# CyberPanel API

### Introduction

This page contains documentation for usage of CyberPanel API, for plugins visit [plugin documentation](https://cyberpanel.net/docs/1-getting-started-with-cyberpanel-plugin-development/).

PHP wrapper: <https://github.com/jetchirag/cyberpanel-whmcs/blob/master/api.php>

### Reference

# Server

## Verify Connection

#### [Verify Connection](https://cyberpanel.docs.apiary.io/reference/server/verify-connection/verify-connection)

Verifying access to API, passing all these arguments as JSON object.

* adminUser (string) - Admin user to verify access to API.
* adminPass (string) - Password of above admin user.

## Login as User

#### [Login as User](https://cyberpanel.docs.apiary.io/reference/server/login-as-user/login-as-user)

Login as user, passing all these arguments as form data.

# Website Functions

## Create Website

#### [Create New Website](https://cyberpanel.docs.apiary.io/reference/website-functions/create-website/create-new-website)

Create new website, passing all these arguments as JSON object.

* adminUser (string) - Admin user to verify access to API.
* adminPass (string) - Password of above admin user.
* domainName (string) - New website address to be created.
* ownerEmail (string) - Email of the new user being created.
* packageName (string) - Package to be assigned to this website.
* websiteOwner (string) - Owner of this website being created (does not need to exist).
* ownerPassword (string) - Password of new user being.
* websitesLimit (int) (optional) - Websites limit of the user, if not specified it defaults to 1.
* acl (string)(optional) - ACL of new user, defaults to 'user'

## Delete Website

#### [Delete Website](https://cyberpanel.docs.apiary.io/reference/website-functions/delete-website/delete-website)

Deleting a website, passing all these arguments as JSON object.

* adminUser (string) - Admin user to verify access to API.
* adminPass (string) - Password of above admin user.
* domainName (string) - Website Name to be deleted.

## Change Website Package

#### [Change Website Package](https://cyberpanel.docs.apiary.io/reference/website-functions/change-website-package/change-website-package)

Changing a Website Package, passing all these arguments as JSON object.

* adminUser (string) - Admin user to verify access to API.
* adminPass (string) - Password of above admin user.
* websiteName (string) - Website Name.
* packageName (string) - Package name (This package must exist in CyberPanel).

## Suspend/Un-suspend Website

#### [Suspend/Un-suspend Website](https://cyberpanel.docs.apiary.io/reference/website-functions/suspendun-suspend-website/suspend/un-suspend-website)

Suspend/Un-suspend a Website, passing all these arguments as JSON object.

* adminUser (string) - Admin user to verify access to API.
* adminPass (string) - Password of above admin user.
* websiteName (string) - Website Name.
* state (string) - State can be 'Suspend' or 'Activate'

# User Functions

## Create new user

#### [Create a new CyberPanel user](https://cyberpanel.docs.apiary.io/reference/user-functions/create-new-user/create-a-new-cyberpanel-user)

Creating new user, passing all these arguments as JSON object.

* adminUser (string) - Admin user to verify access to API.
* adminPass (string) - Password of above admin user.
* firstName (string) - First Name of new user.
* lastName (string) - Last Name of new user.
* email (string) - Email of new user.
* userName (string) - User name (throug which they login) for new user.
* password (string) - Password for new user.
* websitesLimit (int) - Last Name of new user.
* acl (string) - ACL to be assigned to the new user, examples are 'admin', 'reseller', 'user' or any custom ACLs created by admin.
* securityLevel (string) - 'HIGH' or 'LOW'

## Get User Information

#### [Get information about User](https://cyberpanel.docs.apiary.io/reference/user-functions/get-user-information/get-information-about-user)

Getting user information, passing all these arguments as JSON object.

* adminUser (string) - Admin user to verify access to API.
* adminPass (string) - Password of above admin user.
* userName (string) - Supply username whose information is required.

## Change User Password

#### [Change password for User](https://cyberpanel.docs.apiary.io/reference/user-functions/change-user-password/change-password-for-user)

Changing user password, passing all these arguments as JSON object.

* adminUser (string) - Admin user to verify access to API.
* adminPass (string) - Password of above admin user.
* websiteOwner (string) - Supply username (whose password needs to be changed).
* ownerPassword (string) - New password for the user.

# Firewall

## Add Firewall Rule

#### [Add Firewall Rule](https://cyberpanel.docs.apiary.io/reference/firewall/add-firewall-rule/add-firewall-rule)

Adding firewall rule, passing all these arguments as JSON object.

* adminUser (string) - Admin user to verify access to API.
* adminPass (string) - Password of above admin user.
* ruleName (string) - Any user friendly name.
* ruleProtocol (string) - TCP or UDP.
* rulePort (string) - Port to be opened.
* ruleIP (string) - Open port for specified ip (Use 0.0.0.0/0 for all IPs).

add/delete firewall rule via api

* [stable](https://github.com/usmannasir/cyberpanel)

[](https://github.com/usmannasir)

[usmannasir](https://github.com/usmannasir/cyberpanel/commits?author=usmannasir) committed

1 parent [f8efe73](https://github.com/usmannasir/cyberpanel/commit/f8efe735c0bffda4c6b791d5b123572e9bafbc1c) commit 2abb1f84ed09a0e8cd83e9a07fecf0c759ac589e

Showing

with 68 additions and 1 deletion.

