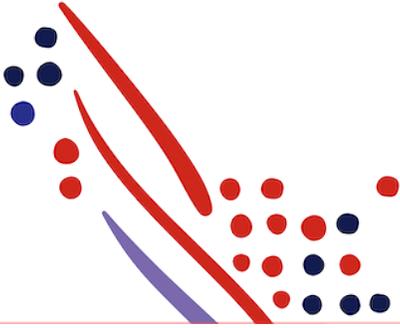


Guide

Making your first API call with POSTMAN

Published on
Nov 12, 2019, 10:11 PM

Last modified
Jan 19, 2024, 12:32 AM



ADP Copyright Information

ADP, the ADP logo, and Always Designing for People are trademarks of ADP, Inc.

Windows is a registered trademark of the Microsoft Corporation.

All other trademarks are the property of their respective owners.

Copyright © 2024 ADP, Inc. ADP Proprietary and Confidential - All Rights Reserved. These materials may not be reproduced in any format without the express written permission of ADP, Inc.

These materials may not be reproduced in any format without the express written permission of ADP, Inc. ADP provides this publication "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. ADP is not responsible for any technical inaccuracies or typographical errors which may be contained in this publication. Changes are periodically made to the information herein, and such changes will be incorporated in new editions of this publication. ADP may make improvements and/or changes in the product and/or the programmes described in this publication.

Published on
Nov 12, 2019, 10:11 PM

Published on
Jan 19, 2024, 12:32 AM

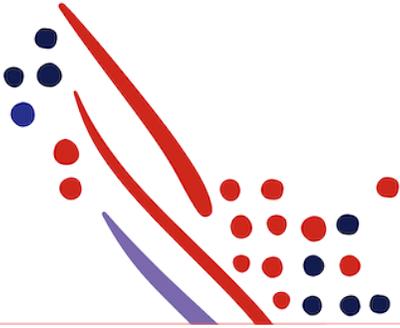


Table of Contents

Chapter 1

Prerequisite

Chapter 2

Installing and Preparing Postman

Chapter 3

Making your first call

- Requesting a Bearer Token

- Making an API Call with your bearer token

Chapter 4

Frequently Encountered Errors and Resolutions

- Received an HTTP 401 Error When Trying to Authenticate with ADP's Security Token Service

 - Error

 - Resolution

- Received an HTTP 401 Error from Other APIs

 - Error

 - Resolution

- Received an HTTP 403 Error

 - Error

 - Resolution

Chapter 5

Other useful link

Chapter 6

Using the Postman Collection

Chapter 7

Making your first API call with POSTMAN

Chapter 8

Create a Connection Object - Python

Chapter 9

Create a Connection Object - Java

Chapter 1

Prerequisite

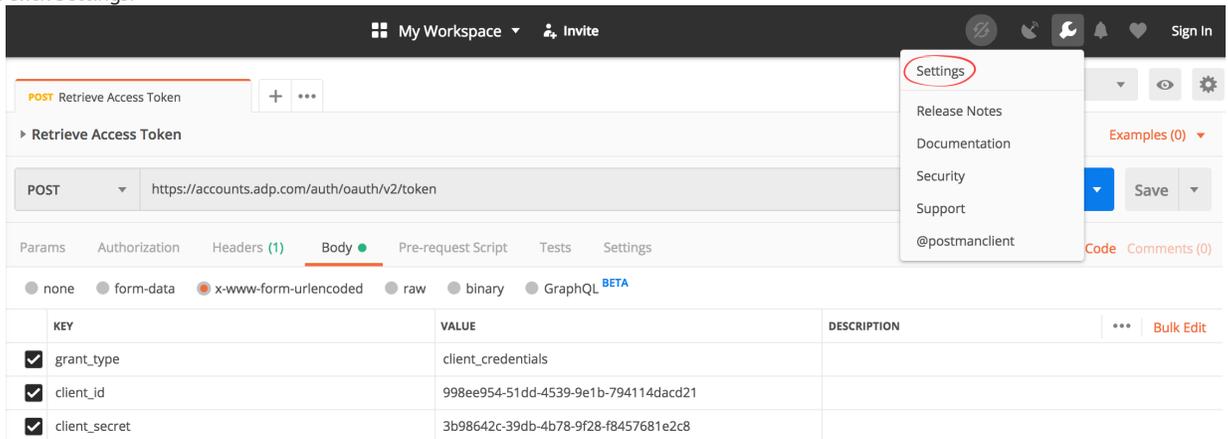
You need to have the following ready before making your first API call:

- Client ID and Client Secret: If you don't have this information, contact your client representative.
- Certificate Signing Request (CSR): For more detailed information, see the Certificate Signing Request: <https://developers.adp.com/articles/general/generate-a-certificate-signing-request>
- Allowed access to ADP API Gateway IP Addresses
 - 170.146.94.11
 - 170.146.94.12
 - 170.146.94.13
 - 170.146.94.14
 - 170.146.98.11
 - 170.146.98.12
 - 170.146.98.13
 - 170.146.98.14

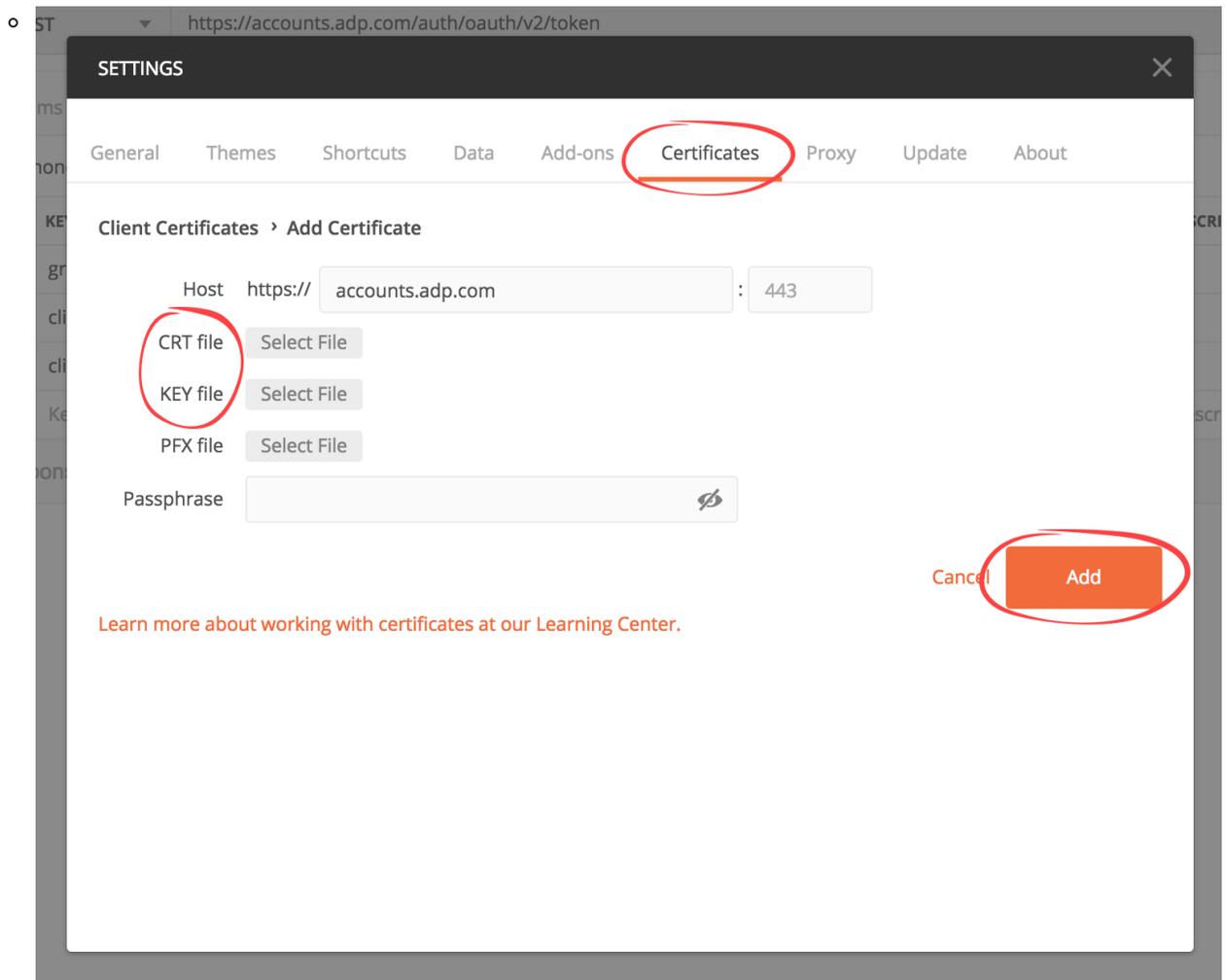
Chapter 2

Installing and Preparing Postman

1. Install Postman from <https://www.getpostman.com/apps>
2. Using the following steps, add your ADP issued .PEM file and your .KEY file (as part of the CSR process) into Postman for the domains. You can use the instructions found at <https://www.getpostman.com/docs/postman/sending-api-requests/certificates>
 1. Click Settings.

