Hands-on Lab: Base Acronis Cyber Platform API operations with Postman

Hands-on Lab Directory

File name	File description
`Acronis Cyber Platform Collection.postman_collection.json`	Acronis Cyber Platform Collection to import into the Postman application.
`Acronis Cyber Platform Development.postman_environment.json`	Acronis Cyber Platform Development Environment to import into the Postman application.
`LICENSE`	The license for the code. It's MIT license.
`README.md`	This file.

The Acronis Cyber Platform API general workflow

#	Operation	When/Period	Prerequisites/Inputs

#	Operation	When/Period	Prerequisites/Inputs
1	Create an API client under which an integration will be authorized	Initially. Periodically if security policies require your company to regenerate all passwords each X months. Through the API or the Management Portal for ACC 9.0 and greater.	Login and password with a needed level of access in Acronis Cyber Cloud. Usually, it's a service Admin account under your company's Partner tenant in Acronis Cyber Cloud.
2	Issue an access token	1. Before the first API Call which is not connected to the authorization flow 2. Each time when your token is near to be expired or expired.	Your API Client credentials
3	Make API calls		An access token issued using your API Client credentials

Prerequisites and basis information

To follow the hands-on lab manual, you need to have installed `Postman` version `7.20 or greater` and have the `Acronis Cyber Platform Development Environment` and `Acronis Cyber Platform Collection` imported to the installed `Postman`.

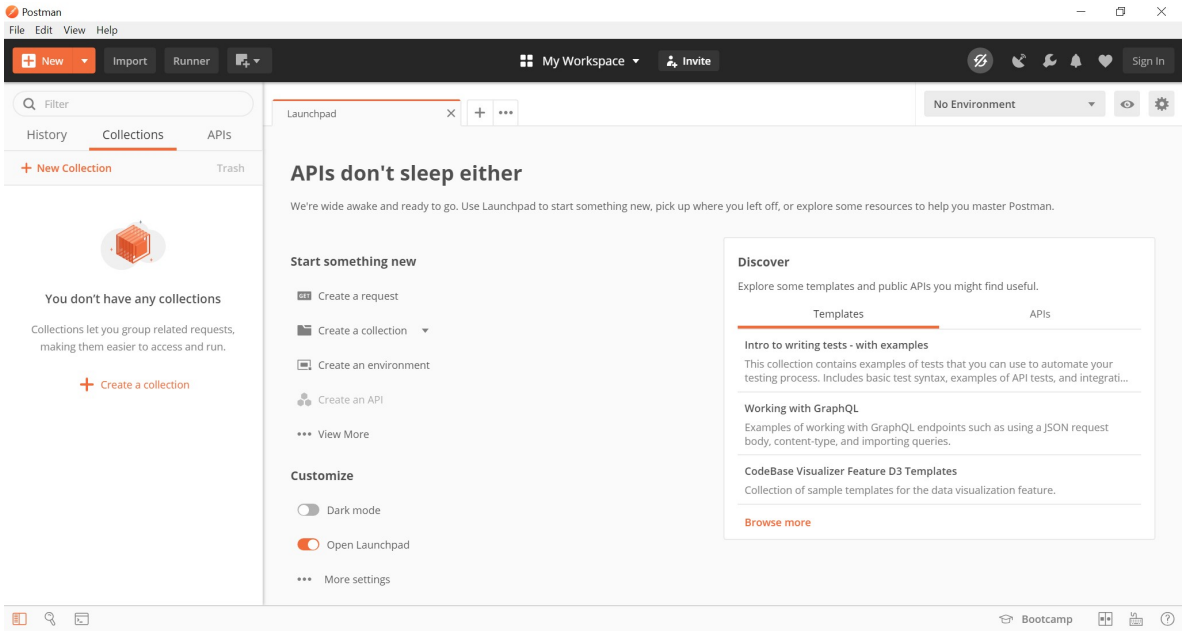
Step-by-step execution guide

1. Install `Postman` if needed.

> **You need to use an account with appropriate permissions on your device to install an application.**

    1. Download an installation package for your OS from the https://www.postman.com/downloads/ page. Windows, macOS and Linux are supported.

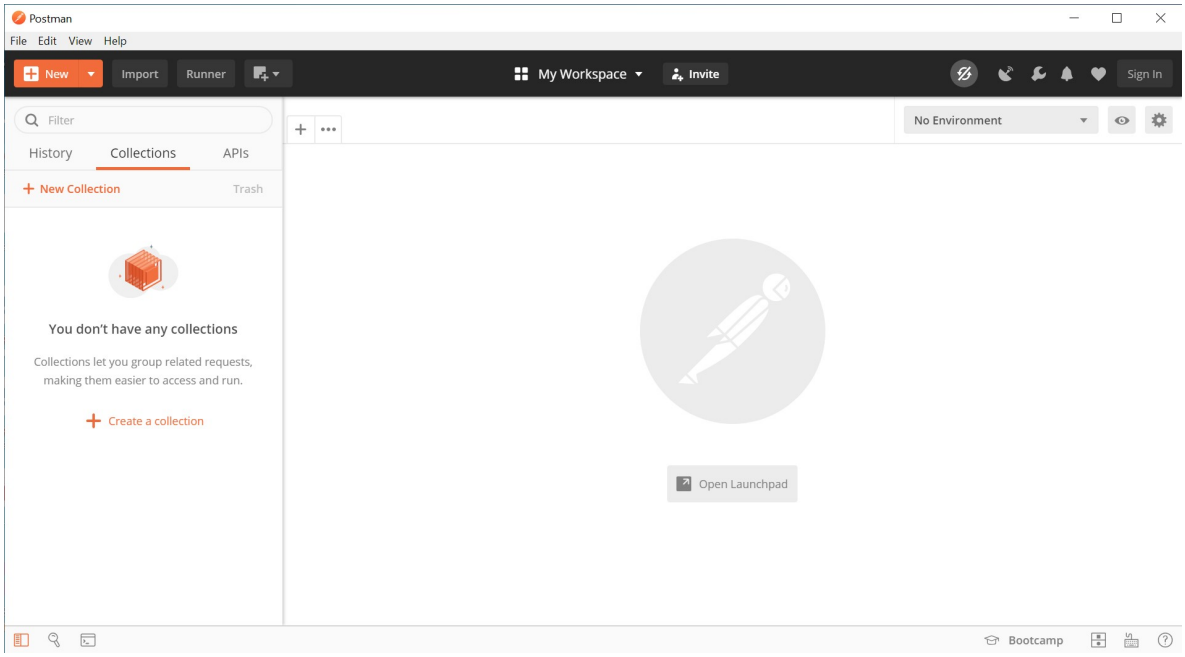
2. Double click on downloaded file and follow the instructions to install.



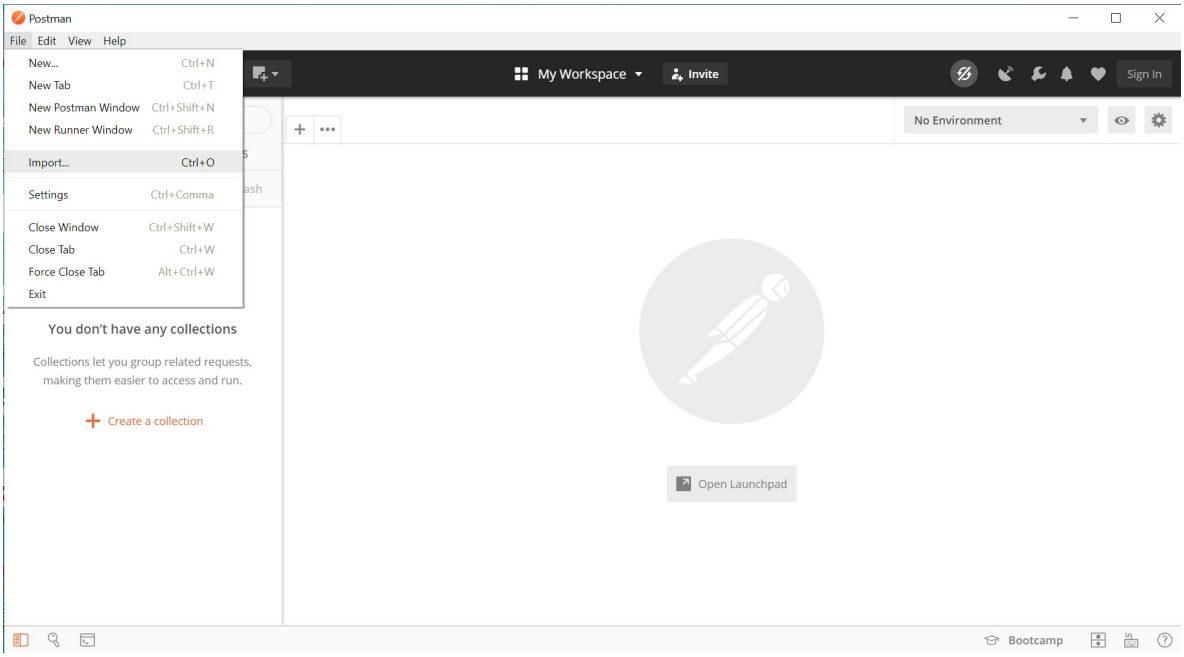
!!! note If you don't want to sync your account data between devices just skip account creation and work with `Postman` without it.

