter length must be between 1 and 35 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.
password (string): Account password. The parameter is required. Parameter length must be between 6 and 71 characters.
folder (string): Remote folder. If missing or empty, the FTP account will use a default folder (like /www/). The parameter is optional. Parameter length must be between 0 and 150 characters.Parameter length must be between 1 and 255 characters. It should be a valid unix path.

Response:
type: int
Example:
'None'

Hosting_Ftp_Account_GetAll

Description:
Returns all existing FTP accounts.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: struct
Example:
array(array('account' => 'test-linux-personal',
'folder' => '',
'date' => '2010-03-17 12:51:02',
'main' => true,
))

Hosting_Certificate_Uninstall

Description:
Unistalls the certificate of the hosting
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Certificate_ActivateLetsEncrypt

Description:
Activates Let's encrypt certificate on the hosting
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
subdomains (string): Subdomains separated by commas. The parameter is optional.

Response:
type: void

Hosting_Certificate_DeactivateLetsEncrypt

Description:
Deactivates Let's encrypt certificate on the hosting
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Certificate_Install

Description:
Installs certificate on the hosting
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
crt (string): Certificate. The parameter is required.
key (string): Key. The parameter is required.
crtIntermediate (string): Certificate intermediate. The parameter is required.

Response:
type: void

Hosting_Php_PhpExecutionOnHtmlIsActive

Description:
Checks if hosting can execute PHP code on HTML.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: bool
Example:
'None'

Hosting_Php_ActivatePhpExecutionOnHtml

Description:

Allows PHP execution on HTML.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Php_DeactivatePhpExecutionOnHtml

Description:
Forbids PHP execution on HTML.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Php_SetVersion

Description:
Changes hosting PHP version.
Version: 1.2 Version parameter allows any content. Availability validation of PHP version added.

Parameters:
hosting (string): Hosting name. The parameter is required.
version (string): Check available versions with Hosting_GetInfo command. The parameter is required.

Response:
type: void

Hosting_Php_OverwritePhpIniWithDefaults

Description:
Overwrite php.ini values with defaults.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_Php_OverwritePhpIni

Description:
Overwrite php.ini values with custom text.
Version: 1.1

Parameters:
hosting (string): Hosting name. The parameter is required.
text (string): New text contents for php.ini file. The parameter is required.

Response:
type: void

Hosting_GetDaysWithBackups

Description:
It returns the days with hosting backups.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.
from (string): It gets the days starting from specified date. If empty, the command returns the dates of the current year. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

Response:
type: struct
Example:
'None'

Hosting_TlsMail_Get

Description:
Get TLS mail activation status.
True means active, false means inactive.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: bool
Example:
'None'

Hosting_TlsMail_Activate

Description:
Activate TLS over mail.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Hosting_TlsMail_Deactivate

Description:
Deactivate TLS over mail.
Version: 1.0

Parameters:
hosting (string): Hosting name. The parameter is required.

Response:
type: void

Housing commands

Housing_GetDataTransferInfo

Description:
Returns data transfer summary.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Housing_GetServerInfo

Description:
Returns data transfer summary.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('main_ip' => '82.98.148.16',
'ips' => array(),
'url' => '',
)

Housing_GetBandwidthConsumption

Description:
Returns bandwidth consumption summary.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
month (int): Month number. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 12 inclusive.
year (int): Year. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 2002 and 2012 inclusive.

Response:
type: struct
Example:
(Example too long, see web.)

Housing_Monitorization_Get

Description:
Returns monitorization configuration.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('emails' => array(''),
'activated' => false,
)

Housing_Monitorization_Activate

Description:
Activate monitorization.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
emails (array): Array of emails. The parameter is required.

Response:
type: void

Housing_Monitorization_Deactivate

Description:
Deactivate monitorization.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Housing_GetTransferValue

Description:
Returns max data transfer value (MBs).
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: int

Example:
0

Housing_Kvm_Get

Description:
Returns KVM configuration.
1.0 -> 1.1: new keys: user, password
Version: 1.1

Parameters:
serverName (string): Server name The parameter is required.

Response:
type: struct
Example:
array('user' => 'lhcu',
'password' => '6olto0e3',
'ip' => '10.10.10.10',
)

Housing_Kvm_Unmount

Description:
Unmount KVM.
Version: 1.0

Parameters:
serverName (string): Server name The parameter is required.
allowed_ip (string): IP that will be granted access. If it's missing, we'll use the requesting client IP. The parameter is optional. Parameter value must be a valid IP v4 address.

