Guide to Testing Custom Code - API Authentication

Guide to Testing Custom Code

- API Authentication
- Branding
- cPanel API and UAPI Calls
- cPanel Plugin Registration
- Standardized Hooks
- Styles
- WHM API Calls

Introduction

This guide explains the basics of how to test your custom code's authentication with the cPanel & WHM server. This document lists appropriate test steps for most custom code, and helpful information to troubleshoot common problems. Make certain that you evaluate the testing requirements of your own code, and allow its functionality to determine the appropriate steps.

> **Warning:**
>
> cPanel Technical Support cannot always assist third-party developers with problems that relate to custom code. For this reason, **always** test your projects **thoroughly** before you attempt to use them on production servers.

Testing steps

> **Note:**
>
> Because the testing requirements of custom code differ, this document begins with the assumption that you already discovered problems.

1

Check to ensure that you authenticated as the correct user.

Many authentication issues occur because the custom code does not authenticate as a user who has the correct permissions.

> **Important:**
>
> - cPanel users can only call cPanel API 1, cPanel API 2, and UAPI functions, and in many cases **must** have access to the correct features for those functions.
> - WHM users can call cPanel API 1, cPanel API 2, and UAPI functions, but you **must** specify a cPanel user to call those functions as.

Check our documentation to ensure that the function you called exists in that API and performs the desired action:

- Guide to cPanel API 1
- Guide to cPanel API 2
- Guide to UAPI
- Guide to WHM API 0
- Guide to WHM API 1

2

Access Hash Authentication: Check to ensure that you formatted the access hash correctly.

Tutorials

- Tutorial - Customize the WHM User Interface with CSS
- Tutorial - Create Custom-Branded Login Pages
- Tutorial - Create a New Paper Lantern Interface
- Tutorial - Create a New Paper Lantern Interface in PHP
- Tutorial - Add a Link to the cPanel Interface
- Tutorial - Create a Standardized Hook
- Tutorial - Call UAPI's SSL:: install_ssl Function in Custom Code
- Tutorial - Use UAPI's Fileman:: upload_files Function in Custom Code
- Tutorial - Create a Custom cPanel Style
- Tutorial - Create an Integration Link
- Tutorial - Localize Text in cPanel Plugins
- Tutorial - Create a New WHM Interface in Template Toolkit
- Tutorial - Register a WHM Plugin with AppConfig
- Tutorial - Create a WHM Plugin
- Tutorial - Create a New WHM Interface in PHP
- Tutorial - Customize the WHM User Interface with CSS
- Tutorial - Create a New Paper Lantern Interface in PHP
- Tutorial - Add a Link to the cPanel Interface
- Tutorial - Use UAPI's Fileman:: upload_files Function in Custom Code

> **Warning:**
>
> We deprecated WHM's *Remote Access Key* feature in cPanel & WHM version 64. We **strongly** recommend that you use API tokens instead.

When you use an access hash or an API Token to authenticate as the `root` user, you **must** supply the hash as a single, unbroken line.

- Line breaks **will** cause authentication errors.
- To retrieve the server's access hash, use WHM's *Remote Access Key* interface (*WHM >> Home >> Clusters >> Remote Access Key*).

3

Single Sign-On: Check to ensure that you retrieved the required cookie.

When you use Single Sign-On to authenticate, you **must** perform a GET request to the URL that the function returns as the `url` value.

- This request returns a cookie that **must** exist for subsequent API calls to authenticate successfully.
- For examples of how to properly store the cookie in your custom code, read our Guide to API Authentication documentation.

Troubleshoot common issues

`FAILED LOGIN cpaneld: invalid cpanel user root`

Problem:

You receive the following error:

```
FAILED LOGIN cpaneld: invalid cpanel user root
```

Solution:
This error occurs when you attempt to supply the `root` user as a cPanel user, generally in order to call a cPanel API 1, cPanel API 2, or UAPI function. The `root` user can only call WHM API functions, not cPanel functions. You may see similar errors if you attempt to authenticate as a reseller-level account, or as an account that does not exist on the server.

To resolve this issue, you **must** authenticate as a valid cPanel account.

`Single Sign-On Request Failed with a Fatal Error: Client`

Problem:

You receive the following error:

- Tutorial - Create an Integration Link
- Tutorial - Localize Text in cPanel Plugins
- Tutorial - Create a New WHM Interface in Template Toolkit
- Tutorial - Register a WHM Plugin with AppConfig
- Tutorial - Create a New WHM Interface in PHP
- Tutorial - Create Custom-Branded Login Pages

```
Single Sign-On Request Failed with a Fatal Error: Client
```

Solution:
This error occurs when custom code that uses the Single Sign-On method to authenticate receives an invalid user (for example, if you attempt to use Single Sign-On to authenticate as a reseller in order to run a cPanel function).

To resolve this issue, you **must** authenticate as a user with the correct permissions.

Missing *Switch Account* or *Current User* menu on reseller login via Single Sign-On method

Problem:

A reseller account authenticated successfully through the Single Sign-On method, but the *Switch Account* menu (cPanel & WHM version 56 and earlier) or *Current User* menu (cPanel & WHM version 58 and later) does not display as expected in the cPanel *Home* interface.
Solution:
Due to the way in which WHM API 1's `create_user_session` function creates a user session, this is intended behavior. The reseller authenticated successfully.