2. Import `Acronis Cyber Platform Development Environment` into the installed `Postman`.

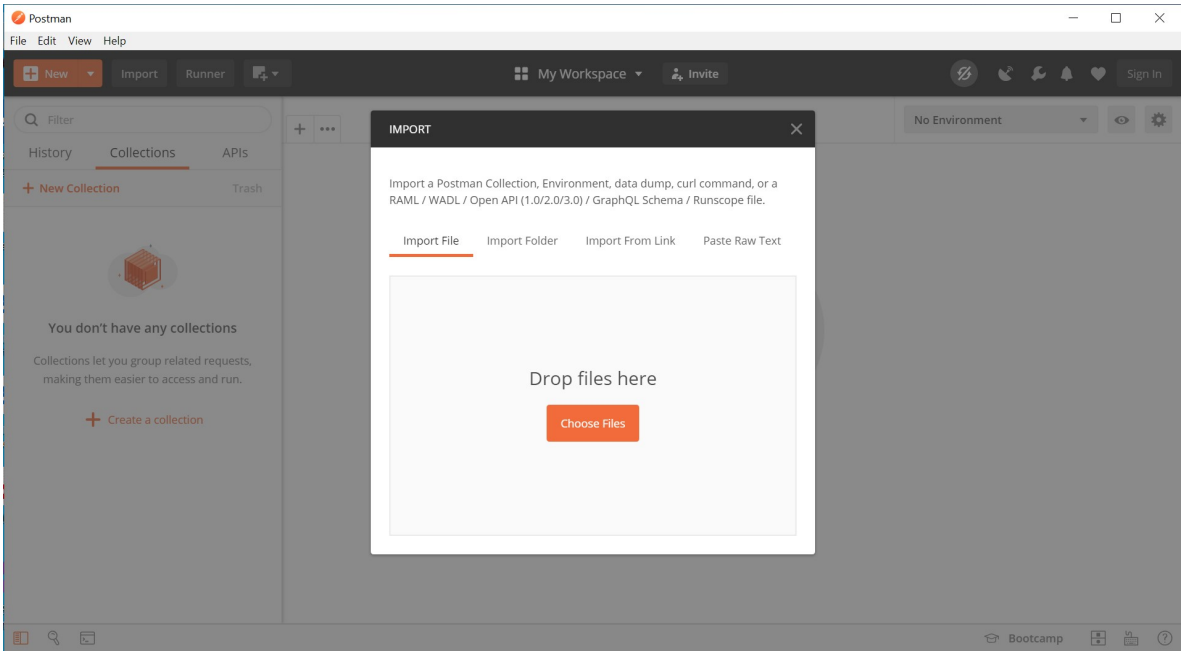
    1. Open `Postman` if needed.



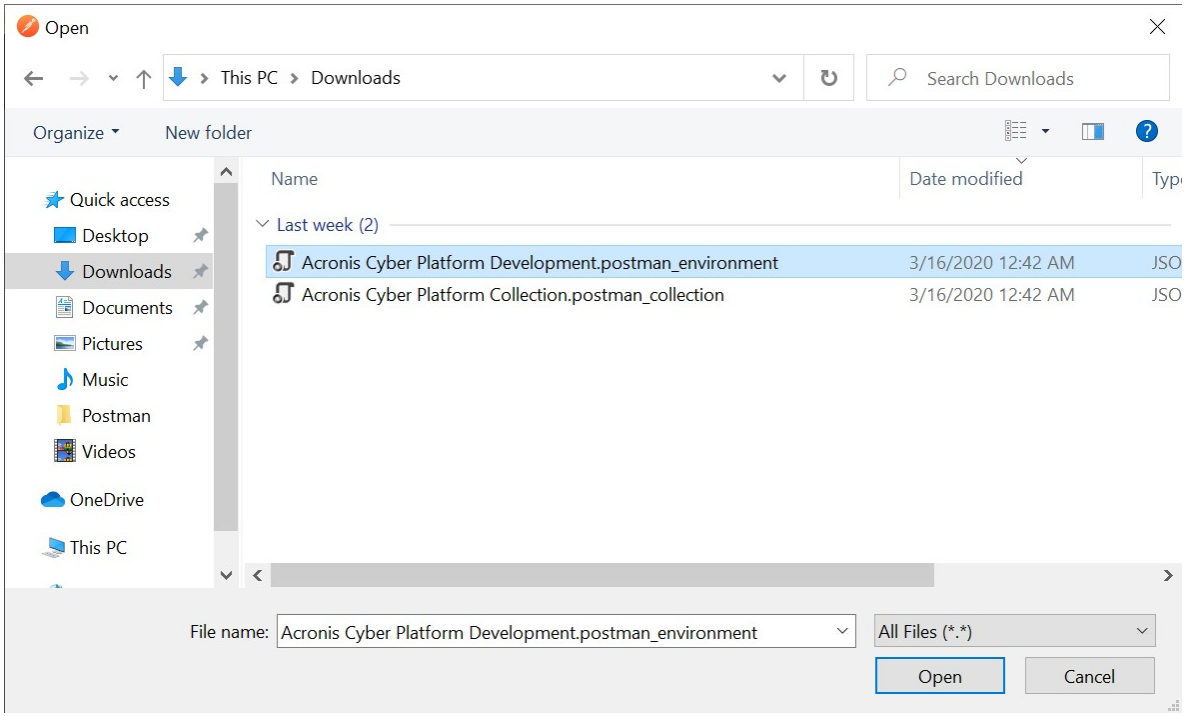
2. Got to `File` menu and select `Import`.



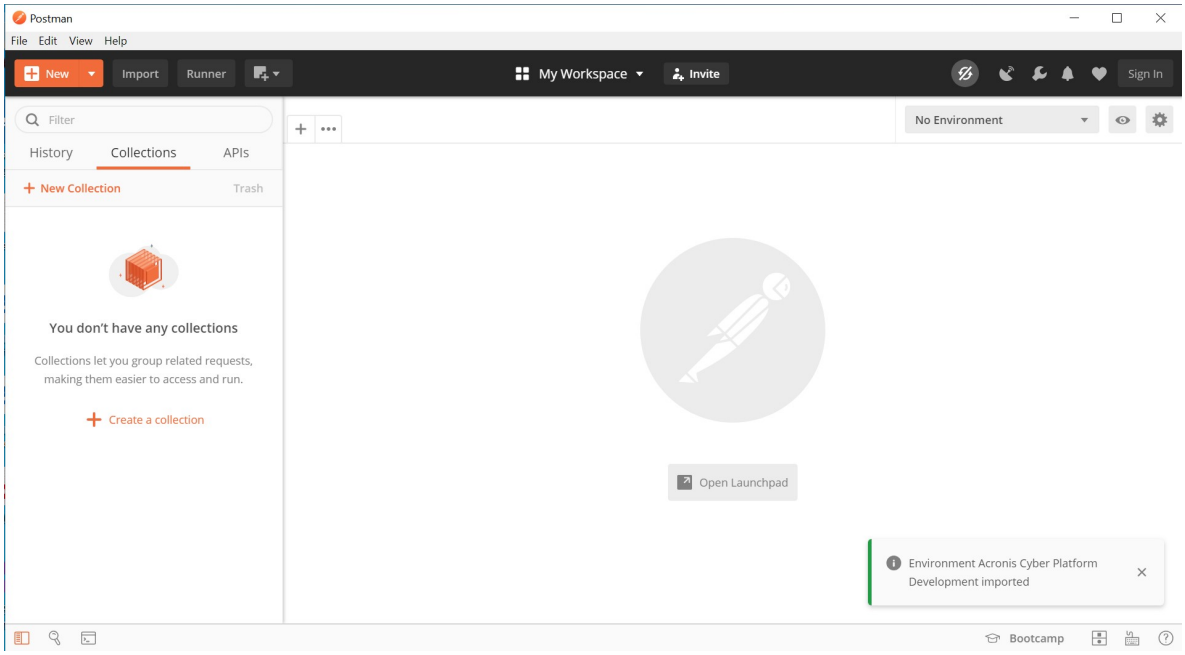
3. The import dialog opens.



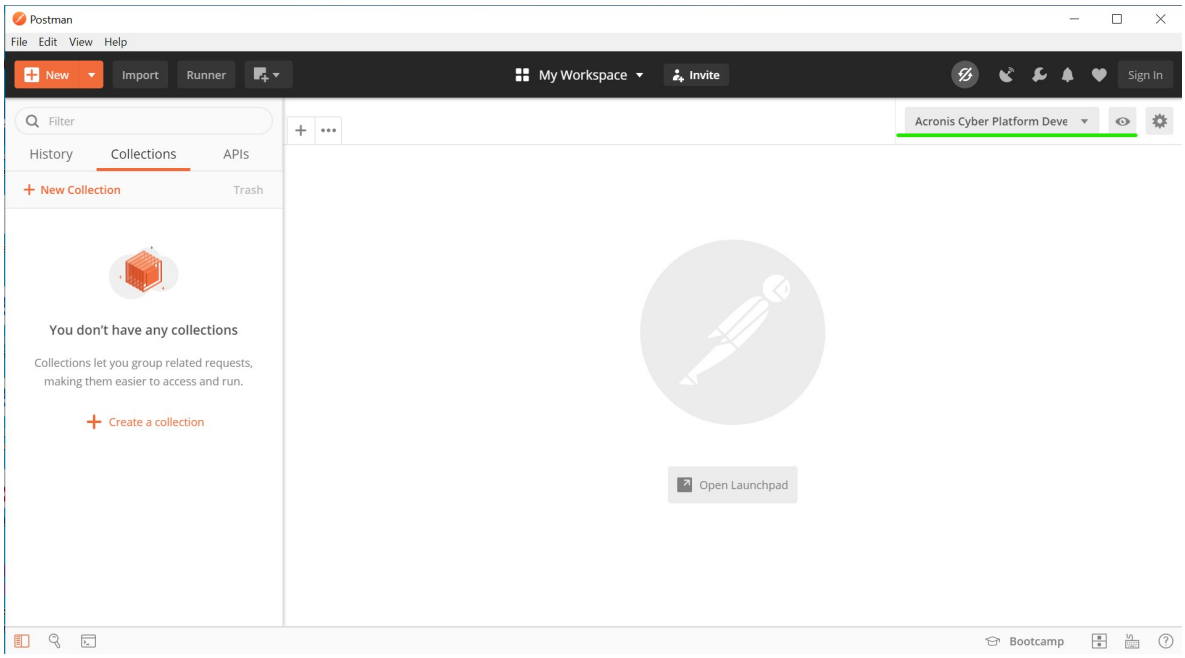
4. Press `Choose File`, the Choose File dialog opens.