Response:
type: void

Housing_Kvm_Mount

Description:
Mount KVM and get public IP.
1.0 -> 1.1: returns mounted IP
1.1 -> 1.2: new parameter: allowed_ip
Version: 1.2

Parameters:
serverName (string): Server name The parameter is required.
allowed_ip (string): IP that will be granted access. If it's missing, we'll use the requesting client IP. The parameter is optional. Parameter value must be a valid IP v4 address.

Response:
type: string
Example:
'None'

Housing_PowerStatus_Restart

Description:
Restart server.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Import commands

Import_GetStatus

Description:
Get import status for a known ticketId
Possible status strings: 'PENDING', 'STARTED', 'OK', 'ERROR', 'CANCEL'
Version: 1.0

Parameters:
importId (string): Import ID The parameter is required. Parameter value must be a valid integer.

Response:
type: string
Example:
'OK'

Multihosting commands

Multihosting_MailReport_Get

Description:
Returns consumption email report status.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('periodicity' => 'never',
'emailTo' => '',
)

Multihosting_MailReport_SetOff

Description:
Deactivate consumption email report.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Multihosting_MailReport_Set

Description:
Set consumption email report.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
emailTo (string): Email destination. The parameter is required. Parameter value must be a valid email address.
periodicity (string): Report periodicity. The parameter is required. Accepted parameter values: "daily","monthly","yearly".

Response:
type: void

Multihosting_GetEmailAccountsNumber

Description:
Returns email accounts number and limit.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('used_email' => 1,
'assign_email' => -1,
'email_limit' => 200,
)

Multihosting_GetUsedSpace

Description:
Returns used space.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('total_space' => 0.5,
'space_limit' => 4000,
'homes_space' => 0.0126953125,
'web_space' => 0.1474609375,
'webmail_space' => 0,
'bds_space' => 0.345703125,
'mail_space' => 0,
'ftp_space' => 0,
'others_space' => 0,
)

Multihosting_HostingSetup_Delete

Description:
Deletes a existing user custom hosting setup.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

Response:
type: void

Multihosting_HostingSetup_Get

Description:
Returns an user custom hosting setup.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

Response:
type: struct
Example:
(Example too long, see web.)

Multihosting_HostingSetup_Update

Description:
Updates an existing user custom hosting setup.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.
options (struct): New setup options (as returned by Multihosting_HostingSetup_Get). The parameter is required.

Response:
type: void

Multihosting_HostingSetup_GetOptions

Description:
Returns all available hosting setup options.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Multihosting_HostingSetup_Create

Description:
Creates a new user custom hosting setup.

Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.
options (struct): Setup options (as returned by Multihosting_HostingSetup_Get). The parameter is required.

Response:
type: void

Multihosting_HostingSetup_GetAll

Description:
Returns all available hosting setups.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array(array('common' => 'N',
'SetupName' => 'Unlimited',
),array('common' => 'Y',
'SetupName' => 'hosting correo10',
),array('common' => 'Y',
'SetupName' => 'hosting profesional linux',
),array('common' => 'Y',
'SetupName' => 'hosting personal linux',
))

Multihosting_GetInfo

Description:
Returns multihosting info.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Multihosting_GetUsedTransfer

Description:

Returns used transfer ammount.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('total_traffic' => 4712,
'traffic_limit' => 90000,
)

Multihosting_GetLicenses

Description:
Returns licenses information.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('totalLicenses' => 5,
'availableLicenses' => 4,
'usedLicenses' => 1,
)

Multihosting_Hosting_Delete

Description:
Deletes an existing hosting.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: void

Multihosting_Hosting_ChangeSetup

Description:
Changes setup of a existing hosting.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.
setupName (string): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

Response:
type: void

Multihosting_Hosting_Activate

Description:
Activates an existing but inactive hosting.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: void

Multihosting_Hosting_Deactivate

Description:
Deactivates an existing and active hosting.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: void

Multihosting_Hosting_GetSpace

Description:
Returns used hosting space.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: struct
Example:
'None'

Multihosting_Hosting_Create

Description:
Creates a new hosting.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.
setupName (string): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.
setZones (bool): True to set new zones, false to leave them unchanged. Default: true. The parameter is required. Parameter value must be a boolean (true or false).