2 [api/urls.py](https://github.com/usmannasir/cyberpanel/commit/2abb1f84ed09a0e8cd83e9a07fecf0c759ac589e" \l "diff-cc6b462ef103b7f2f49d6e56b04c4f0a34163a0e87c55fae3f49541e08240591)

|  |
| --- |
| @@ -5,6 +5,8 @@ |

|  |
| --- |
| url(r'^createWebsite', views.createWebsite, name='createWebsiteAPI'), |

|  |
| --- |
| url(r'^deleteWebsite', views.deleteWebsite, name='deleteWebsiteAPI'), |

|  |
| --- |
| url(r'^submitWebsiteStatus', views.submitWebsiteStatus, name='submitWebsiteStatusAPI'), |

|  |
| --- |
| url(r'^deleteFirewallRule$', views.deleteFirewallRule, name='deleteFirewallRule'), |

|  |
| --- |
| url(r'^addFirewallRule$', views.addFirewallRule, name='addFirewallRule'), |

|  |
| --- |
|  |

|  |
| --- |
| url(r'^verifyConn', views.verifyConn, name='verifyConnAPI'), |

|  |
| --- |
|  |

|  |
| --- |
|  |

67 [api/views.py](https://github.com/usmannasir/cyberpanel/commit/2abb1f84ed09a0e8cd83e9a07fecf0c759ac589e" \l "diff-7fb412808754eb12ee1b172c34c44eb9ce58b3bba43d62ffcb986ad7917588a8)

|  |
| --- |
| @@ -85,7 +85,6 @@ def getPackagesListAPI(request): |

|  |
| --- |
| data\_ret = {"status": 0,'error\_message': "Could not authorize access to API"} |

|  |
| --- |
| json\_data = json.dumps(data\_ret) |

|  |
| --- |
| return HttpResponse(json\_data) |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
| @csrf\_exempt |

|  |
| --- |
| def getUserInfo(request): |

|  |
| --- |
| @@ -672,3 +671,69 @@ def submitUserCreation(request): |

|  |
| --- |
| data\_ret = {'changeStatus': 0, 'error\_message': str(msg)} |

|  |
| --- |
| json\_data = json.dumps(data\_ret) |

|  |
| --- |
| return HttpResponse(json\_data) |

|  |
| --- |
|  |

|  |
| --- |
| @csrf\_exempt |

|  |
| --- |
| def addFirewallRule(request): |

|  |
| --- |
| try: |

|  |
| --- |
| if request.method == 'POST': |

|  |
| --- |
|  |

|  |
| --- |
| data = json.loads(request.body) |

|  |
| --- |
|  |

|  |
| --- |
| adminUser = data['adminUser'] |

|  |
| --- |
| adminPass = data['adminPass'] |

|  |
| --- |
|  |

|  |
| --- |
| admin = Administrator.objects.get(userName=adminUser) |

|  |
| --- |
|  |

|  |
| --- |
| if admin.api == 0: |

|  |
| --- |
| data\_ret = {"status": 0, 'error\_message': "API Access Disabled."} |

|  |
| --- |
| json\_data = json.dumps(data\_ret) |

|  |
| --- |
| return HttpResponse(json\_data) |

|  |
| --- |
|  |

|  |
| --- |
| if hashPassword.check\_password(admin.password, adminPass): |

|  |
| --- |
| from firewall.firewallManager import FirewallManager |

|  |
| --- |
|  |

|  |
| --- |
| fm = FirewallManager() |

|  |
| --- |
| return fm.addRule(admin.pk, json.loads(request.body)) |

|  |
| --- |
| else: |

|  |
| --- |
| data\_ret = {"status": 0, |

|  |
| --- |
| 'error\_message': "Could not authorize access to API"} |

|  |
| --- |
| json\_data = json.dumps(data\_ret) |

|  |
| --- |
| return HttpResponse(json\_data) |

|  |
| --- |
|  |

|  |
| --- |
| except BaseException as msg: |

|  |
| --- |
| data\_ret = {'submitUserDeletion': 0, 'error\_message': str(msg)} |

|  |
| --- |
| json\_data = json.dumps(data\_ret) |

|  |
| --- |
| return HttpResponse(json\_data) |

|  |
| --- |
|  |

|  |
| --- |
| @csrf\_exempt |

|  |
| --- |
| def deleteFirewallRule(request): |

|  |
| --- |
| try: |

|  |
| --- |
| if request.method == 'POST': |

|  |
| --- |
|  |

|  |
| --- |
| data = json.loads(request.body) |

|  |
| --- |
|  |

|  |
| --- |
| adminUser = data['adminUser'] |

|  |
| --- |
| adminPass = data['adminPass'] |

|  |
| --- |
|  |

|  |
| --- |
| admin = Administrator.objects.get(userName=adminUser) |

|  |
| --- |
|  |

|  |
| --- |
| if admin.api == 0: |

|  |
| --- |
| data\_ret = {"status": 0, 'error\_message': "API Access Disabled."} |

|  |
| --- |
| json\_data = json.dumps(data\_ret) |

|  |
| --- |
| return HttpResponse(json\_data) |

|  |
| --- |
|  |

|  |
| --- |
| if hashPassword.check\_password(admin.password, adminPass): |

|  |
| --- |
| from firewall.firewallManager import FirewallManager |

|  |
| --- |
|  |

|  |
| --- |
| fm = FirewallManager() |

|  |
| --- |
| return fm.deleteRule(admin.pk, json.loads(request.body)) |

|  |
| --- |
| else: |

|  |
| --- |
| data\_ret = {"status": 0, |

|  |
| --- |
| 'error\_message': "Could not authorize access to API"} |

|  |
| --- |
| json\_data = json.dumps(data\_ret) |

|  |
| --- |
| return HttpResponse(json\_data) |

|  |
| --- |
|  |

|  |
| --- |
| except BaseException as msg: |

|  |
| --- |
| data\_ret = {'submitUserDeletion': 0, 'error\_message': str(msg)} |

|  |
| --- |
| json\_data = json.dumps(data\_ret) |

return HttpResponse(json\_data)