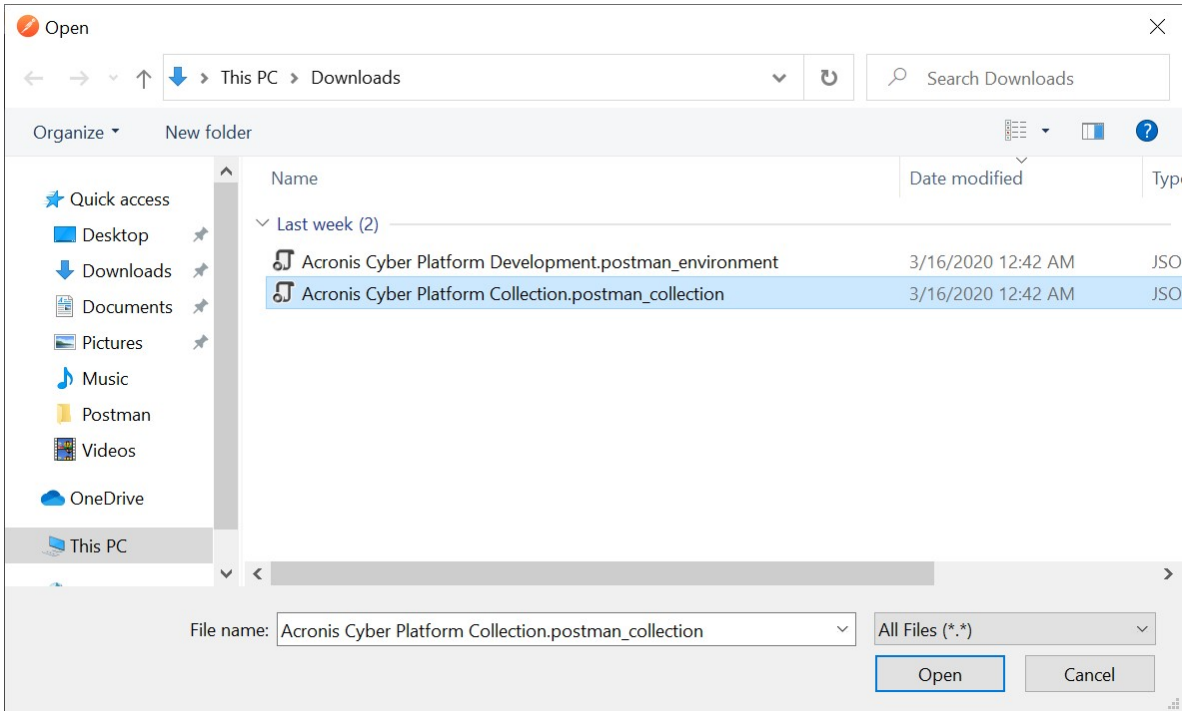
5. Select `Acronis Cyber Platform Development.postman_environment.json` and press `Open`. You should see a successful import pop-up.



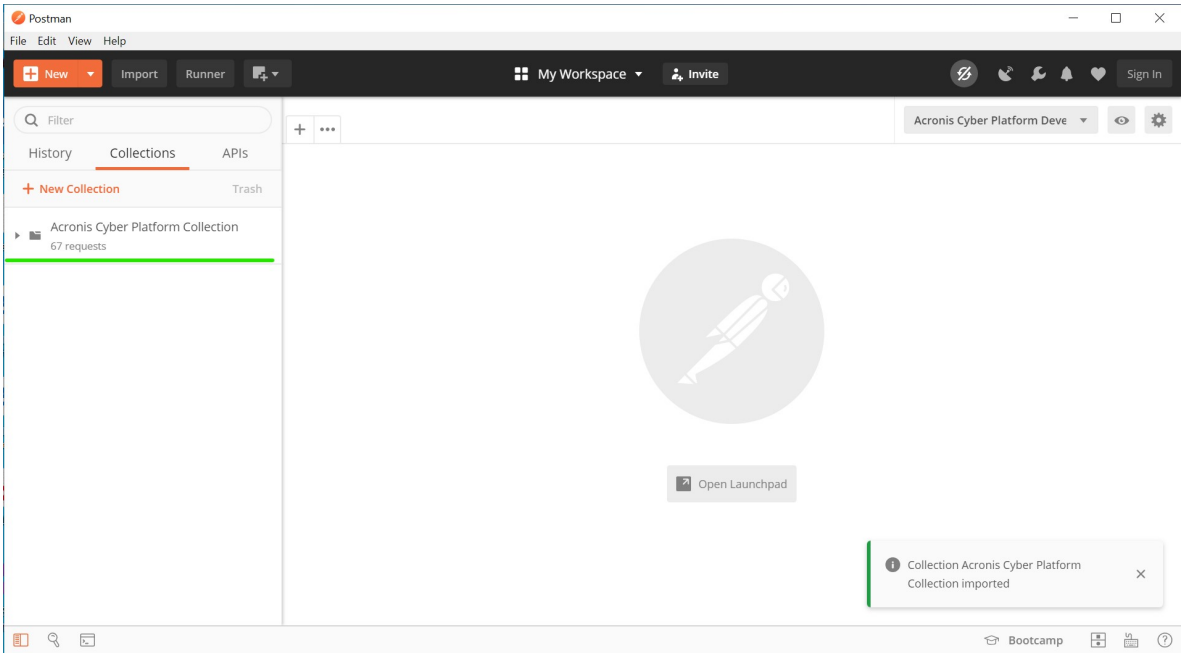
6. Check that the environment was successfully imported and select it drop-down list at top right corner of the `Postman` application.



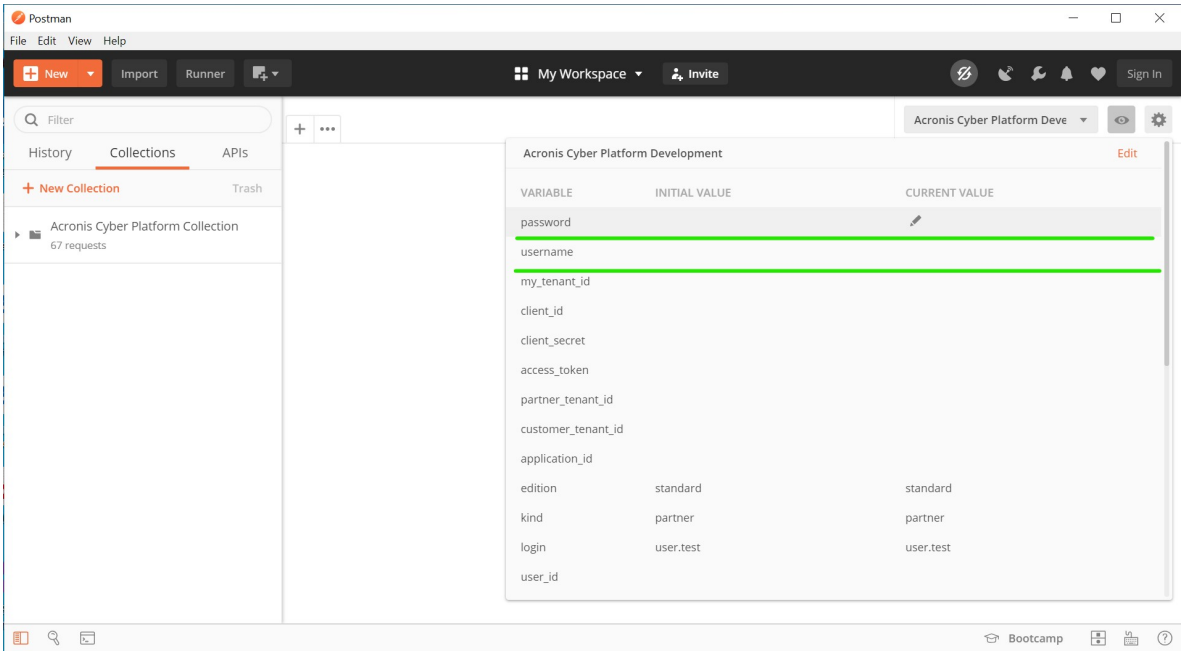
7. Then do the same for the `Acronis Cyber Platform Collection.postman_collection.json` file.

