

# CLEARPASS REST API

## Technical Climb Webinar

10:00 GMT | 11:00 CET | 13:00 GST  
March 27th, 2018

Presenter: Saravanan Rajagopal  
[saravanan.rajagopal@hpe.com](mailto:saravanan.rajagopal@hpe.com)



# REST API INTRODUCTION

## Cont..

- ❖ **ClearPass 6.5 introduced a set of REST APIs for use with several Guest related functions.**
- ❖ **In ClearPass 6.6, a large number of Policy Manager APIs had been exposed.**
- ❖ **In ClearPass 6.7, the number of APIs had further been increased to lot of Guest and Policy Manager entities.**

# Other Supported HTTP APIs

- ❖ **SOAP**
- ❖ **XML-RPC**
- ❖ **Tips-API (basic XML api and can be used only with Policy Manager)**

## **HTTP APIs - Basic work flow.**

- ❖ **API Authentication**
- ❖ **URL Location**
- ❖ **HTTP Method**
- ❖ **API Payload / Content.**

# OAuth (Open Standard for Authorization)

# OAuth2

- ❖ **OAuth is an authorization protocol that deals with the authorization of third-party applications or website to access data from resource provider (ClearPass).**
- ❖ **Works over http(s) and authorizes applications based on Access Token.**
- ❖ **Once an application has an access token, it can access the various APIs serviced by the resource provider.**

# OAuth2 - Roles.

- ❖ **Resource Owner (User/Person)**
- ❖ **Resource Server (ClearPass)**
- ❖ **Client Application**
- ❖ **Authorization Server (ClearPass)**

# OAuth2 - Grant Types

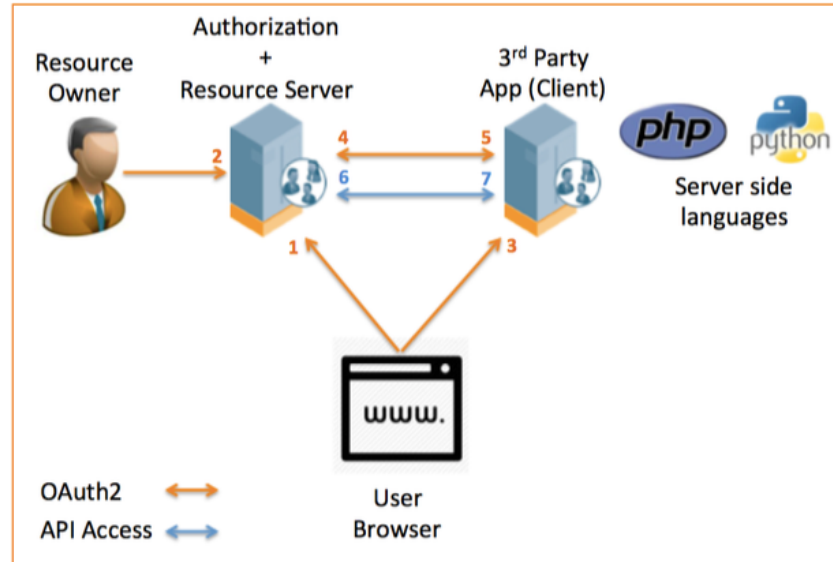
- ❖ **Authorization Code** — Web application with server side coding.
- ❖ **Implicit** — Client side web and mobile apps.
- ❖ **Password (Resource Owner Password)** - Official web and mobile apps.
- ❖ **Client Credentials** — Meant to be used with application code.