Response:
type: void

Multihosting_Hosting_GetAll

Description:
Returns existing hostings on multihosting.
Only for multihosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
onlyActiveHostings (bool): True to add filter: return only active hostings. False or empty, execute without filter. The parameter is optional. Parameter value must be a boolean (true or false).

Response:
type: array
Example:
array('buscador.iumx.eu')

Payment commands

Payment_Method_GetAll

Description:
Returns all payment methods.
1.0 -> 1.1: Only returns payment methods for autorenewals. Other payment methods are discarded.
Version: 1.1

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

Payment_Credit_List

Description:
Returns credit modification list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

Payment_Credit_GetDetailed

Description:
Returns detailed credit information.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: struct
Example:
array('credit' => 0,
'points' => 0.1,
'refundable' => 0,
'notRefundable' => 0,
'others' => 0,
'afiliated' => 0,
'entered' => 0,
'bonos' => 0,
'devolucion_credit' => 0,
)

Payment_Credit_GetTotal

Description:
Returns total credit.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: float
Example:
0.1

Reseller commands

Reseller_AssociateServices

Description:
Creates an association between a contact and one or more services. Warning: associating specific services with a contact replaces any existing full-app access with service-level access. Reseller-only operation.
Version: 1.0

Parameters:
service (string): Service name. The parameter is required.
family (string): Service family used to determine the service association (domain, hosting, server or any). Any: selects any service that matches the given name, regardless of service family. The parameter is required. Accepted parameter values: "domain","hosting","server","any".
contactId (int): Contact ID (see Api_Command_User_Secondary_GetAll). The parameter is required.
notifyContact (bool): Default is false. Indicates whether the contact will be notified when services are associated. SMTP is required. The parameter is optional. Parameter value must be a boolean (true or false).

Response:
type: void

Reseller_DissociateServices

Description:
Removes the association between a contact and one or more services. Reseller-only operation.
Version: 1.0

Parameters:
service (string): Service name. The parameter is required.
family (string): Service family used to determine the service association (domain, hosting, server or any). Any: selects any service that matches the given name, regardless of service family. The parameter is required. Accepted parameter values: "domain","hosting","server","any".

Response:
type: void

Services commands

Services_GetUnmanagedCloudServers

Description:
Returns unmanaged cloud server list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
'None'

Services_GetUnmanagedVps

Description:
Returns unmanaged VPS list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

Services_GetUnmanagedDedicatedServers

Description:
Returns unmanaged dedicated servers list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
array(array('serverName' => 'dl446.dinaserver.com',
'startDate' => '2013-04-05',
'endDate' => '2013-07-05',
))

Services_GetManagedCloudServers

Description:
Returns managed cloud server list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
array(array('cloudAlias' => 'c568',
'startDate' => '2014-02-06',
'endDate' => '2014-02-06',
))

Services_GetSms

Description:
Returns Sms list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
array(array('account' => 650074563,
'startDate' => '2009-07-17',
'endDate' => '2013-09-17',
))

Services_GetDomainTransfers

Description:
Returns the user's domain transfers list. Status information is not inmediate; it may take up to 1 hour to change.
On transfer codes: 'WAIT_INIT','WAIT_APROB','WAIT_AUTH','WAIT_TRANSFER'
Error codes: 'KO_AUTH','FAILED','BLOCKED','REJECTED','CANCELED','DELETED','UNKNOWN'
OK codes: 'ACTIVE'
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
'None'

Services_GetHousingServers

Description:
Returns housing servers list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:

type: array
Example:
array(array('serverName' => 'colo01.dinaserver.com',
'startDate' => '2011-07-01',
'endDate' => '2011-08-01',
),array('serverName' => 'colo02.dinaserver.com',
'startDate' => '2011-07-01',
'endDate' => '2011-08-01',
))

Services_GetMultihostings

Description:
Returns Multihostings list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

Services_GetVps

Description:
Returns VPS list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
array(array('serverName' => 'vl328.dinaserver.com',
'startDate' => '2011-04-19',
'endDate' => '2011-05-19',
))

Services_GetDomains

Description:
Returns domain list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