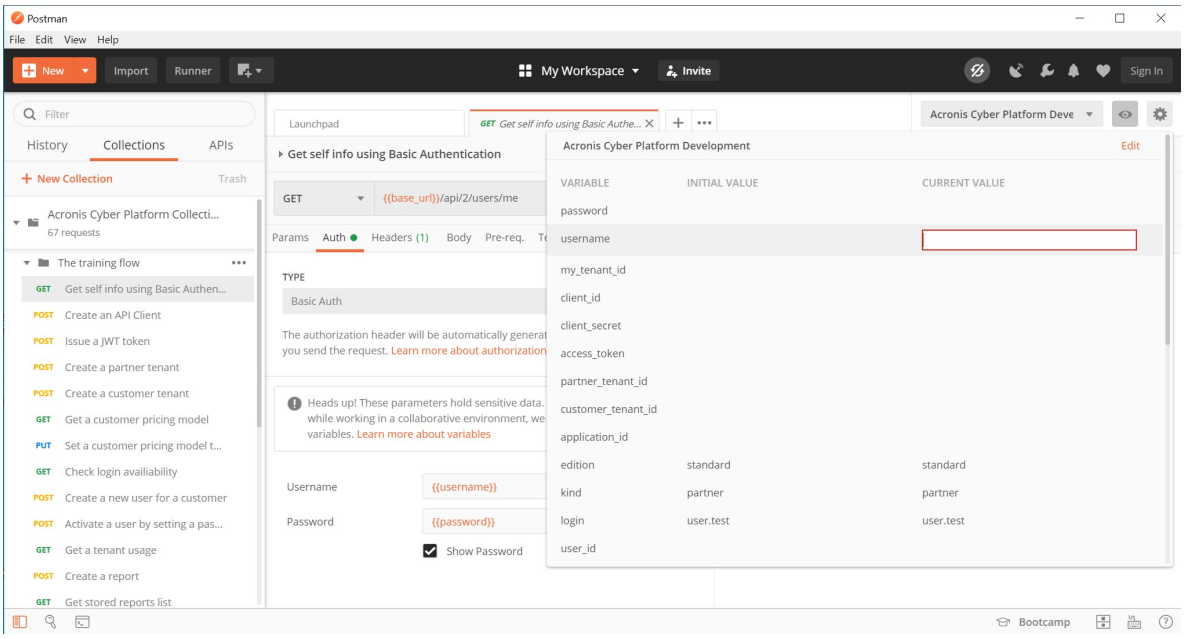


8. Check that the collection was successfully imported in the `Postman` application.

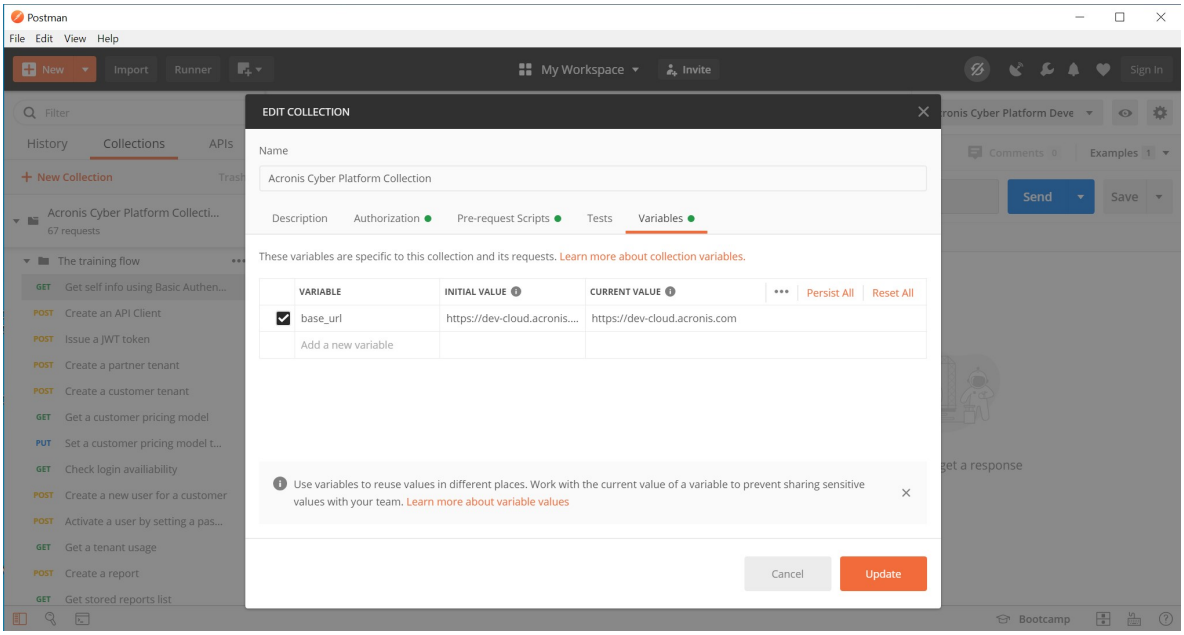


9. Open the environment to edit using an icon at the right of the drop-down list. Enter your username and your password for basic authentication.

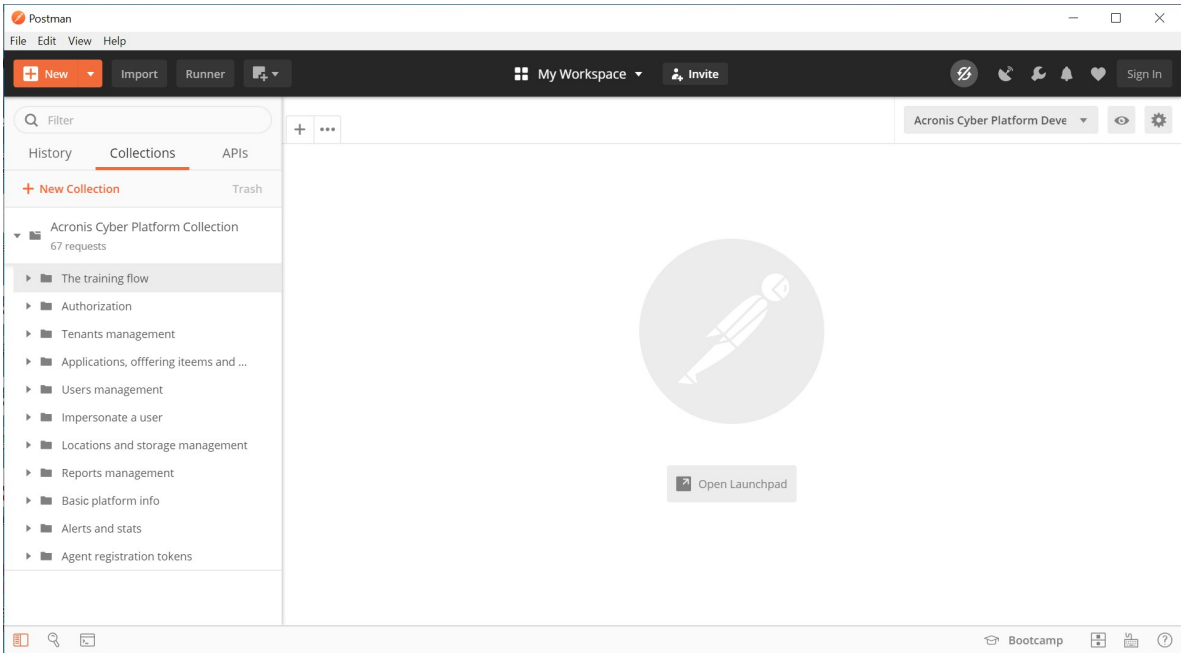




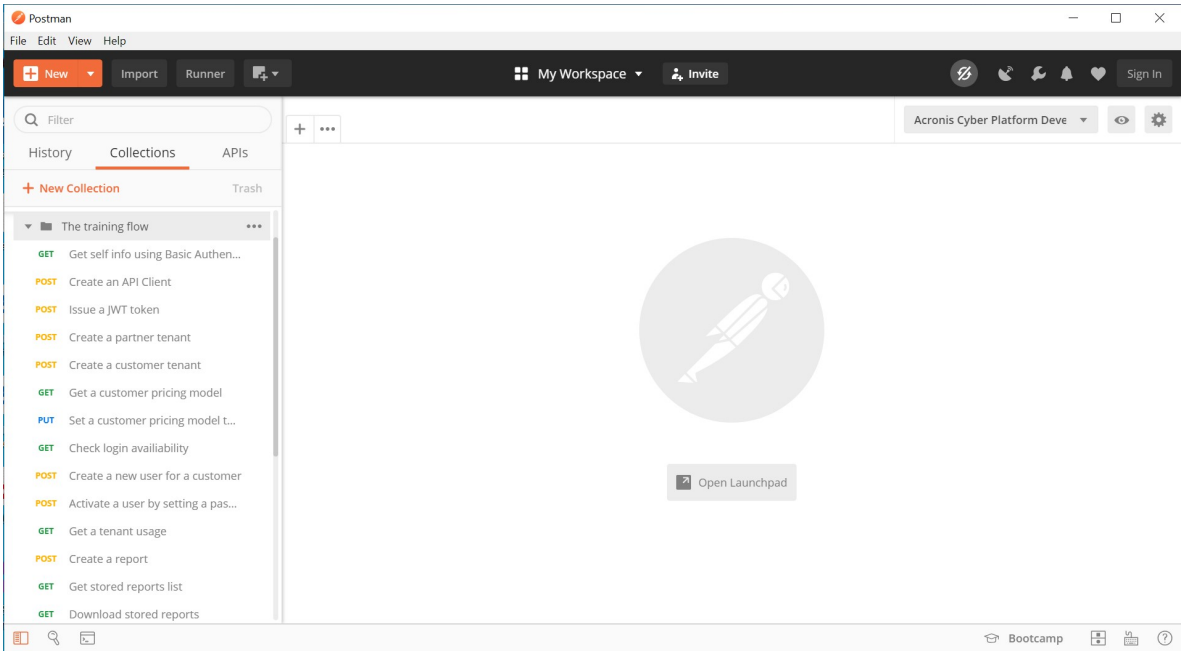
10.Open settings for the imported collection and set `base_url` variable. It's used for all API calls and identified your Acronis data center.