- **Client ID and Secret.**

**You will receive a Client ID and Client Secret after registering your App with the resource provider.**



# Grant - Authorization Code



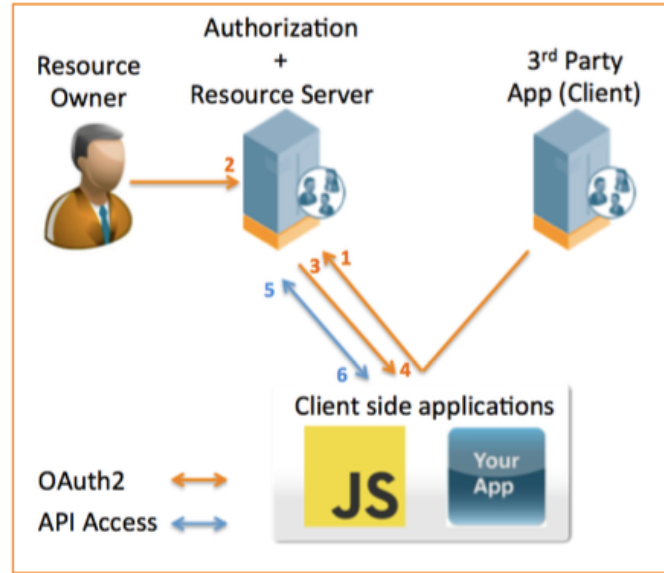
Client ID and Client Secret should be obtained from resource provider (Ex: Facebook/ClearPass)

[https://login.fbxxx.com/oauth?response\\_type=code&client\\_id=xxx&redirect\\_uri=xxx&scope=<email>](https://login.fbxxx.com/oauth?response_type=code&client_id=xxx&redirect_uri=xxx&scope=<email>)

[https://test\\_cppm.com/oauth/callback?code=xxx](https://test_cppm.com/oauth/callback?code=xxx)

[https://api.fbxxx.com/oauth/token?grant\\_type=authorization\\_code&code=xxx&client\\_id=xxx&client\\_secret=xxx&redirect\\_uri=xxx](https://api.fbxxx.com/oauth/token?grant_type=authorization_code&code=xxx&client_id=xxx&client_secret=xxx&redirect_uri=xxx)

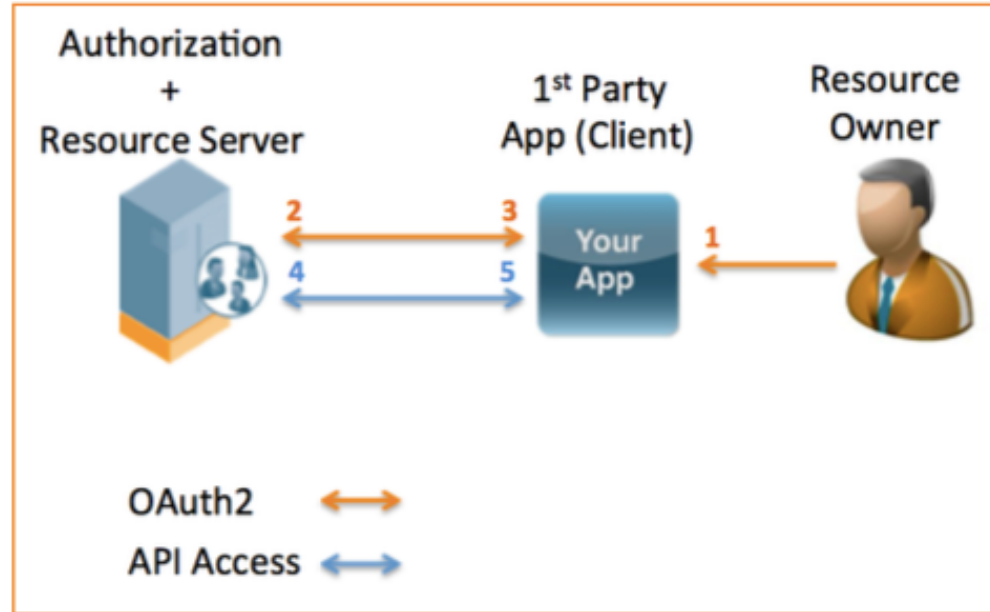
# Grant - Implicit



[https://login.xyz.com/oauth?response\\_type=token&client\\_id=xxx&redirect\\_uri=xxx&scope=<email>](https://login.xyz.com/oauth?response_type=token&client_id=xxx&redirect_uri=xxx&scope=<email>)