Services_GetManagedVps

Description:
Returns managed VPS list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
array(array('serverName' => 'vl193.dinaserver.com',
'startDate' => '2012-12-14',
'endDate' => '2013-08-09',
),array('serverName' => 'vw151.dinaserver.com',
'startDate' => '2013-03-19',
'endDate' => '2014-03-19',
))

Services_GetCloudServers

Description:
Returns cloud server list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
array(array('cloudAlias' => 'cl364',
'startDate' => '2011-06-10',
'endDate' => '2011-06-10',
))

Services_GetAllManagedServersNames

Description:
Returns managed server names list (cloud, dedicated, vps, multihosting).
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
'None'

Services_GetManagedDedicatedServers

Description:
Returns managed dedicated servers list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
array(array('serverName' => 'd546.dinaserver.com',
'startDate' => '2013-06-05',
'endDate' => '2014-06-05',
))

Services_GetDedicatedServers

Description:
Returns dedicated servers list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
array(array('serverName' => 'dw27.dinaserver.com',
'startDate' => '2011-02-16',
'endDate' => '2012-02-16',
),array('serverName' => 'd331.dinaserver.com',
'startDate' => '2011-07-12',
'endDate' => '2011-10-12',
))

Services_GetHostings

Description:
Returns hosting list.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

Sms commands

Sms_Send_Bulk_Long_Gsm7

Description:
Sends SMS to one or many destination numbers.
This command accepts GSM7 valid UTF-8 encoded characters only.
Please note the SMS may be divided on many smaller SMS (more credit is consumed).
Valid charlist: @ £ $ ¥ è é ù ì ò Ç Ø ø Å å ? _ ? ? ? ? ? ? ? ? ? ^ { } \ [~] | Æ æ ß É ! " # ¤ % ' ( )+ , - . / : ; ? ¡ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Ä Ö Ñ Ü § ¿ a b c d e f g h i j k l m n o p q r s t u v w x y z ä ö ñ ü à
Valid special chars: whitespace, carriage return and line feed.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
contents (string): Text message. Limited to a 7bit charlist. The parameter is required. Parameter length must be between 1 and 765 characters.Parameter value must contain GSM7 valid chars only.
to (array): Destination numbers. The parameter is required.
from (string): Sender name. The parameter is required. Parameter length must be between 1 and 11 characters.
when (string): Date and time, for programmed sending. Set empty for send now. Year-month-day Hour:minute:second. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

Response:
type: void

Sms_Send_Bulk_Long_Unicode

Description:
Sends SMS to one or many destination numbers.
This command accepts any UTF-8 encoded character.
Please note the SMS may be divided on many smaller SMS (more credit is consumed).
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
contents (string): Text message. It may contain any UTF-8 character. The parameter is required. Parameter length must be between 1 and 335 characters.
to (array): Destination numbers. The parameter is required.
from (string): Sender name. The parameter is required. Parameter length must be between 1 and 11 characters.
when (string): Date and time, for programmed sending. Set empty for send now. Year-month-day Hour:minute:second. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

Response:
type: void

Sms_Send_Bulk_Limited_Gsm7

Description:
Sends one SMS to one or many destination numbers.
This command accepts GSM7 valid UTF-8 encoded characters only.
Limited to one SMS for destination number.
Valid charlist: @ £ $ ¥ è é ù ì ò Ç Ø ø Å å ? _ ? ? ? ? ? ? ? ? ? ^ { } \ [~] | Æ æ ß É ! " # ¤ % ' ( )+ , - . / : ; ? ¡ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Ä Ö Ñ Ü § ¿ a b c d e f g h i j k l m n o p q r s t u v w x y z ä ö ñ ü à 0 1 2 3 4 5 6 7 8 9
Valid special chars: whitespace, carriage return and line feed.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
contents (string): Text message. Limited to a 7bit charlist. The parameter is required. Parameter length must be between 1 and 160 characters.Parameter value must contain GSM7 valid chars only.
to (array): Destination numbers. The parameter is required.
from (string): Sender name. The parameter is required. Parameter length must be between 1 and 11 characters.
when (string): Date and time, for programmed sending. Set empty for send now. Year-month-day Hour:minute:second. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