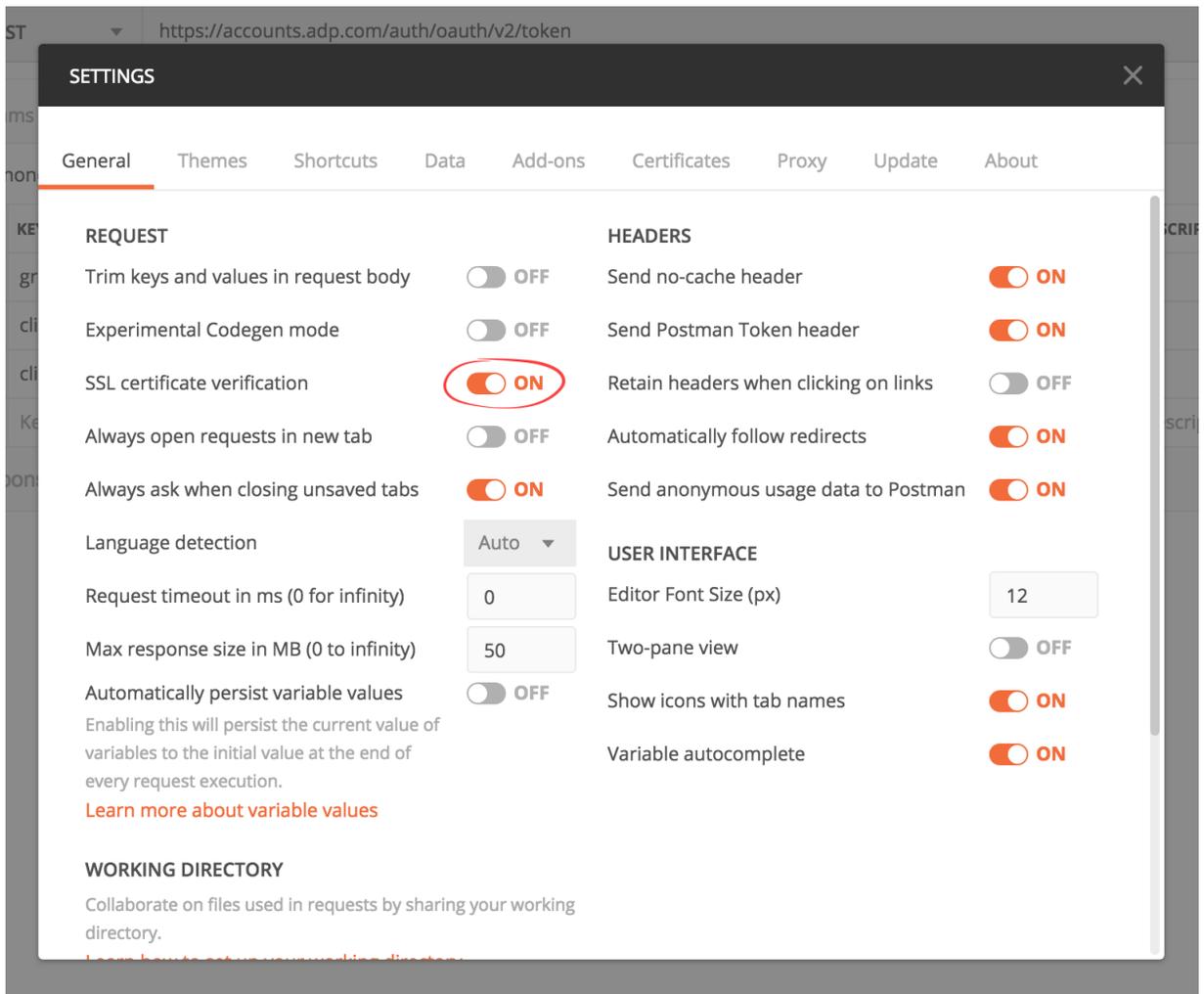


2. Click **Certificates**, add host: `https://accounts.adp.com` and select the **CRT file** (certificate) and **KEY file**. Then, click **Add**. Repeat the process for `https://api.adp.com`