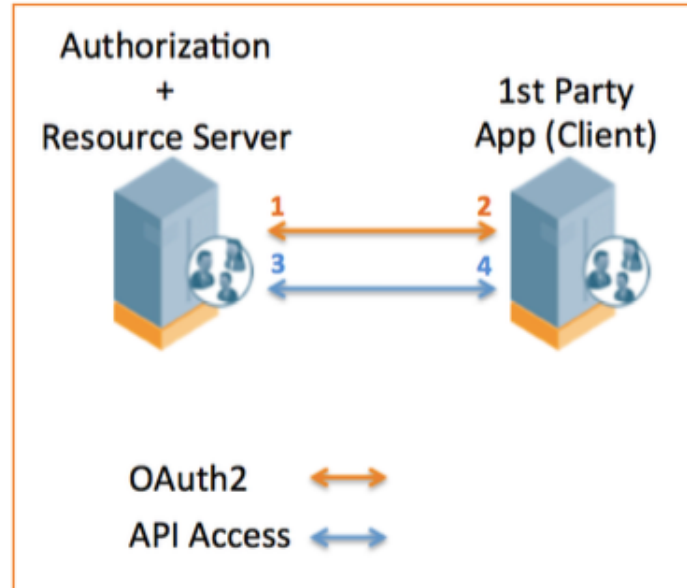
[https://test\\_cppm.com/oauth/callback?token=xxx](https://test_cppm.com/oauth/callback?token=xxx)

# Grant – Password



POST [https://test.cppm.com/api/oauth?grant\\_type=password&username=xxx&password=xxx&client\\_id=xxx](https://test.cppm.com/api/oauth?grant_type=password&username=xxx&password=xxx&client_id=xxx)

# Grant – Client Credentials



POST [https://test.cppm.com/api/oauth?grant\\_type=client\\_credentials&client\\_id=xxx&client\\_secret=xxx](https://test.cppm.com/api/oauth?grant_type=client_credentials&client_id=xxx&client_secret=xxx)

# REST API HTTP METHODS

# HTTP Methods

- ❖ **HTTP GET** – Get/download a web page from a server.
- ❖ **HTTP POST** – Fill a form in web page and submit to a server (ClearPass).
- ❖ **HTTP PUT** – Replace an object on the server.
- ❖ **HTTP PATCH** – Update an object on the server.
- ❖ **HTTP DELETE** – Delete an object on the server.

# HTTP Methods (Cont..)

Guest Users	HTTP Method	Intention
/api/guest	GET	Get a list of guest accounts
/api/guest/{guest_id}	GET	Get a guest account (identified by {guest_id})
/api/guest	POST	Create a new guest account(s)
/api/guest/{guest_id}	PUT	Replace a guest account (identified by {guest_id})
/api/guest/{guest_id}	PATCH	Update some fields of a guest account (identified by {guest_id})
/api/guest/{guest_id}	DELETE	Delete a guest account (identified by {guest_id})

# REST API PAYLOAD/CONTENT



# API Payload/Content

The API content needs to be presented in a format that can be understood by the API server (ClearPass). Some of the more common formats (Content-Type) supported by APIs are:

- ❖ **JSON** (`application/json`)
- ❖ **XML** (`application/xml`)
- ❖ **Form Encoded** (`application/x-www-form-urlencoded`)
- ❖ **Plain Text** (`text/plain`)

The Content-Type should be included in the HTTP header, so the API server will know what format you are sending in the API payload.

ClearPass REST APIs are designed to accept the JSON Content-Type in the HTTP body to submit API calls.

# JSON (JavaScript Object Notation).

**JSON is a minimal readable format for structuring data and used to transmit data between a server and web application.**

**JSON data is written as key/value pairs.**

**Key:** String enclosed in quotation.

**Value:** Value can be a string, integer, Boolean, array or object.

**Key/Value Pair** - follows a syntax with the key followed by a colon followed by the value.

**Sample Syntax :**

1: "Username" : "saran"

2: "attributes": { "Location": "IND", "Username": "saran", "Device Name":  
"Saran's IOS" }

# CLEARPASS CONFIGURATION

# STEP 1 - Creating API Client

The first step is to create/register a new API client in the ClearPass Server.

Navigation: ClearPass Guest → Administration → API Service → API Clients → Create API Client.

Home » Administration » API Services » API Clients

## Create API Client

Use this form to create a new API client.