10. Expand the imported collection



11. Expand `The training flow` folder.



You are ready to start the hands-on lab.

Exercise 1: Create an API Client to access the API

Implementation details

A JWT token with a limited time to life approach is used to securely manage access of any API clients, like your scripts and integrations, for the Acronis Cyber Cloud. Using a login and password for a specific user is not a secure and manageable way to create a token, but technically it's possible. Thus, we create an API client with a client id and a client secret to use as credentials to issue a JWT token. To create an API Client, we call the `/clients` end-point with POST request specifying in the JSON body of the request a tenant we want to have access to. To authorize this the request, the Basic Authorization with user login and password for Acronis Cyber Cloud is used.

> **In Acronis Cyber Cloud 9.0 API Client credentials can be generated in the Management Portal.**

> **Creating an API Client is a one-time process. As the API client is used to access the API, treat it as credentials and store securely. Also, do not store the login and password in the scripts itself.**

In this Postman Collection it is expected that the Acronis Developer Sandbox is used. It is available for registered developers at Acronis Developer Network Portal. So the base URL for all requests (https://devcloud.acronis.com/) is used. Please, replace it with correct URL for your production environment if needed. For more details, please, review the Authenticating to the platform via the Python shell tutorial from the Acronis Cyber Platform documentation.

For demo purposes, this script issues an API client for a tenant for a user for whom a login and a password are specified. You should add your logic as to what tenant should be used for the API Client creation.

> **A generated client is inherited access rights from a user used for the generation but it's disconnected from them. You don't need to issue a new client even if the user account is removed from Acronis Cloud.**

> **Treat API Clients as a specific service account with access to your cloud. All internal security policies applied to your normal account operations should be in place for API Clients. Thus, don't create new API Clients if you don't really required and disable/delete unused API Clients through the Management Console or API Calls.**

> **You can receive a `client_secret` only once, just at the issue time. If you loose your `client_secret` further you must reset secret for the client through the Management Console or API Calls. Please, be aware, that all the tokens will be invalidated.**

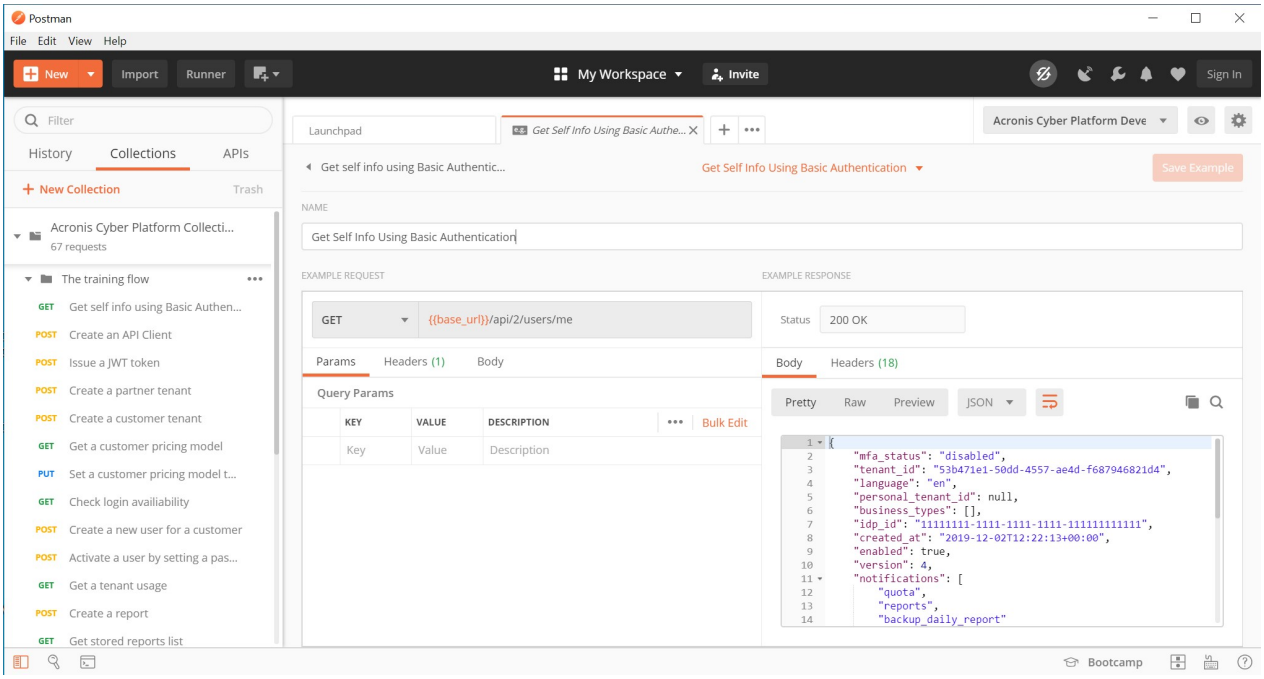
> **You need to securely store the received credentials. Please remember to implement secure storage for your client credentials.**

Step-by-step execution and checks

1. Be sure that you've fully finished *prerequisites* step-by-step guide.

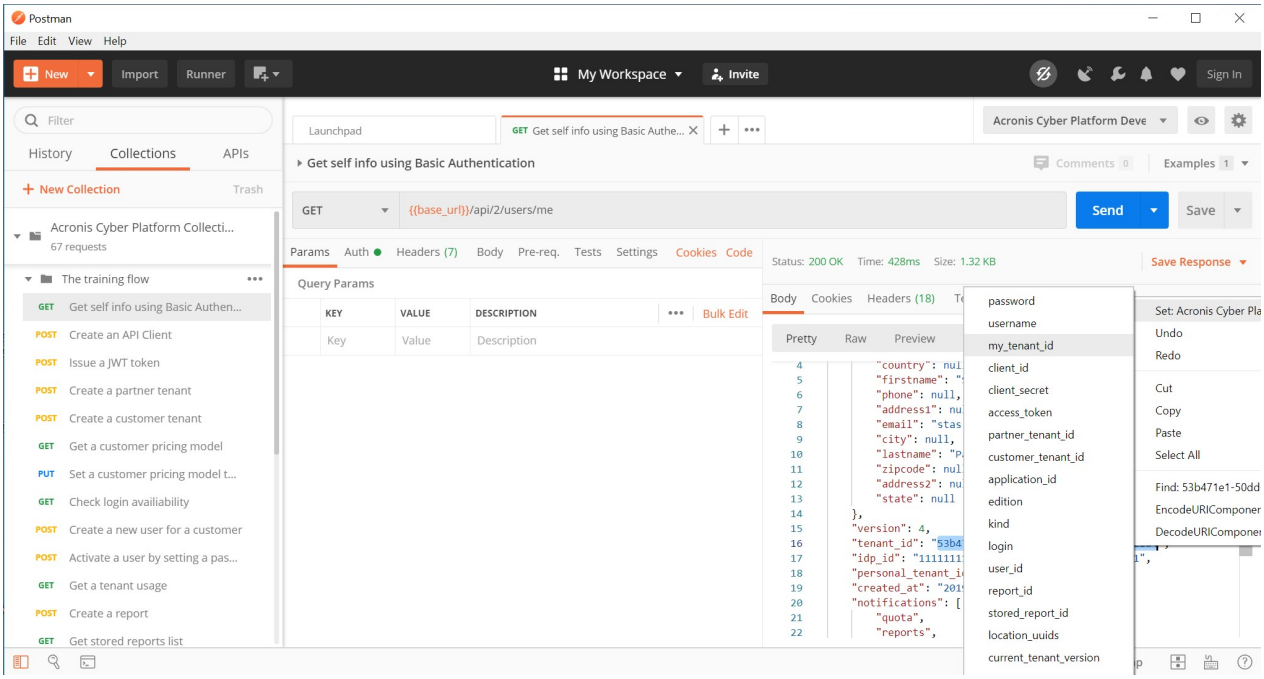
> **For simplicity, this flow was built for the case when `2FA` not enabled. In case if `2FA` enabled, you need either use the ACC 9.0 Management Console to create an API Client and skip firsts steps of the tutorial, or disable `2FA` before create an API client and enable back after.**

2. Open `Get self info using Basic Authentication` request in the `Postman` and click `Send`.

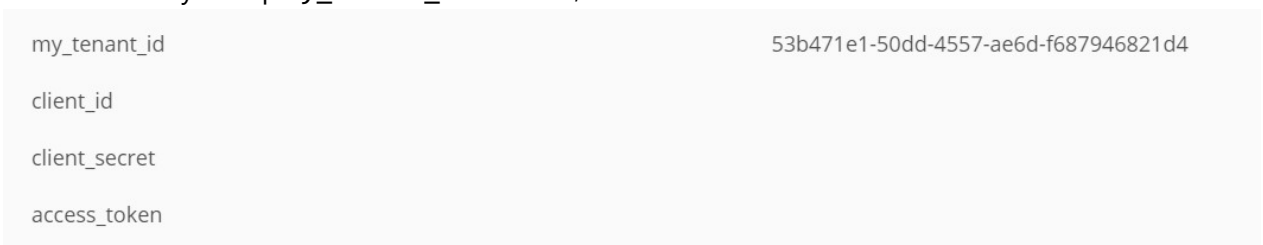


You receive information regarding authenticated user.