Response:
type: void

Sms_Send_Bulk_Limited_Unicode

Description:
Sends one SMS to one or many destination numbers.
This command accepts any UTF-8 encoded character.
Limited to one SMS for destination number.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
contents (string): Text message. It may contain any UTF-8 character. The parameter is required. Parameter length must be between 1 and 70 characters.
to (array): Destination numbers. The parameter is required.
from (string): Sender name. The parameter is required. Parameter length must be between 1 and 11 characters.
when (string): Date and time, for programmed sending. Set empty for send now. Year-month-day Hour:minute:second. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

Response:
type: void

Sms_Phonebook_UpdateContact

Description:
Update an existing phonebook contact.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
id (int): Contact id. The parameter is required. Parameter value must be a valid integer.

name (string): Contact name. The parameter is required. Parameter length must be between 1 and 50 characters.
phoneNumber (string): Contact phone number. The parameter is required. Parameter value must be a valid phone number.

Response:
type: void

Sms_Phonebook_DeleteContact

Description:
Delete an existing phonebook contact.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
id (int): Contact id. The parameter is required. Parameter value must be a valid integer.

Response:
type: void

Sms_Phonebook_GetAllContacts

Description:
Returns phonebook contact list.
1.0 -> 1.1: Added mising id key on response.
Version: 1.1

Parameters:
account (string): Sms account number. The parameter is required.

Response:
type: struct
Example:
array('229' => array('name' => 'Soporte',
'number' => 900854000,
'id' => 229,
),
)

Sms_Phonebook_CreateContact

Description:
Create a new phonebook contact.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
name (string): Contact name. The parameter is required. Parameter length must be between 1 and 50 characters.
phoneNumber (string): Contact phone number. The parameter is required. Parameter value must be a valid phone number.

Response:
type: void

Sms_PanelIndependentAccess_SetPassword

Description:
Set a new password for SMS account panel independent access.
Version: 1.1

Parameters:
account (string): Sms account number. The parameter is required.
newPassword (string): New password. The parameter is required.

Response:
type: void

Sms_DeliveryList_Delete

Description:
Delete an existing delivery list.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
name (string): Delivery list name. The parameter is required. Parameter length must be between 1 and 20 characters.

Response:
type: void

Sms_DeliveryList_Get

Description:
Returns a delivery list.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
name (string): Delivery list name. The parameter is required. Parameter length must be between 1 and 20 characters.

Response:
type: struct
Example:
array('members' => array(),
)

Sms_DeliveryList_GetPhones

Description:
Returns a delivery list.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
name (string): Delivery list name. The parameter is required. Parameter length must be between 1 and 20 characters.

Response:
type: array
Example:
array(743214323)

Sms_DeliveryList_Update

Description:
Update an existing delivery list (overwrites previous ids).
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
name (string): Delivery list name. The parameter is required. Parameter length must be between 1 and 20 characters.
ids (array): New phonebook ids (array). The parameter is required.

Response:
type: void

Sms_DeliveryList_Create

Description:
Create a new delivery list.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
name (string): Delivery list name. The parameter is required. Parameter length must be between 1 and 20 characters.
ids (array): Phonebook ids (array). The parameter is required.

Response:
type: void

Sms_DeliveryList_GetAll

Description:
Return all delivery lists.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.

Response:
type: array

Example:
array('name' => 'sfgjshgef',
'members' => array(),
)

Sms_Programmed_Cancel

Description:
Cancel programmed SMS.
Please note that this only works with SMS that has not yet been sent.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
id (int): History ID. The parameter is required.

Response:
type: void

Sms_GetCredit

Description:
Returns available SMS credit.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.

Response:
type: string
Example:
0.000000

Sms_History_GetSent

Description:
Returns a delivery list.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
start (int): Start. First SMS to obtain, starting at 0. The parameter is required. Parameter value must be a valid integer.
end (int): End. Last SMS to obtain. The parameter is required. Parameter value must be a valid integer.

Response:
type: array
Example:
'None'

Sms_History_GetBulkSent

Description:
Returns a delivery list.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
start (int): Start. The parameter is required. Parameter value must be a valid integer.
end (int): End. The parameter is required. Parameter value must be a valid integer.

Response:
type: array
Example:
'None'

Sms_History_GetProgrammed

Description:
Returns a delivery list.
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.
start (int): Start. The parameter is required. Parameter value must be a valid integer.
end (int): End. The parameter is required. Parameter value must be a valid integer.