Create API Client	
* Client ID:	GuestManagement <small>The unique string identifying this API client. Use this value in the OAuth2 "client_id" parameter.</small>
Description:	<div></div> <small>Use this field to store comments or notes about this API client.</small>
Enabled:	<input checked="" type="checkbox"/> Enable API client
* Operator Profile:	API Guest Operator <small>The operator profile applies role-based access control to authorized OAuth2 clients. This determines what API objects and methods are available for use.</small>
* Grant Type:	Username and password credentials (grant_type=password) <small>Only the selected authentication method will be permitted for use with this client ID.</small>
Public Client:	<input type="checkbox"/> This client is a public (trusted) client <small>Public clients have no client secret.</small>
Refresh Token:	<input checked="" type="checkbox"/> Allow the use of refresh tokens for this client <small>An OAuth2 refresh token may be used to obtain an updated access token. Use grant_type=refresh_token for this.</small>
Client Secret:	<div> tS5oSFwJ6xIFvbsOZgJsgFjhF87Wj5imSN8eYShjChPM</div> <small>Use this value in the OAuth2 "client_secret" parameter. NOTE: This value is encrypted when stored and cannot be displayed again.</small>
Access Token Lifetime:	8 hours <small>Specify the lifetime of an OAuth2 access token.</small>
Refresh Token Lifetime:	14 days <small>Specify the lifetime of an OAuth2 refresh token.</small>
<div>Create API Client Cancel</div>	

\* required field

Home » Administration » API Services » API Clients

## Create API Client

Use this form to create a new API client.

Create API Client	
* Client ID:	<input type="text" value="GuestManagement"/> <small>The unique string identifying this API client. Use this value in the OAuth2 "client_id" parameter.</small>
Description:	<input type="text"/> <small>Use this field to store comments or notes about this API client.</small>
Enabled:	<input checked="" type="checkbox"/> Enable API client
* Operator Profile:	<input type="text" value="API Guest Operator"/> <small>The operator profile applies role-based access control to authorized OAuth2 clients. This determines what API objects and methods are available for use.</small>
* Grant Type:	<input type="text" value="Username and password credentials (grant_type=password)"/> <small>Only the selected authentication method will be permitted for use with this client ID.</small>
Public Client:	<input checked="" type="checkbox"/> This client is a public (trusted) client <small>Public clients have no client secret.</small>
Refresh Token:	<input checked="" type="checkbox"/> Allow the use of refresh tokens for this client <small>An OAuth2 refresh token may be used to obtain an updated access token. Use grant_type=refresh_token for this.</small>
Access Token Lifetime:	<input type="text" value="8"/> <input type="text" value="hours"/> <small>Specify the lifetime of an OAuth2 access token.</small>
Refresh Token Lifetime:	<input type="text" value="14"/> <input type="text" value="days"/> <small>Specify the lifetime of an OAuth2 refresh token.</small>
<input type="button" value="Create API Client"/> <input type="button" value="Cancel"/>	

\* required field

# Cont..

- ❖ **Operator Profile** – Defines the class of user and privileges in ClearPass server.
- ❖ **Grant Type** – Had been discussed previously in OAuth2 overview.
- ❖ **Public Client** – App doesn't need to present the client secret in the OAuth2 authorization request.
- ❖ **Refresh Token** – Allows an App to recover short lived access token.
- ❖ **Access Token Lifetime**
- ❖ **Refresh Token Lifetime**

Home » Administration » Operator Logins » Profiles

## Edit Operator Profile (API Guest Operator)

Use this form to make changes to the operator profile **API Guest Operator**.

Operator Profile Editor

\* Name:

API Guest Operator

Enter a name for this operator profile.

Description:

Operators with this profile can use the API to manage guest accounts.

Comments or descriptive text about the operator profile.

Access

These options control what operators with this profile are permitted to do.

Enabled:

☒ Allow operator logins

If unchecked, operators with this profile will not be able to log in.

Operator Privileges

**Administrator**

Select operator permissions for system administration and management tasks.

No Access

**Advertising Services**

Select operator permissions for managing advertising content and services.

No Access

**AirGroup Services**

Select operator permissions for access to AirGroup services.

No Access

**API Services**

Select operator permissions for API access and management.

