

# 6.8.1 Third Party Testing Approach

This article to propose an approach for 3rd Party Testing to CDCB.

Get token by Postman

To access the API, the user needs to get a token. After getting the token, we can use that token to test the APIs in Web connect.

## The Swagger API Documentation link

API Service	Links
Web application	<a href="https://webconnect.uscdcb.com/docs">https://webconnect.uscdcb.com/docs</a>
Data Exchange	<a href="https://webconnect.uscdcb.com/api/data-exchange/docs">https://webconnect.uscdcb.com/api/data-exchange/docs</a>
ETL process	<a href="https://webconnect.uscdcb.com/api/etl/docs">https://webconnect.uscdcb.com/api/etl/docs</a>

## Account

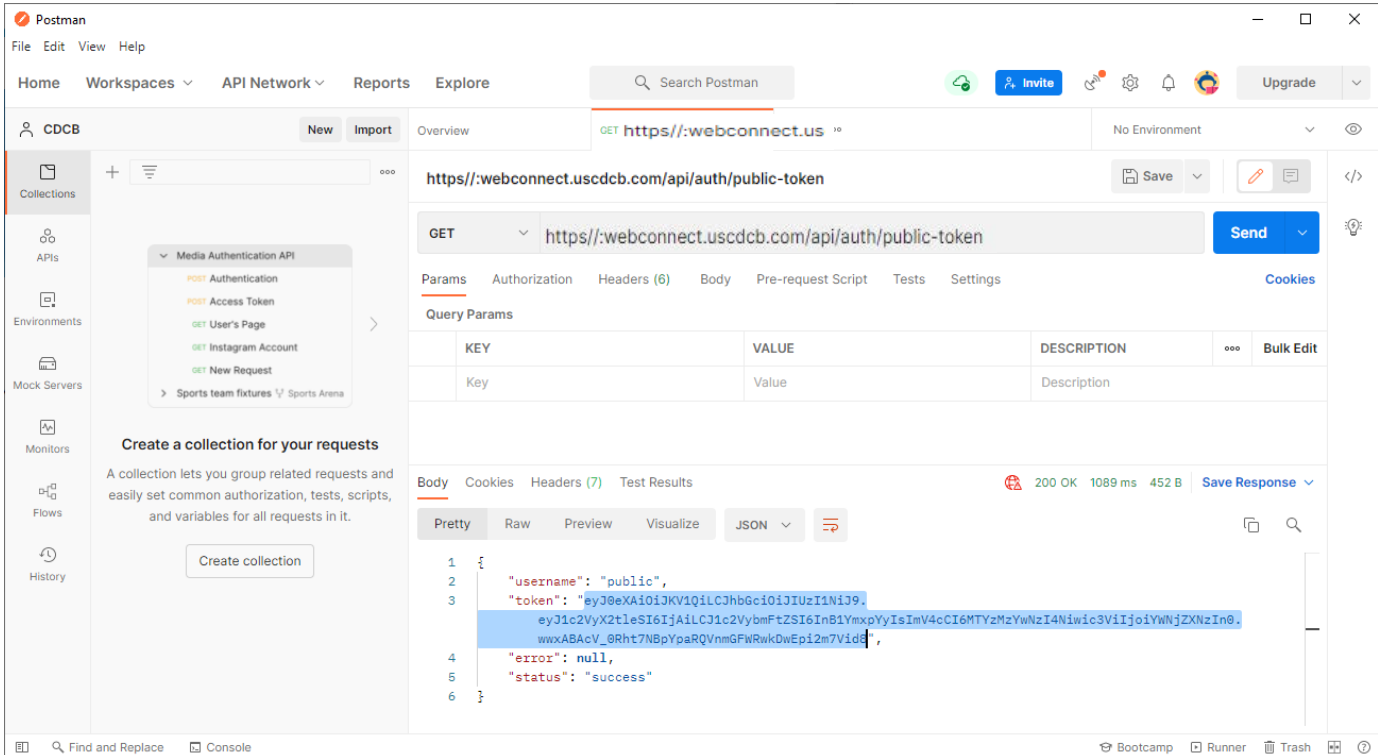
- Username: thirdparty
- Password: Tma@cbcb123

There're 2 types of token:

- For Public Users: to APIs in public pages.
- For Register Users: to APIs in pages required logging in.