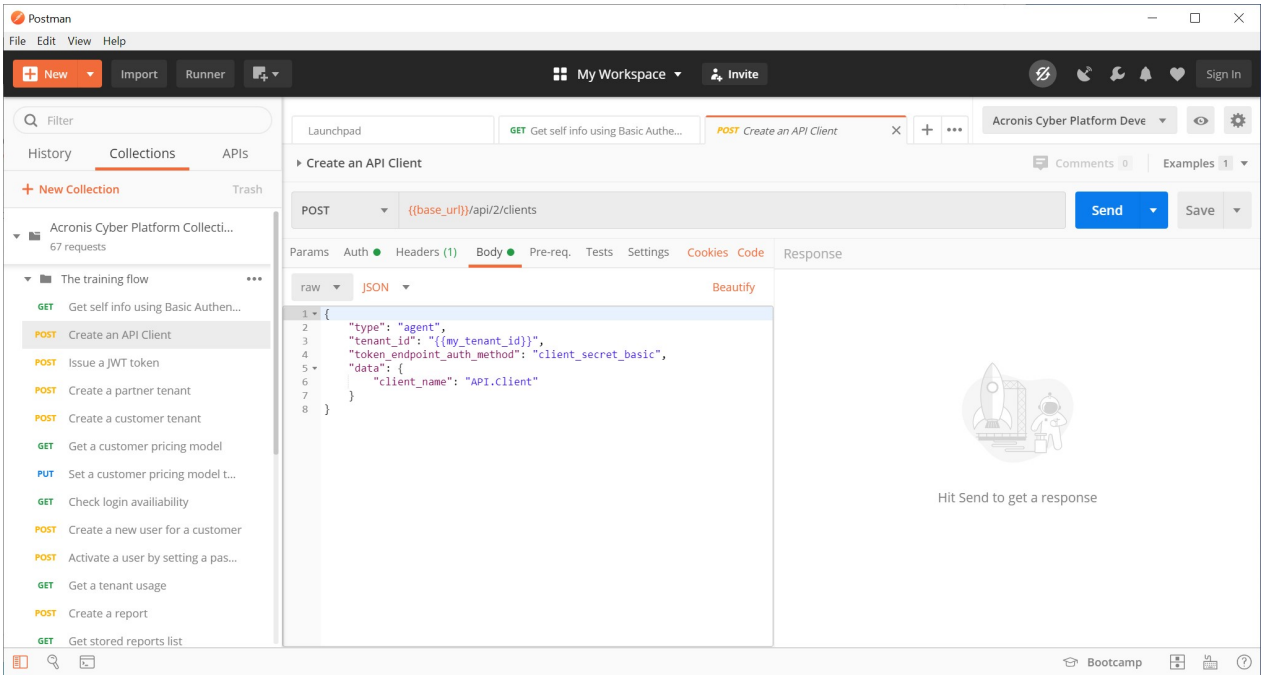
3. We create an API Client for a specific tenant. For our hands-on lab we create a client for the your root tenant. So you need to put `tenant_id` to the `my_tenant_id` variable. To do this, select `tenant_id` value in the JSON response without quotas and press right mouse button for PC. Then select `Set: Acronis Cyber Platform Development` from the drop-down menu and `my_tenant_id` from the cascade menu.



4. We successfully set up `my_tenant_id` variable, check it in the environment.

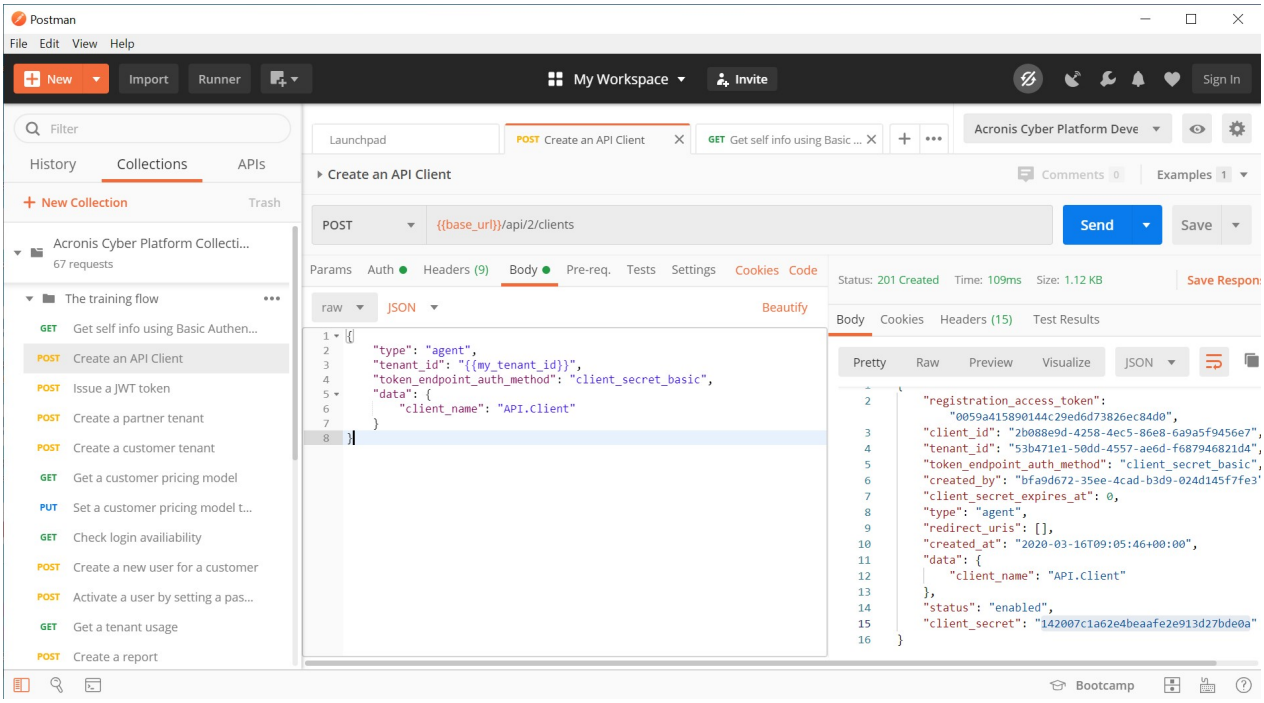


5. Open `Create an API Client` request and then, select `Body` tab. You can see parametrized JSN body request to create an API Client. Click `Send`.



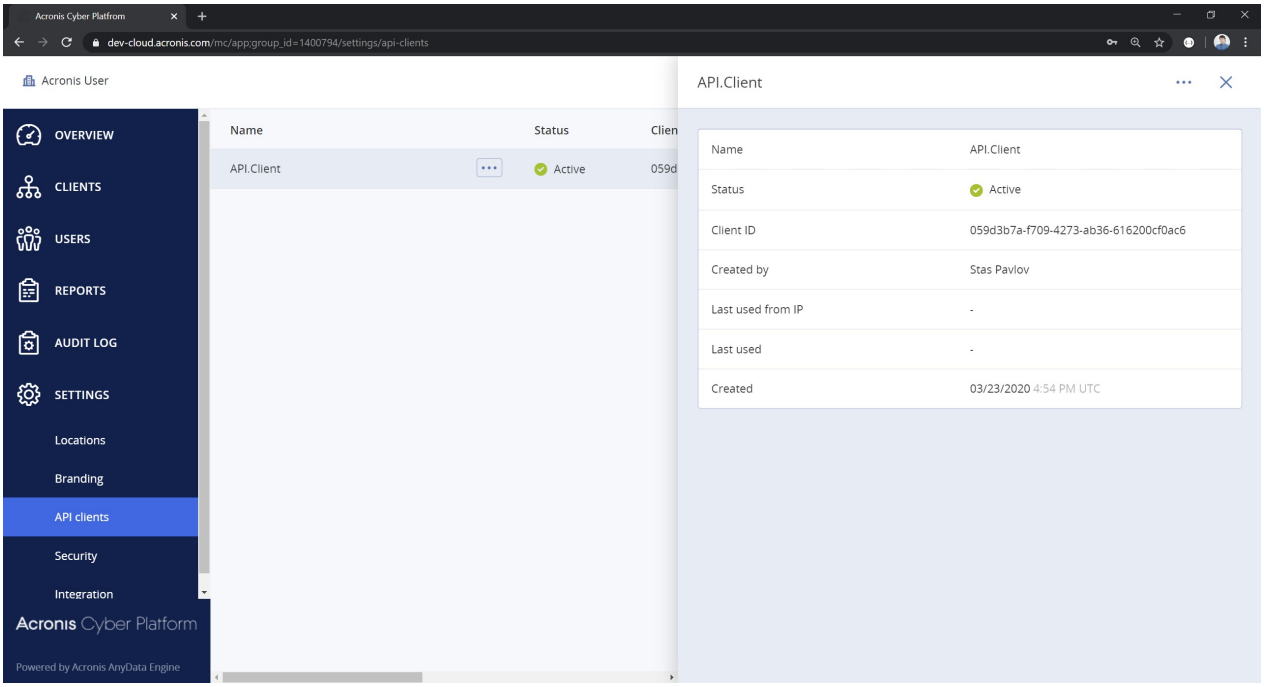
`client_name` is used as a display name for an API Client in the Management Console.