Allow API Access

No Access

☒ Allow Access

Operators with this privilege are permitted to make API calls. Additional privileges are also required, depending on the API.

Configure SOAP Web Services (Legacy)

No Access

☒ Read Only

☐ Full

Operators with this privilege can change system settings for SOAP web services.

List SOAP Web Services (Legacy)

No Access

☐ Read Only

☒ Full

Operators with this privilege can browse the available SOAP web services and access the service definitions (WSDL).

Manage API Clients

☒ No Access

☐ Read Only

☐ Full

Operators with this privilege may view and manage API clients (OAuth2 authentication).

SOAP API (Legacy)

No Access

☐ Read Only

☒ Full

Operators with this privilege can use SOAP web services to perform system functions. Additional privileges are also required, depending on the API.

XMLRPC API (Legacy)

No Access

☒ Allow Access

Operators with this privilege can access system functions through the XMLRPC API. Additional privileges are also required, depending on the API.

**Guest Manager**

Select operator permissions for managing guest users for a network.

Active Sessions

No Access

☐ Read Only

☒ Full

Operators with the Active Sessions privilege may disconnect active sessions or change authorization for user accounts.

Active Sessions History

No Access

☒ Read Only

Navigation: ClearPass Guest → Administrator → Operator Logins → Profiles

23







# Step 2 – Verify User Account

Create a new service to authentication the OAuth2 authorization request, as the grant type is **“Password”**.

Use the ClearPass Policy Manager built in wizard to create a new service for API user authentication.

Navigation: ClearPass Configuration » Start Here  
To configure a Service and related policies using the full wizard, go [here](#).

Select Template Category: **Application Templates**

-  **Aruba Auto Sign-On**  
Service template for accessing SAML based single sign-on enabled applications using network authenticated identity through Aruba controllers.
-  **Certificate/Two-factor Authentication for ClearPass Application Login**  
To use certificate or two-factor authentication to allow access to ClearPass applications.
-  **ClearPass Admin Access (Active Directory)**  
Service template for access to ClearPass Policy Manager administration console (Active Directory users).
-  **ClearPass Admin SSO Login (SAML SP Service)**  
SAML-based Single Sign-On (SSO) access to ClearPass Policy Manager, Insight, Guest and Operator screens via external Identity Provider.
-  **ClearPass Identity Provider (SAML IdP Service)**  
Service template to provide a SAML based single sign-on service that can be used by other applications.
-  **OAuth2 API User Access**  
Service template for API clients authenticating with username and password (OAuth2 grant type "password")



# Cont..

Configuration » Services » Edit - OAuth2 API User Access

## Services - OAuth2 API User Access

Summary	Service	Authentication	Roles	Enforcement
<b>Service:</b>				
Name:	OAuth2 API User Access			
Description:	Authentication service for API access using OAuth2			
Type:	Aruba Application Authentication			
Status:	Enabled			
Monitor Mode:	Disabled			
More Options:	-			
<b>Service Rule</b>				
Match ANY of the following conditions:				
Type	Name	Operator	Value	
1. Application	Name	EQUALS	OAuth2	
<b>Authentication:</b>				
Authentication Sources:	1. [Local User Repository] ← Created the user account in local db for testing. 2. [Admin User Repository] User_id - test_api passwd - aruba123			
Strip Username Rules:	-			
<b>Roles:</b>				
Role Mapping Policy:	-			
<b>Enforcement:</b>				
Use Cached Results:	Disabled			
Enforcement Policy:	[Guest Operator Logins]			
<a href="#">Back to Services</a> <span>Disable</span> <span>Copy</span> <span>Save</span> <span>Cancel</span>				

Note: The “OAuth2 API User Access” service is not required for the Grant Type – **Client Credentials**

# Step 3 – Test API Authorization.

## URL Location - /api/oauth/

**Ex: https://<ClearPass\_ip/hostname>/api/oauth/**

```
curl -X POST "https://10.17.164.223/api/oauth"; \  
-H "Content-Type: application/json" \  
-d $'{ "grant_type": "password", "username": "test_api", "password": "aruba123", "client_id":  
"GuestManagement"}' \  
-m 30 \  
-v \  
-k
```

```
{"access_token":"1dad233e63c29d7e203e0207f51d1cbe76e205c3","expires_in":28800,"token_type":"Bearer","scope":null,"refresh_token":"454b169e2d43d1640fcc4ce4c4cde00e58254701"}
```

# Cont..

ClearPass API Access – GuestManagement

GET <https://api.myproduct.com/v1/users> Send

ClearPass API Access

No Environment Cookies

Filter

GET GuestManagement

Body OAuth 2 Query Header 1 Docs

GRANT TYPE Resource Owner Password Credentials

USERNAME test\_api

PASSWORD .....