## Public Users

1. With the Public Users, they need to send a request to `api/auth/public-token`.



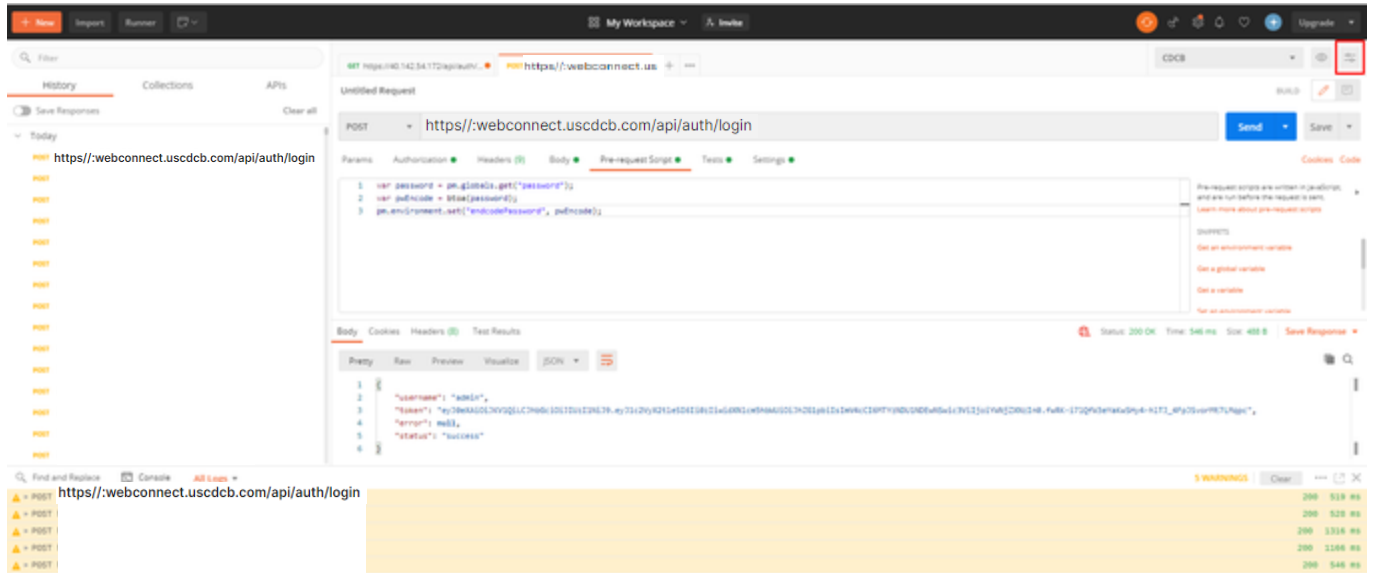
2. The result returns the token, then we can use this token to test the APIs.

## Registered Users

With the Registered Users, they need to send a request to `api/auth/login`. Before we send a request we need to encode the password by using `Base64`.

Please see example below using **Postman**:

**Step 1:** Import Enviroment CDCB to Postman



The screenshot displays the Postman application interface. At the top, the workspace is named 'My Workspace'. The left sidebar shows a 'History' tab with a list of requests, including 'https://webconnect.uscddb.com/api/auth/login'. The main workspace area shows an 'Untitled Request' for a POST method to 'https://webconnect.uscddb.com/api/auth/login'. The 'Pre-request Script' tab is active, containing the following JavaScript code:

```
1 var password = pm.globals.get("password");
2 var authcode = $eval(password);
3 pm.environment.set("envcode=password", authcode);
```

The 'Body' tab shows a JSON response:

```
{
  "username": "admin",
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9",
  "error": null,
  "status": "success"
}
```

At the bottom, the console shows the execution of the request, with a table of results:

Method	URL	Status	Time	Size
POST	https://webconnect.uscddb.com/api/auth/login	200	519 ms	
POST		200	520 ms	
POST		200	1314 ms	
POST		200	1165 ms	
POST		200	548 ms	

An environment is a set of variables that allow you to switch the context of your requests. Environments can be shared between multiple workspaces. [Learn more about environments](#)

You can declare a variable in an environment and give it a starting value, then use it in a request by putting the variable name within curly-braces. [Create an environment](#) to get started.