6. You should receive a JSON response body as on the screenshot below.



7. Add `client_id` and `client_secret` from the JSON response body to the same named variables as it was done for `my_tenants_is`.

8. Open the Management Console and check that you successfully create an API Client.



Exercise 2: Issue a token to access the API

Implementation details

A `client_id` and a `client_secret` can be used to access the API using the Basic Authorization but it's not a secure way as we discussed above. It's more secure to have a JWT token with limited life-time and implement a renew/refresh logic for that token.

To issue a token `/idp/token` end-point is called using `POST` request with param `grant_type` equal `client_credentials` and content type `application/x-www-form-urlencoded` with Basic Authorization using a `client_id` as a user name and a `client_secret` as a password.

> **You need to securely store the received token. For demo purposes we store a token as an environment variable. Please implement secure storage for your tokens.**

> **A token has time-to-live and must be renewed/refreshed before expiration time. The best practice is to check before starting any API calls sequence and renew/refresh if needed.**

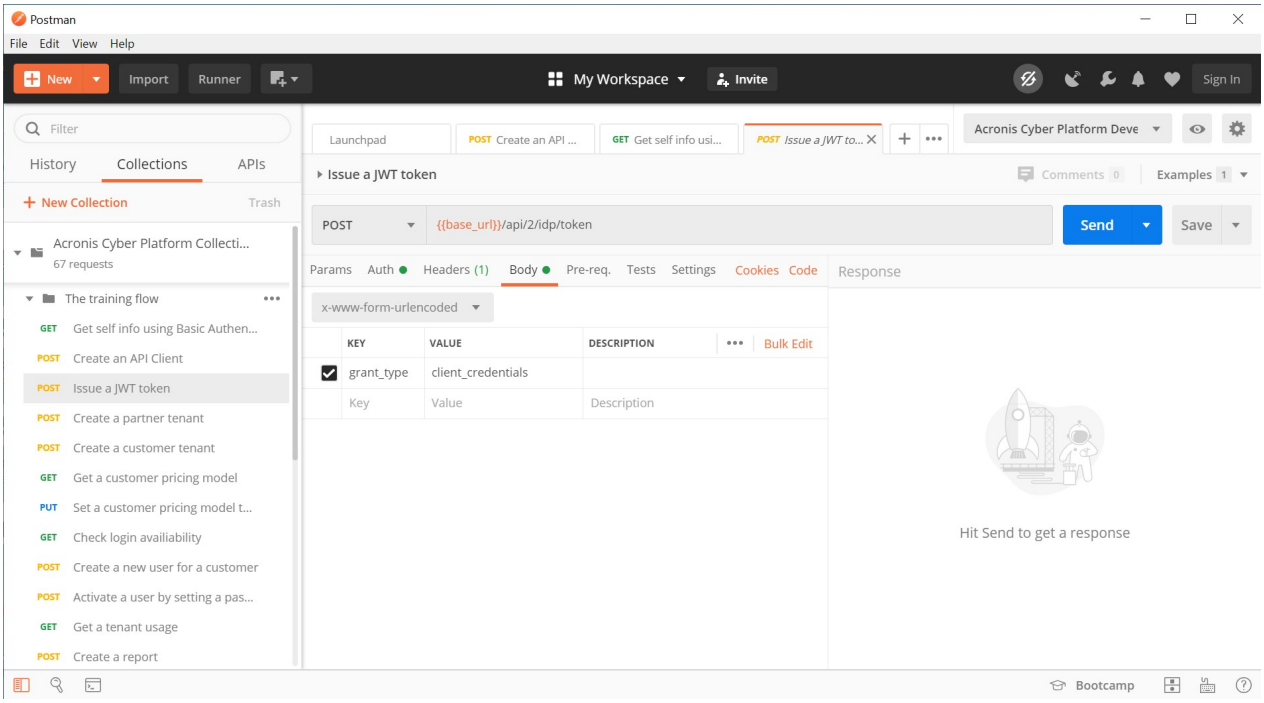
> **Currently, the default time-to-live to a token for the API is 2 hours.**

`expires_on` is a time when the token will expire in Unix time format -- seconds from January 1, 1970. Here we assume that you will renew/refresh a token as soon as you receive a `401` answer in Postman.

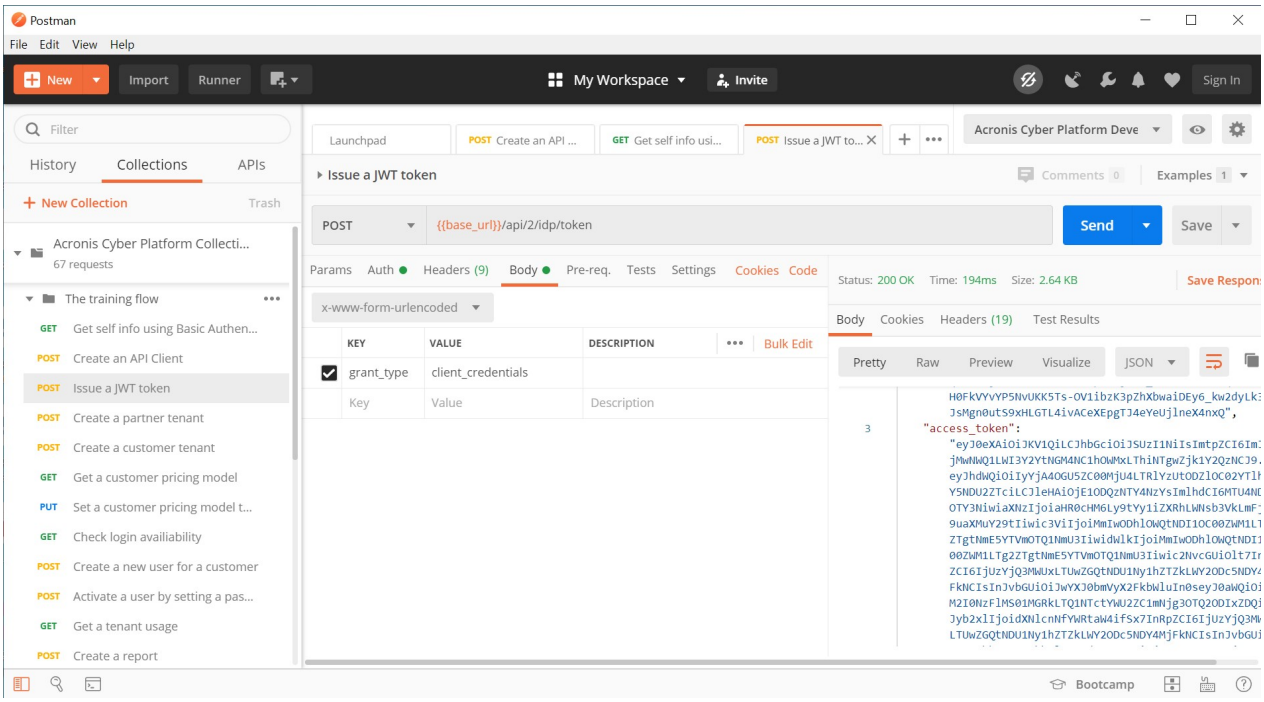
Step-by-step execution and checks

1. Be sure that you've fully finished *Create an API Client to access the API* step-by-step guide.

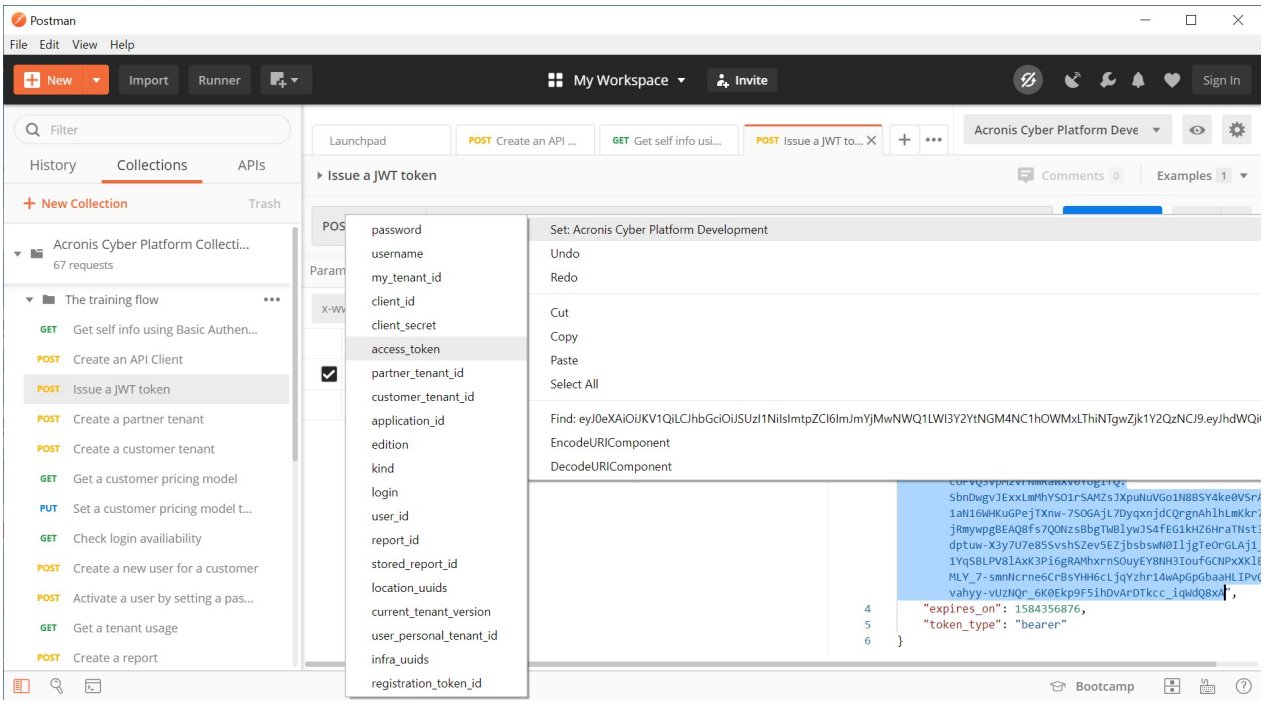
2. Open `Issue a JWT token` request in the `Postman` and then `Body` tab. Check that we use `x-www-form-urlencoded` as a `Content-Type` and pass `grant_type` equal to `client_credentials`.