ACCESS TOKEN URL https://10.17.164.223/api/oauth

CLIENT ID GuestManagement

CLIENT SECRET

Advanced Options

REFRESH TOKEN

75aec5fb8d1191288c8bef4a1c79c857283b0ca8

ACCESS TOKEN (EXPIRES IN 7 HOURS)

8ddb44305b067563629bbe12fd0fc97a14f47b4d

Clear Refresh Token

# Cont..

## ClearPass API Explorer

**ApiAuthentication** : Obtain an OAuth2 access token for making API calls

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

POST /oauth

Obtain an OAuth2 access token for making API calls

### Implementation Notes

Obtain an OAuth2 access token for making API calls

### Response Class

Model | Model Schema

#### OAuth2Response {

**access\_token** (string): Access token issued by the authorization server,  
**token\_type** (string): Type of access token. Always set to "Bearer",  
**expires\_in** (integer): The lifetime in seconds of the access token,  
**refresh\_token** (string, optional): Refresh token, if enabled for this API client. Can be used to obtain a new access token,  
**scope** (string, optional): Scope of the access token  
}

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	<pre>{   "grant_type": "password",   "client_id": "GuestManagement",   "client_secret": "",   "username": "test_api",   "password": "aruba123",   "scope": "",   "refresh_token": "" }</pre>		body	Model   Model Schema <b>ApiAuthentication {</b> <b>grant_type</b> (string) = ['client_credentials' or 'password' or 'refresh_token']: OAuth2 authentication method, <b>client_id</b> (string): Client ID defined in API Clients, <b>client_secret</b> (string, optional): Client secret, required if the API client is not a public client, <b>username</b> (string, optional): Username for authentication, required for

Parameter content type:

Try it out!

View Response

### Request URL

https://10.17.164.223:443/api/oauth

### Response Body

```
{
  "access_token": "f3a1ba87b910e177c3b48ce235c0135b5523adc9",
  "expires_in": 28800,
  "token_type": "Bearer",
  "scope": null,
  "refresh_token": "27d852c5db43a7e07094c2d94ff2996bab37dc"
}
```

### Response Code

200

# GUEST ACCOUNT GET AND POST SAMPLES

# GET

ClearPass API Access – GuestManagement

GET <https://10.17.164.223/api/guest/> Send 200 OK TIME 865 ms SIZE 941 B

No Environment Cookies

Filter

GET GuestManagement

Body OAuth 2 Query Header Docs

GRANT TYPE Resource Owner Password Credentials

USERNAME test\_api

PASSWORD .....

ACCESS TOKEN URL https://10.17.164.223/api/oauth

CLIENT ID GuestManagement

CLIENT SECRET

Advanced Options

REFRESH TOKEN

dcf14722a78136c070906030a79806fc198bcaec

ACCESS TOKEN (EXPIRES IN 8 HOURS)