Globals

Import

Add

MANAGE ENVIRONMENTS

Send

Import Environment

Select environment files from your computer

Choose Files

No file chosen

Open

This PC > Documents >

Search Documents

Organize New folder

Name	Date modified	Type
Custom Office Templates	10/5/2020 10:35 AM	File folder
FeedbackHub	9/14/2021 11:14 AM	File folder
IISExpress	10/2/2020 9:15 AM	File folder
Lightshot	3/11/2021 5:20 PM	File folder
My Data Sources	7/30/2021 4:38 PM	File folder
My Web Sites	10/2/2020 9:15 AM	File folder
Outlook Files	10/11/2021 6:32 PM	File folder
Screenshots	10/23/2020 4:00 PM	File folder
SQL Server Management Studio	10/1/2020 2:52 PM	File folder
Visual Studio 2015	7/22/2021 9:04 PM	File folder
Visual Studio 2017	10/1/2020 2:56 PM	File folder
CDCB.postman_environment.json	10/11/2021 6:28 PM	JSON File

No preview available.

File name: CDCB.postman\_environment.json

All Files (\*.\*)

Open Cancel


Cancel

## MANAGE ENVIRONMENTS



An environment is a set of variables that allow you to switch the context of your requests. Environments can be shared between multiple workspaces. [Learn more about environments](#)