1. On the **Settings** page, turn the **SSL certificate verification** on.

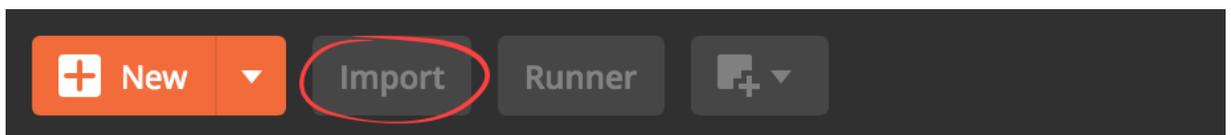
If you're not prompted to select a Secure Sockets Layer (SSL) certificate, and this is your first request in this Postman session, you should check to make sure your SSL >certificate is installed correctly. After you have selected the proper certificate, you will not be prompted for a certificate again until you start a new Postman session. If you know your SSL certificate is installed correctly, close Postman and open the Google Chrome browser and try your request again.



2. (Optional) Import the ADP APIs.

Postman allows you to store a collection of APIs and share them with others. ADP will continue to share sample collections on GITHUB.

For example, you can find the ADP Workforce Now collection by clicking [here](#). When at the location, download the file and click **Import** to import the list file.



Rate this article

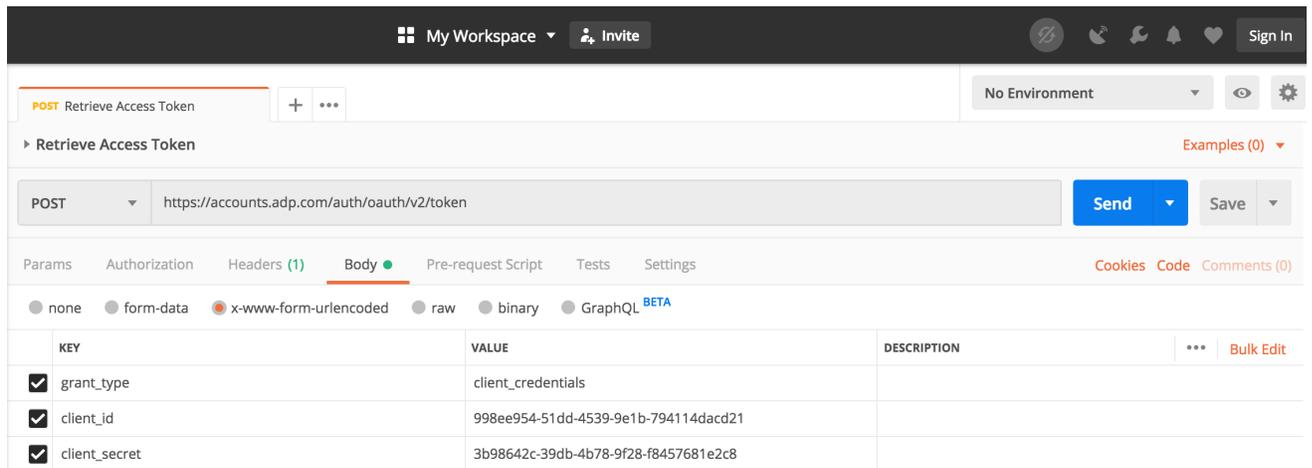
Chapter 3

Making your first call

Requesting a Bearer Token

Each request to one of ADP's APIs needs to be accompanied by an Authorization header containing a bearer token issued by the ADP Security Token Service.

1. In Postman, open the **Token Request**. To expose the headers, click **Headers**.



2. If your POST request is successful, you will receive an HTTP 200 from the server with your token in the body of the response. Copy the access_token value.



3. You will receive an Access Token in response, which is valid for one hour. The same can be used to make API calls by adding the following header:

Authorization: Bearer{accessToken}

Postman allows you to store a collection of APIs and share them with others. ADP will continue to share sample collections on GITHUB.

For example, you can find the ADP Workforce Now collection by clicking [here](#). When at the location, download the file and click **Import** to import the list file.

Making an API Call with your bearer token

If you already have an API collection, you can select an API from a collection. The following sample shows selecting the GET HR - Worker (List) API and making the first call.