5682856b4b1b7a69b02cbf592e3c81db7d1918a1

Clear Refresh Token

Preview Header Cookie Timeline

```
1 - {
2   ~ "links": {
3     ~ "self": {
4       ~ "href": "https://10.17.164.223/api/guest/?
5         calculate_count=false&offset=0&limit=25&sort=-id&filter=%7B%7D"
6     },
7     ~ "first": {
8       ~ "href": "https://10.17.164.223/api/guest/?
9         calculate_count=false&offset=0&limit=25&sort=-id&filter=%7B%7D"
10    }
11  },
12  ~ "_embedded": {
13    ~ "items": [
14      {
15        ~ "id": "3001",
16        ~ "username": "saravanan@arubanetworks.com",
17        ~ "start_time": 1521810725,
18        ~ "expire_time": 1521897125,
19        ~ "sponsor_name": "admin",
20        ~ "sponsor_profile": "1",
21        ~ "enabled": true,
22        ~ "current_state": "active",
23        ~ "notes": "",
24        ~ "visitor_carrier": null,
25        ~ "role_name": "[Guest]",
26        ~ "role_id": 2,
27        ~ "email": "saravanan@arubanetworks.com",
28        ~ "source": "create_user",
29        ~ "do_expire": "1",
30        ~ "create_time": 1521810725,
31        ~ "remote_addr": "10.20.34.63",
32        ~ "visitor_company": "Aruba",
33        ~ "visitor_name": "Saran",
34        ~ "expire_postlogin": "0",
35        ~ "simultaneous_use": "1",
36        ~ "sponsor_profile_name": "Super Administrator",
37        ~ "expired_notify_status": "1",
38      },
39    ],
40  },
41  ~ "self": {
42    ~ "href": "https://10.17.164.223/api/guest/?
43      calculate_count=false&offset=0&limit=25&sort=-id&filter=%7B%7D"
44  }
45 }
```

# POST

The screenshot shows a REST client interface for a POST request to `https://10.17.164.223/api/guest/`. The request body is a JSON object with the following fields:

```
1 {
2   "do_expire": 4,
3   "expire_time": "1521989421",
4   "username": "saravanan.rajagopal@hpe.com",
5   "password": "aruba123",
6   "role_id": 2
7 }
```

Annotations on the request body:

- A red arrow points to the `expire_time` field with the text "Date/Time data should be epoch".
- A red arrow points to the `password` field with the text "Must send password".

The response status is `201 Created` with a response time of `TIME 462 ms` and a size of `SIZE 450 B`. The response body is a JSON object:

```
1 {
2   "id": "3002",
3   "username": "saravanan.rajagopal@hpe.com",
4   "start_time": 1521816677,
5   "expire_time": 1521989421,
6   "sponsor_name": "oauth2:GuestManagement",
7   "sponsor_profile": "10",
8   "enabled": false,
9   "current_state": "disabled",
10  "notes": null,
11  "visitor_carrier": null,
12  "role_name": "[Guest]",
13  "role_id": 2,
14  "source": "api",
15  "do_expire": "4",
16  "create_time": 1521816677,
17  "sponsor_profile_name": "API Guest Operator",
18  "links": {
19    "self": {
20      "href": "https://10.17.164.223/api/guest/3002"
21    }
22  }
23 }
```

Annotations on the response body:

- A red arrow points to the `username` field.

# HTTP Status Codes

## Common Error/Status Codes (HTTP)

- ❖ 201 Created
- ❖ 401 Unauthorized
- ❖ 403 Forbidden
- ❖ 406 Not Acceptable
- ❖ 415 Unsupported Media Type
- ❖ 422 Unprocessable Entity

```
{"type":"http://www.w3.org/Protocols/rfc2616/rfc2616- sec10.html","title":"invalid_client","status":400,"detail":"This client is invalid or must authenticate using a client secret"}
```

```
{"type":"http://www.w3.org/Protocols/rfc2616/rfc2616- sec10.html","title":"Forbidden","status":403,"detail":"Client does not have \u2018Allow API Access\u2019 privilege"}
```

```
{"type":"http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html","title":"Not Acceptable","status":406,"detail":"Cannot honor Accept type specified"}
```



# LIVE DEMO AND DEBUGGING

# API Explorer – Testing and Debugging

**Please make use of built-in API explorer in ClearPass for testing and debugging.**

Navigation: ClearPass Guest → Administration → API Services → API Clients → API Explorer.

or ClearPass Guest → Administration → API Services → API Explorer.