CDCB

 Share

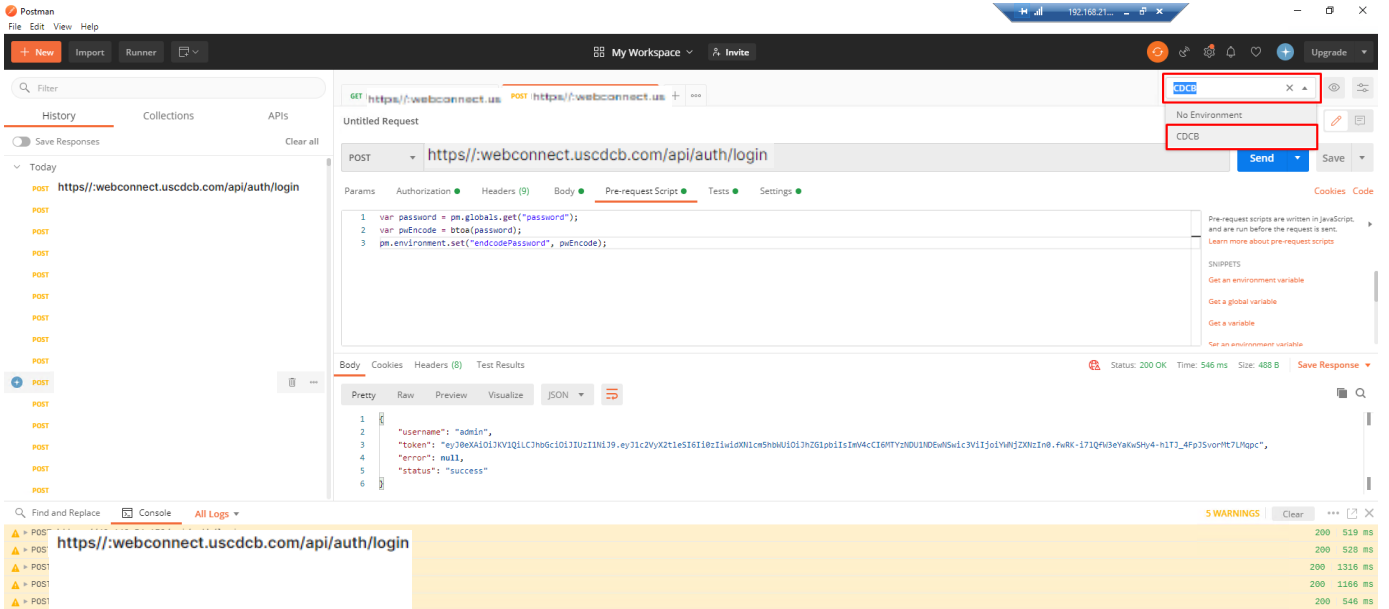
  

Globals

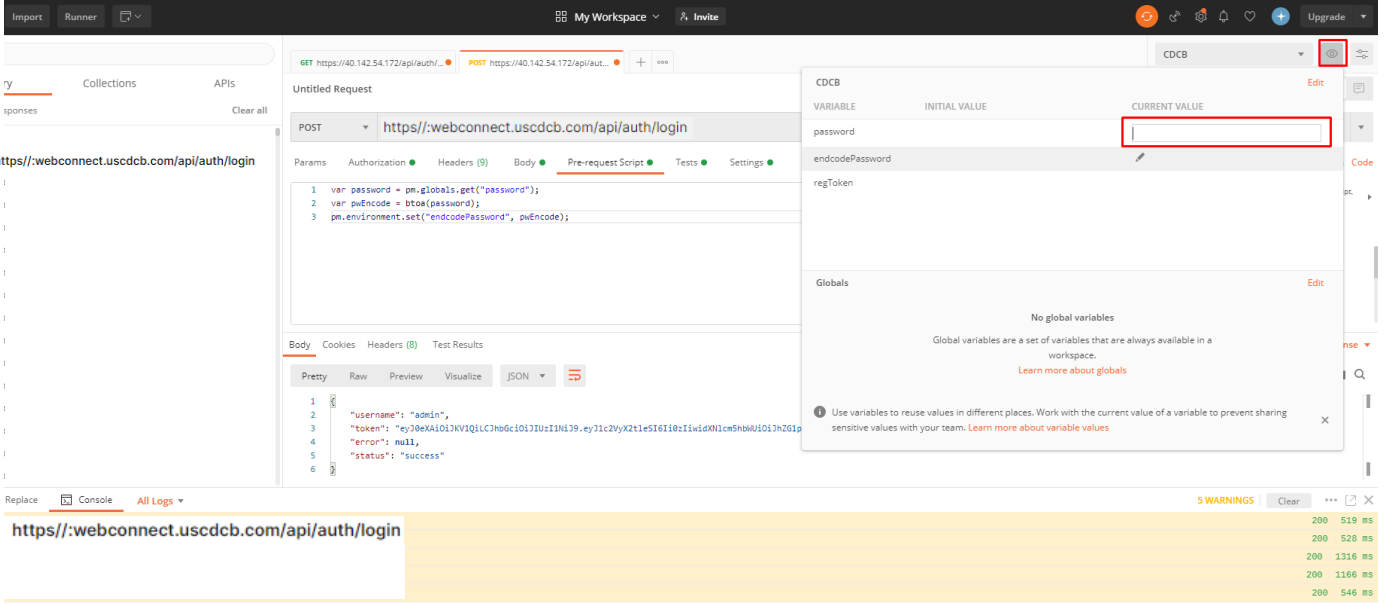
Import

Add

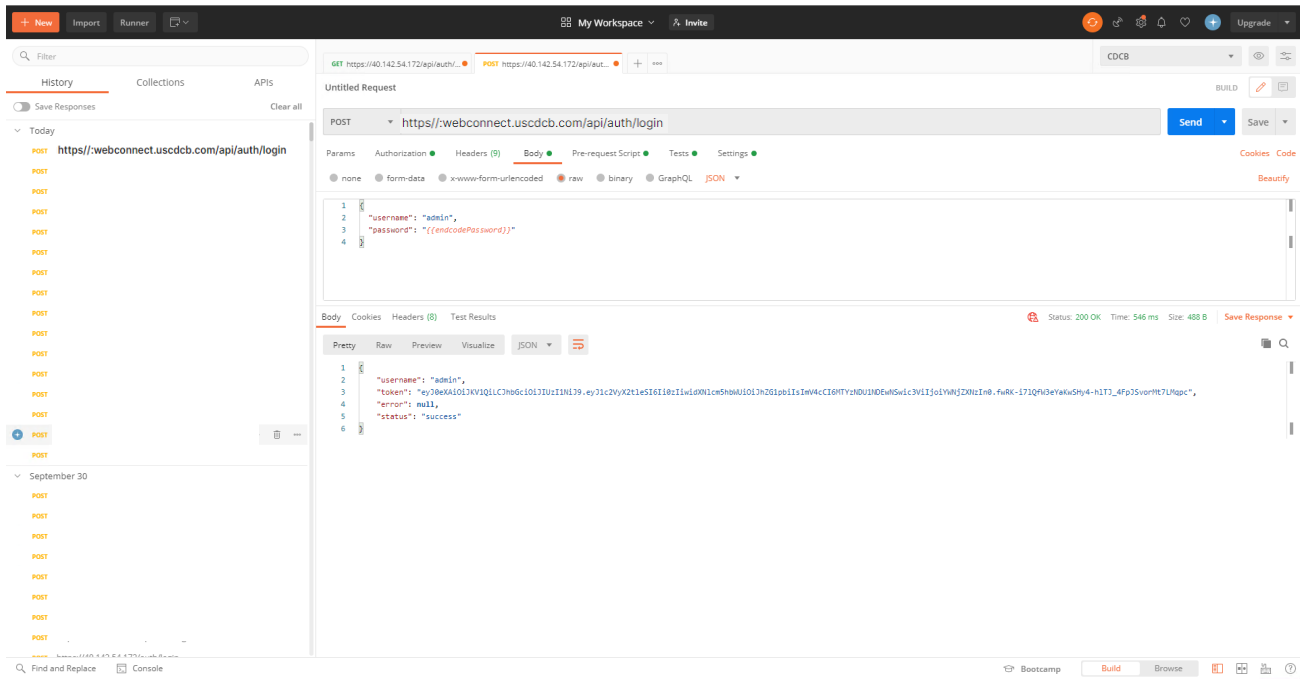
Select CDCB environment in dropdown list



**Fill out password:**



- **Step 2:** Setup login request with **post** method, input `{{ endcodePassword }}` (the variable that defined above) to "password" property:

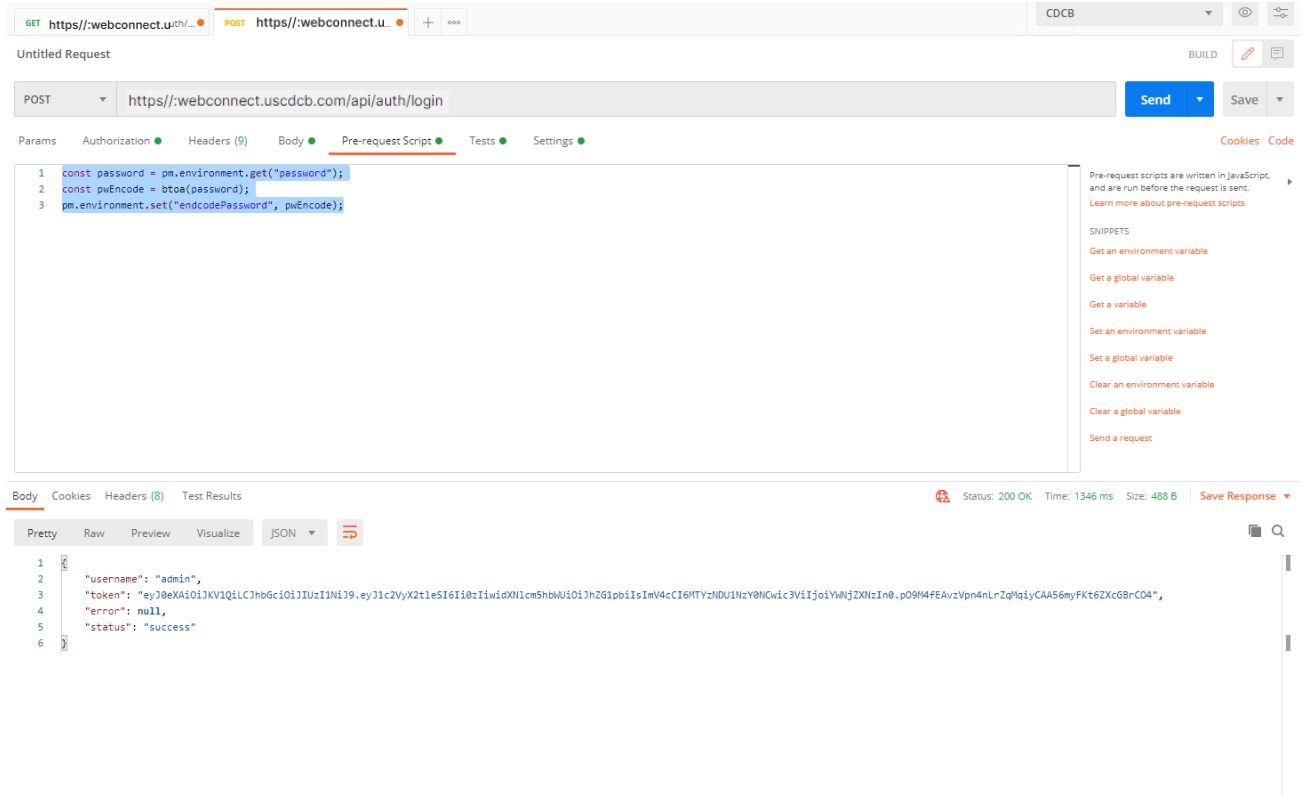


- **Step 3:** Add tab Pre-request Script, add code encode base64 as below:

```

const password = pm.environment.get("password");
const pwEncode = btoa(password);
pm.environment.set("endcodePassword", pwEncode);

```



- **Step 4:** Add tab Tests, get response assigned to regToken

```
if(responseCode.code === 200){
  const jsonData = JSON.parse(responseBody)
  pm.environment.set("regToken", jsonData.token)
}
```

The screenshot displays the Postman REST client interface. At the top, the request method is set to POST and the URL is `https://webconnect.uscdcb.com/api/auth/login`. The 'Tests' tab is active, showing the following JavaScript code:

```
1 if(responseCode.code === 200){
2   const jsonData = JSON.parse(responseBody)
3   pm.environment.set("regToken", jsonData.token)
4 }
```

Below the code editor, the 'Body' tab is selected, showing the response in JSON format:

```
1 {
2   "username": "admin",
3   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1IjoiYm99M4FEAvzVpn4NzZqMqIyCAA56myFKt6ZXCGBrC04",
4   "error": null,
5   "status": "success"
6 }
```

The status bar at the bottom indicates a successful response with a status code of 200 OK, a response time of 1346 ms, and a size of 488 B.

- **Step 5:** The result returns the token, then we can use this token to test the APIs.



GET https://webconnect.u... POST https://webconnect.u... CDCB

Untitled Request BUILD Code

POST https://webconnect.uscdcb.com/api/auth/login Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

```

1 const password = pm.environment.get("password");
2 const pwEncode = btoa(password);
3 pm.environment.set("endcodePassword", pwEncode);

```

Pre-request scripts are written in JavaScript, and are run before the request is sent. [Learn more about pre-request scripts](#)

SNIPPETS

- Get an environment variable
- Get a global variable
- Get a variable
- Set an environment variable
- Set a global variable
- Clear an environment variable
- Clear a global variable
- Send a request

Body Cookies Headers (8) Test Results Status: 200 OK Time: 1346 ms Size: 488 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "username": "admin",
3   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2tleSI6Ii0zIiwiaWF0IjoiYmVhZG1iLn99",
4   "error": null,
5   "status": "success"
6 }

```

CDCB Edit

VARIABLE	INITIAL VALUE	CURRENT VALUE
password		[REDACTED]
endcodePassword		[REDACTED]
regToken		eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2tleSI6Ii0zIiwiaWF0IjoiYmVhZG1iLn99

Globals Edit

No global variables

Global variables are a set of variables that are always available in a workspace. [Learn more about globals](#)

**i** Use variables to reuse values in different places. Work with the current value of a variable to prevent sharing sensitive values with your team. [Learn more about variable values](#)