Response:
type: array
Example:
'None'

Sms_SetUp

Description:
Sets a sms account up.
Version: 1.0

Parameters:
account (string): Account number. The parameter is required.
alias (string): Account alias. The parameter is optional. Parameter length must be between 0 and 100 characters.
sender (string): Default sender. The parameter is optional. Parameter length must be between 0 and 11 characters.

Response:
type: string
Example:
'None'

Sms_GetSender

Description:
Returns sender name (used for "from" field).
Version: 1.0

Parameters:
account (string): Sms account number. The parameter is required.

Response:
type: string
Example:
'Pruebas'

System commands

System_GetApiVersion

Description:
Returns Api version information.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: struct
Example:
(Example too long, see web.)

System_GetRequestTypes

Description:
Returns valid Request Types.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: struct

System_TestSimulation

Description:
Test if request is on simulation mode.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: bool

System_ListMethods

Description:
Lists command names.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

System_GetResponseTypes

Description:
Returns valid Response Types.
Default response type can be changed with parameter "responseType=xxx".
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: struct

System_MethodSignature

Description:
Returns the method signature.
Warning: This command is intented for automatic XmlRpc client use with XmlRpc requests only .
Version: 1.0

Parameters:
commandName (string): Command name. The parameter is required.

Response:
type: array

System_MethodHelp

Description:
Returns command documentation.
Version: 1.0

Parameters:
commandName (string): Command name. The parameter is required.

Response:
type: string

User commands

User_GetServices

Description:
Returns a list of services.
Version: 1.1

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

User_GetInvoicedAmount

Description:
Returns invoiced amount.
Version: 1.0

Parameters:
startDate (string): Starting date and time. Example: 2011-01-31 00:00:00. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .
endDate (string): Ending date and time. Example: 2011-12-31 23:59:59. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

Response:
type: float
Example:
9355.02

User_GetContactVerificationStatus

Description:
Checks the EPP verification status of a contact email or phone.
Version: 1.0

Parameters:
email (string): Contact email. The parameter is optional.
phone (string): Contact phone with international prefix (e.g., +34.612483957). The parameter is optional.
tld (string): TLD. The parameter is optional.

Response:
type: struct
Example:
array('email' => array('value' => 'lucia.martinez84@example.com',
'verified' => false,
),
'phone' => array('value' => +34.612483957,
'verified' => false,

),
)

User_Secondary_Delete

Description:
Delete a secondary user.
Version: 1.0

Parameters:
userId (int): Secondary user ID. The parameter is required. Parameter value must be a valid integer.

Response:
type: string
Example:
'None'

User_Secondary_Update

Description:
Update an existing secondary user.
1.0 -> 1.1 loginData is now flagged as a needed parameter.
Version: 1.1

Parameters:
idSecondaryUser (int): Existing secondary user ID. The parameter is required. Parameter value must be a valid integer.
personalData (struct): Secondary user personal identification data. The parameter is required. Parameter value must be a struct containing user information: firstname, lastname, NIF
companyData (struct): Secondary user company identification data. The parameter is optional. Parameter value must be a struct containing user information: company, legal_form, CIF, company_phone, company_fax
contactData (struct): Secondary user contact information. The parameter is required. Parameter value must be a struct containing user information: country_code, state, city, postal_code, address, phone, fax, email_address
loginData (struct): Secondary user login data. The parameter is required. Parameter value must be a struct containing user information: login, password

Response:
type: string
Example:
'None'

User_Secondary_Create

Description:
Create a secondary user.
1.0 -> 1.1 loginData is now flagged as a needed parameter.
Version: 1.1

Parameters:
personalData (struct): Secondary user personal identification data. The parameter is required. Parameter value must be a struct containing user information: firstname, lastname, NIF

companyData (struct): Secondary user company identification data. The parameter is optional. Parameter value must be a struct containing user information: company, legal_form, CIF, company_phone, company_fax
contactData (struct): Secondary user contact information. The parameter is required. Parameter value must be a struct containing user information: country_code, state, city, postal_code, address, phone, fax, email_address
loginData (struct): Secondary user login data. The parameter is required. Parameter value must be a struct containing user information: login, password

Response:
type: string
Example:
'None'

User_Secondary_GetAll

Description:
Returns all secondary users.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

User_GetInfo

Description:
Returns all secondary users.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: struct
Example:
(Example too long, see web.)