aruba ClearPass API Explorer		
API	Services	Versions
ApiFramework	ApiAuthentication, ApiClient	v1
Authentication	AuthMethod	v1
Certificates	CertSignRequest, CertTrustList, CertTrustListDetails, SelfSignedCert, ServerCert, ServiceCert	v1
Dictionaries	Attribute, ContextServerAction, Fingerprint	v1
DigitalPass	DigitalPass	v1
Events	LoginAudit, SystemEvents	v1
Extension	Instance, InstanceConfig, InstanceLog, InstanceReinstall, InstanceRestart, InstanceStart, InstanceStop, Store	v1
ExternalServers	EndpointContextServer, FileBackupServer, SyslogExportFilter, SyslogTarget	v1
GuestManager	ActiveSession, ActiveSessionDisconnect, ActiveSessionReauthorize, Configuration, Device, Guest, GuestDigitalPass, GuestReceipt, GuestSponsor, PrintTemplate, RandomPassword, SMSReceipt, SMTPReceipt, WebLogin	v1
Identity	Endpoint, LocalUser, LocalUserPasswordPolicy, Role, StaticHostList	v1
Insight	Endpoint	v1
MessagingServices	EmailSend, MessagingSetup	v1
Network	NetworkDevice, NetworkDeviceGroup, ProxyTarget	v1
Onboard	Certificate, CertificateChain, CertificateExport, CertificateImport, CertificateNew, CertificateReject, CertificateRequest, CertificateRevoke, CertificateSign, Device, User	v1
OnGuard	GlobalSettings, Settings	v1

# Debugging ClearPass API Framework

Enable debug under ClearPass Guest → Administration → Plugin Manager → API Framework → Configuration.

Navigate to ClearPass Guest → Administration → Support → Application Log, to view the logs.

Home » Administration » Plugin Manager

## API Framework 6.7.1-35330 Configuration

Set the configuration options for API Framework 6.7.1-35330.

Configure API Framework 6.7.1-35330	
Access Token Lifetime:	<input type="text" value="8"/> hours Specify the default lifetime for an OAuth2 access token. This parameter may be configured separately for each API client.
Refresh Token Lifetime:	<input type="text" value="14"/> days Specify the lifetime of an OAuth2 refresh token. This parameter may be configured separately for each API client.
* API Logging:	<div>Debug — log debug information</div> <div>Select an option for logging API-related events.<ul style="list-style-type: none"><li>• 'Extended' will log most API calls.</li><li>• 'Trace' will log full details of all API calls, including authorization failures.</li></ul></div>
Allowed Origins:	<div></div> <div>Security settings for Cross-Origin Resource Sharing (CORS). List the allowed origins for browser-initiated API requests, one per line. Specify * to allow all origins. This may also be used as a wildcard, e.g. *.example.com. Leave blank to never allow cross-domain API requests.</div>
Arbitrary Sort:	<div><input type="checkbox"/> Allow API calls to specify arbitrary sort fields</div> <div>Enable this option if you receive a "Cannot sort by field" error. Note: This has performance implications when large query results are involved.</div>
<div>Save Configuration</div>	

\* required field

Home » Administration » Support » Application Log

## Application Log

The events and messages generated by this application are logged here. For in-depth information about an event, click on it.

Quick Help

Filter

Export

Keywords:

Enter keywords to filter the logs. Use '-' to negate and quotes to group keywords.

Time	IP	User	Severity	Message
2018-03-23 21:00:10	10.20.34.63		error	OAuth2 username and password authentication failed

OAuth2 username and password authentication failed

Client: 10.20.34.63:55132

Script: /guest/apigility.php

Function: validateRequest

Arguments: array (

'client\_id' => 'GuestManagement',

'credentials' => array (

'username' => 'test\_api',

),

'attributes' => array (

'OAuth2-Client-ID' => 'GuestManagement',

'OAuth2-Client-IP-Address' => '10.20.34.63',

'OAuth2-Client-User-Agent' => 'insomnia/5.14.7',

),

)

Details: array (

'rtn' => array (

'error' => 1,

'user' => array (

'AuthRequestId' => 'W0000000d-01-5ab51d82',

),

'certificate\_error' => false,

'message' => 'ClearPass Policy Manager server returned an error: Authentication Failed',

),

)

2018-03-23 21:00:09

10.20.34.63

info

API: Created new OAuth2 access token for client ID 'GuestManagement'

2018-03-23 20:55:33

10.20.34.63

admin

info

Configuration changed - Kernel - Updated configuration for plugin: API Framework 6.7.1-35330

THANK YOU!