1. Open the **HR > Worker (List)** request and click **Headers** to expose the headers.
2. Paste your bearer token into the Authorization header and click Send.

Note

Remember to leave the Bearer and a space to the left of your token. Since Postman is a browser-based application, the browser caching mechanism will save responses to the cache. To ensure that changes you make in ADP applications are reflected in Postman, we recommend placing a cache-buster into the query string between each request.

For example, preventCache=timestamp, where timestamp is a unique value such as the current time. You won't need to do this when you're building your application because browser-based caching won't come into play.

If your request was successful, you'll receive an HTTP 200 message from the server within a few records. The following sample shows a response of the GET HR - Worker (List) API request:

Frequently Encountered Errors and Resolutions

Received an HTTP 401 Error When Trying to Authenticate with ADP's Security Token Service

Error

An HTTP 401 error is returned from the ADP Security Token Service when you fail to provide valid credentials in the request header.

Resolution

Take the following suggested troubleshooting steps before contacting your ADP Representative for assistance:

- Make sure you have base-64 encoded in your Open Authorization 2 (OAuth2) Client ID and Client Secret before sending them to the Authorization header.
- Make sure you are also base-64 encoding the colon (":") between your OAuth2 Client ID and Client Secret.
- Make sure there is a space between Basic and your base-64 encoded credentials.
- Make sure you are including your ADP-issued SSL Certificate in the request.

Received an HTTP 401 Error from Other APIs

Error

An HTTP 401 error is returned from APIs other than the ADP Security Token Service when you fail to provide a valid bearer token in the request header.

Resolution

Take the following suggested troubleshooting steps before contacting your ADP Representative for assistance:

- Make sure you have added an Authorization header to your request along with the bearer token you fetched from the ADP Security Token Service.
- Make sure you have a space between the Bearer and the token you are using in the Authorization header.
- Check the body of the response for an "expired token" message. If that message is present in the response, fetching a fresh bearer token and resubmitting your request should resolve the issue.

Received an HTTP 403 Error

Error

An HTTP 403 error is returned from the server when the bearer token you provided is valid, but you are not authorized to access the resource you have requested.

Resolution

First, confirm that you have not made a mistake in the request Uniform Resource Indicator (URI). If the request URI is correct, contact your ADP Representative to request access to the URI in question.

- [API Common Exceptions and Tips for Handling](#)
- Sample Postman collection:
https://github.com/adpllc/marketplace-sample-payloads/blob/master/wfn/ADP%20WFN%20APIs.postman_collection.json

Chapter 6

Using the Postman Collection

1. Navigate to <https://github.com/adpllc/marketplace-sample-payloads/tree/master/wfn>.
2. Copy the entire JavaScript Object Notation (JSON), which is highlighted as follows:

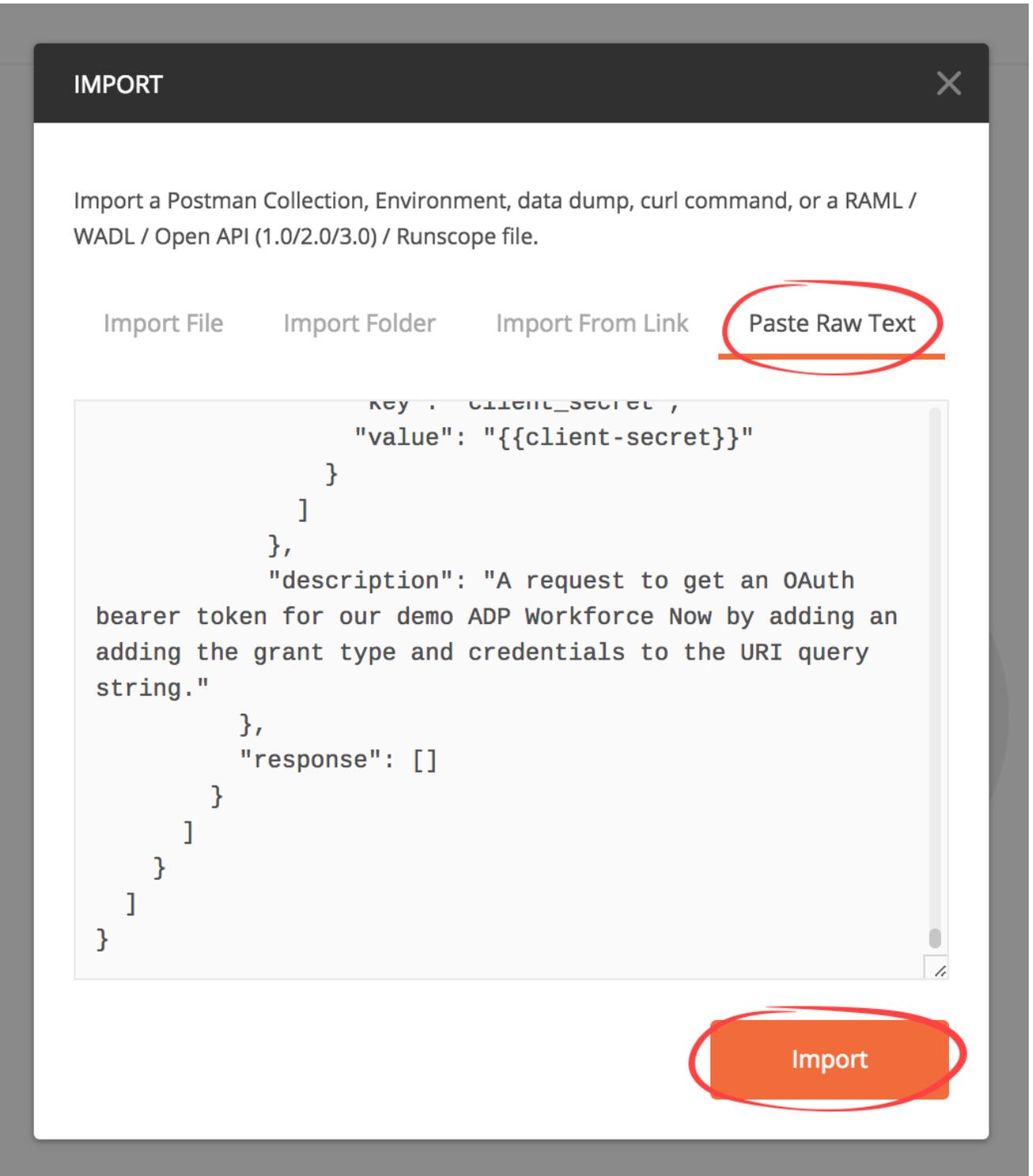
The screenshot shows a GitHub repository page for `adpllc / marketplace-sample-payloads`. The file path is `marketplace-sample-payloads / wfn / ADP WFN APIs.postman_collection.json`. The file content is as follows:

```

1  {
2      "info": {
3          "name": "ADP APIs",
4          "_postman_id": "dd97c527-5c24-59dc-24a7-49c8e2a28e86",
5          "description": "ADP API endpoints.",
6          "schema": "https://schema.getpostman.com/json/collection/v2.0.0/collection.json"
7      },
8      "item": [
9          {
10             "name": "Core",
11             "description": "",
12             "item": [
13                 {
14                     "name": "Event Notification Message V1 (Latest)",
15                     "request": {
16                         "method": "GET",

```

3. Select **Postman > Import Tab > Paste Raw Text**.



4. Click **Import**

Chapter 7

Making your first API call with POSTMAN

Create a connection based on your grant type using the connection factory

```
# Using the new configuration object create a connection
acConnection = ADPAPIConnectionFactory().createConnection(AuthorizationCodeConfiguration)
```

Chapter 8

Create a Connection Object - Python

Create a connection based on your grant type using the connection factory

```
# Using the new configuration object create a connection
acConnection = ADPAPIConnectionFactory().createConnection(AuthorizationCodeConfiguration)
```

Chapter 9

Create a Connection Object - Java

Create a connection based on your grant type using the connection factory

```
//get ADP connection using Connection factory and configuration
authorizationCodeConnection = (AuthorizationCodeConnection)
CONNECTION_FACTORY_INSTANCE.createConnection(authorizationCodeConfiguration);

//get Url from authorization connection
authorizationUrl = "redirect:" + authorizationCodeConnection.getAuthorizationUrl();

//CALL BACK HANDLER
//get code value
String requestCode = (String) request.getParameter("code");

//map request code to connection configuration
((AuthorizationCodeConfiguration) authorizationCodeConnection.getConnectionConfiguration()).setAuthorizationCode(requestCode);
```