User_GetNotifications

Description:
Returns user notifications.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: struct
Example:
'None'

User_SendContactVerification

Description:
Initiates the EPP verification process for a contact's email or phone.
Version: 1.0

Parameters:
contactId (int): The ID of the contact to verify. The parameter is required. Parameter value must be a valid integer.
type (string): The type of verification: 'email' or 'phone'. The parameter is required.

Response:
type: struct
Example:
'None'

User_GetLastLoginDate

Description:
Returns user last login date time.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: string
Example:
'2011-05-03 13:22:05'

User_Data_GetValidLegalForms

Description:
Returns all valid legal form IDs and descriptions.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
(Example too long, see web.)

User_Default_GetTechContact

Description:
Returns user default tech contact data.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: struct

Example:
'None'

User_Default_GetAdminContact

Description:
Returns user default admin contact data.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: struct
Example:
'None'

User_Default_GetDnss

Description:
Returns user default DNS.
Version: 1.0

Parameters:
This command doesn't accept any parameters
Response:
type: array
Example:
array('ns.dinahosting.com','ns2.dinahosting.com','ns3.dinahosting.com','ns4.dinahosting.com')

Vps commands

Vps_Services_ChangeStatus

Description:
Change server service status.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
serviceName (string): Service name. The parameter is required. Accepted parameter values: "web","mysql","pop3","smtp","ftp","dinascanner".
serviceAction (string): Action over service. The parameter is required. Accepted parameter values: "start","stop","activate","deactivate","restart".

Response:
type: void

Vps_Services_GetAll

Description:
Returns services status.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('smtp' => 'started',
'streaming' => 'deactivated',
'web' => 'started',
'mysql' => 'started',
'ftp' => 'started',
'pop3' => 'started',
'dinascanner' => 'started',
)

Vps_Nfs_Get

Description:
Returns NFS server name
Version: 1.0

Parameters:
serverName (string): Server name The parameter is required.

Response:

type: struct
Example:
'None'

Vps_MailReport_Get

Description:
Returns consumption email report status.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('periodicidad' => 'nunca',
'direccionDeDestino' => '',
)

Vps_MailReport_SetOff

Description:
Deactivate consumption email report.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Vps_MailReport_Set

Description:
Set consumption email report.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
emailTo (string): Destination emails separated with commas. The parameter is required. Parameter value must be a valid email address.
periodicity (string): Report periodicity. The parameter is required. Accepted parameter values: "daily","weekly","monthly".

Response:
type: void

Vps_GetDataTransferInfo

Description:
Returns data transfer summary.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Vps_GetUsedSpace

Description:
Returns used space.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('total_space' => 71,
'space_limit' => 10240,
'homes_space' => 0.1240234375,
'web_space' => 66.9951171875,
'webmail_space' => 0,
'bds_space' => 3.171875,
'mail_space' => 0.396484375,
'ftp_space' => 0.0078125,
'others_space' => 0,
)

Vps_GetServerInfo

Description:
Returns VPS server info.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('name' => 'vl328.dinaserver.com',
'main_ip' => '82.98.167.9',
'ips' => array(),
)

Vps_GetBandwidthConsumption

Description:
Returns bandwidth consumption summary.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
month (int): Month number. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 12 inclusive.
year (int): Year. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 2002 and 2013 inclusive.

Response:
type: struct
Example:
(Example too long, see web.)

Vps_HostingSetup_Delete

Description:
Deletes a existing user custom hosting setup.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

Response:
type: void

Vps_HostingSetup_Get

Description:
Returns an user custom hosting setup.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

Response:
type: struct
Example:
(Example too long, see web.)

Vps_HostingSetup_Update

Description:
Updates an existing user custom hosting setup.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.
options (struct): New setup options (as returned by Vps_HostingSetup_Get). The parameter is required.

Response:
type: void

Vps_HostingSetup_GetOptions

Description:
Returns all available hosting setup options.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Vps_HostingSetup_Create

Description:
Creates a new user custom hosting setup.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
setupName (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.
options (struct): Setup options (as returned by Vps_HostingSetup_Get). The parameter is required.

Response:
type: void

Vps_HostingSetup_GetAll

Description:
Returns all available hosting setups.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Vps_GetSpaceAndTransferReportPerHosting

Description:
Returns a summary of the space and transfer data per hosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
'None'

Vps_Monitorization_Get

Description:
Returns monitorization configuration.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
array('emails' => array('ejemplo@example.com'),
'activated' => true,
)

Vps_Monitorization_Activate

Description:
Activate monitorization.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
emails (array): Array of emails. The parameter is required.

Response:
type: void

Vps_Monitorization_Deactivate

Description:
Deactivate monitorization.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Vps_GetAllHostingsFullReport

Description:
Returns a summary of the space, transfer data and plan per hosting.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
'None'

Vps_Control_GetCpuLoadByHosting

Description:
Returns cpu load by hosting information.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: array
Example:
array(array('hosting' => 'vl328.dinaserver.com',
'used_cpu' => 0.04,
'percent_used_cpu' => 100,
),array('hosting' => 'others',
'used_cpu' => 0,
'percent_used_cpu' => 0,
))

Vps_Control_GetMysqlProcesses

Description:
Returns mysql active processes information.
Only for managed virtual servers.

Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: array
Example:
array(array('info' => 'show',
'id' => 4614,
'host' => 'gdc.dinaserver.com:59830',
'ddbb' => 'NULL',
'user' => 'root',
'command' => 'Query',
'time' => 'NULL',
'status' => 'NULL',
))

Vps_Control_GetSystemStatus

Description:
Returns sysyem information.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Vps_Control_KillMysqlProcess

Description:
Kill mysql active processes.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
pid (int): Process id. The parameter is required. Parameter value must be a valid integer.

Response:
type: void

Vps_Control_GetApacheProcesses

Description:
Returns apache active processes information.
Only for linux managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)

Vps_Control_GetSystemLoad

Description:
Returns cpu load by hosting information.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:
(Example too long, see web.)

Vps_PowerStatus_Restart

Description:
Soft restart server.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Vps_PowerStatus_Get

Description:
Get server status.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: string
Example:
'Idle'

Vps_PowerStatus_TurnOn

Description:
Turn on server.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Vps_PowerStatus_Suspend

Description:
Suspend server.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Vps_PowerStatus_Reactivate

Description:
Reactivate suspended server.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Vps_PowerStatus_TurnOff

Description:
Turn off server.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Vps_PowerStatus_ElectricRestart

Description:
Electric restart of server.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Vps_PowerStatus_Unpause

Description:
Unpause paused server.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Vps_PowerStatus_Pause

Description:
Pause server.
Only for non-managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: void

Vps_GetLicenses

Description:
Returns licenses information.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: struct
Example:

array('totalLicenses' => 25,
'usedLicenses' => 8,
)

Vps_Hosting_Delete

Description:
Deletes an existing hosting.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: void

Vps_Hosting_ChangeSetup

Description:
Changes setup of a existing hosting.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.
setupName (string): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

Response:
type: void

Vps_Hosting_Activate

Description:
Activates an existing but inactive hosting.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: void

Vps_Hosting_GetSetupName

Description:
Returns hosting setup name.

Only for managed vps servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: string
Example:
'None'

Vps_Hosting_Deactivate

Description:
Deactivates an existing and active hosting.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.

Response:
type: void

Vps_Hosting_Create

Description:
Creates a new hosting.
Only for managed virtual servers.
1.0 -> 1.1: Added an existing setup name validation.
Version: 1.1

Parameters:
serverName (string): Server name. The parameter is required.
domainName (string): Hosting domain name. The parameter is required.
setupName (string): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.
setZones (bool): True to set new zones, false to leave them unchanged. Default: true. The parameter is required. Parameter value must be a boolean (true or false).

Response:
type: void

Vps_Hosting_GetAll

Description:
Returns existing hostings on Vps.
Only for managed virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.
onlyActiveHostings (bool): True to add filter: return only active hostings. False or empty, execute without filter. The parameter is optional. Parameter value must be a boolean (true or false).

Response:
type: array
Example:
array('xen.delira.eu','java.delira.eu','soportedinahosting.com','dominiopersonal.com','nophp.com','probaformacion.soportedinahosting.com','pruebapersonal.com','correodeprueba.com')

Vps_GetMonitorizedPorts

Description:
Returns monitorized ports with their status.
Only for unmanaged virtual servers.
Version: 1.0

Parameters:
serverName (string): Server name. The parameter is required.

Response:
type: array
Example:
(Example too long, see web.)