3. Click `Send` button. You should receive a JSON response body as on the screenshot below.



4. Add `access_token` value without quotas from the JSON response body to the same named variables.



5. We are ready to make other API calls.

> **As soon as you receive `401 Unauthorized` error during any other API calls you must re-issue the token and copy the new token to `access_token` variable.**

Exercise 3: Create partner, customer and user tenants and set offering items

Implementation details

So now we can securely access the Acronis Cyber Platform API calls. In this topic we discuss how to create a partner, a customer tenants and enable for them all available offering items, and then create a user for the customer and activate the user by setting a password.

Assuming that we create the API client for our root tenant, we start from retrieving the API Client tenant information using GET request to `/clients/{{client_id}}` end-point. Then, using received `tenant_id` information as a parameter and `kind` equal to `partner`, we build a JSON body for POST request to `/tenants` end-point to create the partner. Next, we are going to enable all applications and offering items for the tenants. Briefly, we take all available offering items for the parent tenant of the partner or the customer using GET request to `/tenants/{{tenant_id}}/offering_items/available_for_child` end-point with needed query parameters specifying `edition` and `kind` of the tenant. Then, we need to enable these offering items for the partner or the customer using PUT request to `/tenants/{{tenant_id}}/offering_items` end-point with all offering items JSON in the request body and appropriate `{{tenant_id}}`.

> **The following `kind` values are supported `root`, `partner`, `folder`, `customer`, `unit`.**

This is absolutely the same process as for a customer, the only difference is `kind` equal to `customer` in the request body JSON and `/offering_items/available_for_child` parameters.

By default, customers are created in a trial mode. To switch to production mode we need to update customer pricing. To perform this task, we start from requesting current pricing using a GET request to `/tenants/{{customer_tenant_id}}/pricing` end-point then change `mode` property to `production` in the received JSON, then, finally, update the pricing using PUT request to `/tenants/{{customer_tenant_id}}/pricing` end-point with a new pricing JSON.

> ⚠ **Please, be aware, that this switch is non-revertible.**

Finally, we create a user for the customer. At first, we check if a login is available using GET request to `/users/check_login` end-point with `username` parameter set to an expected login. Then, we create a JSON body for POST request to `/users` end-point to create a new user.

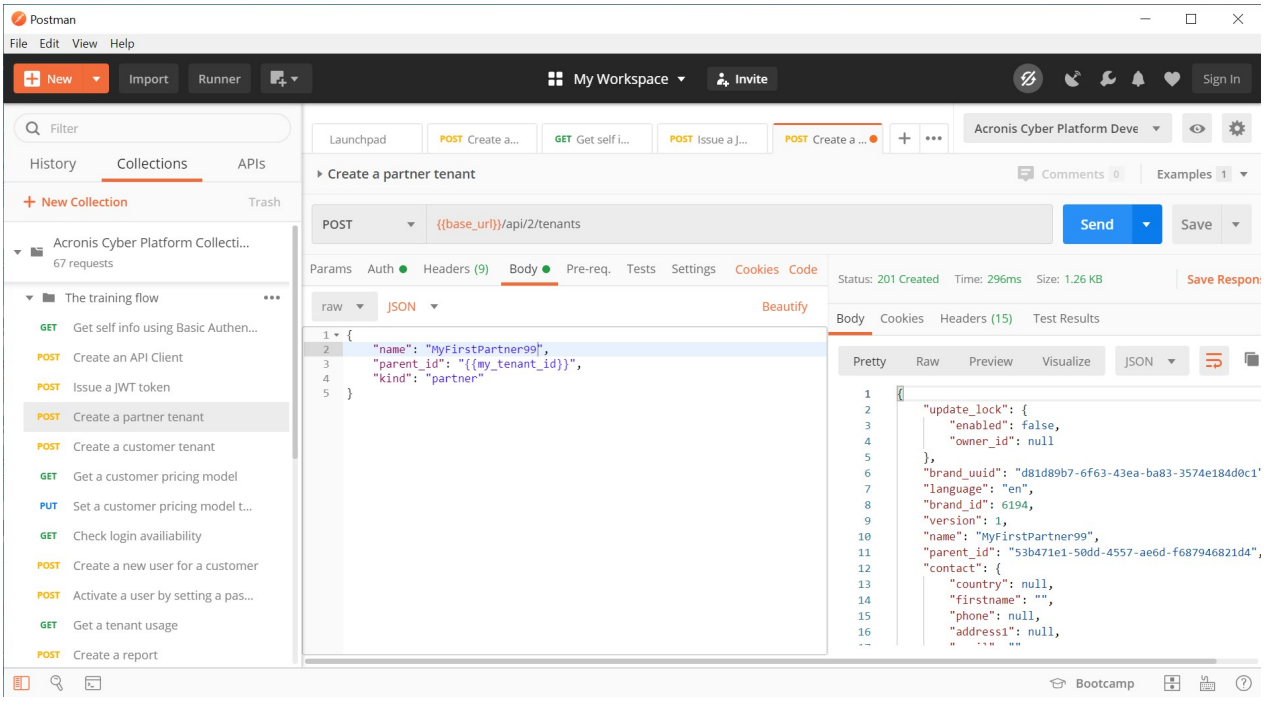
A created user is not active. To activate them we can either send them an activation e-mail or set them a password. The sending of an activation e-mail is the preferable way, as in this case a user can set their own password by themselves. We use a set password way for demo purposes and a fake e-mail is used. To set a password we send a simple JSON and POST request to `/users/{{user_id}}/password` end-point.

At this point, we've created a partner, a customer, enable offering items for them, create a user and activate them.

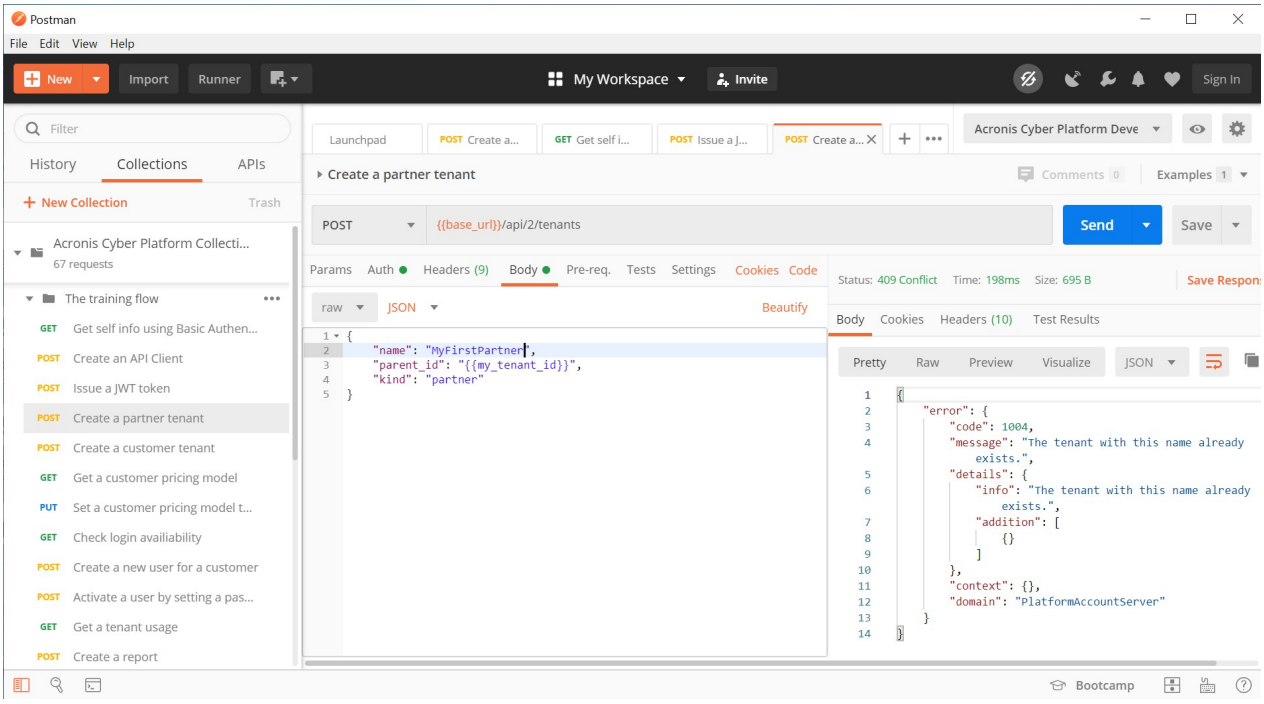
Step-by-step execution and checks

Create partner and enable all available standard edition offering items

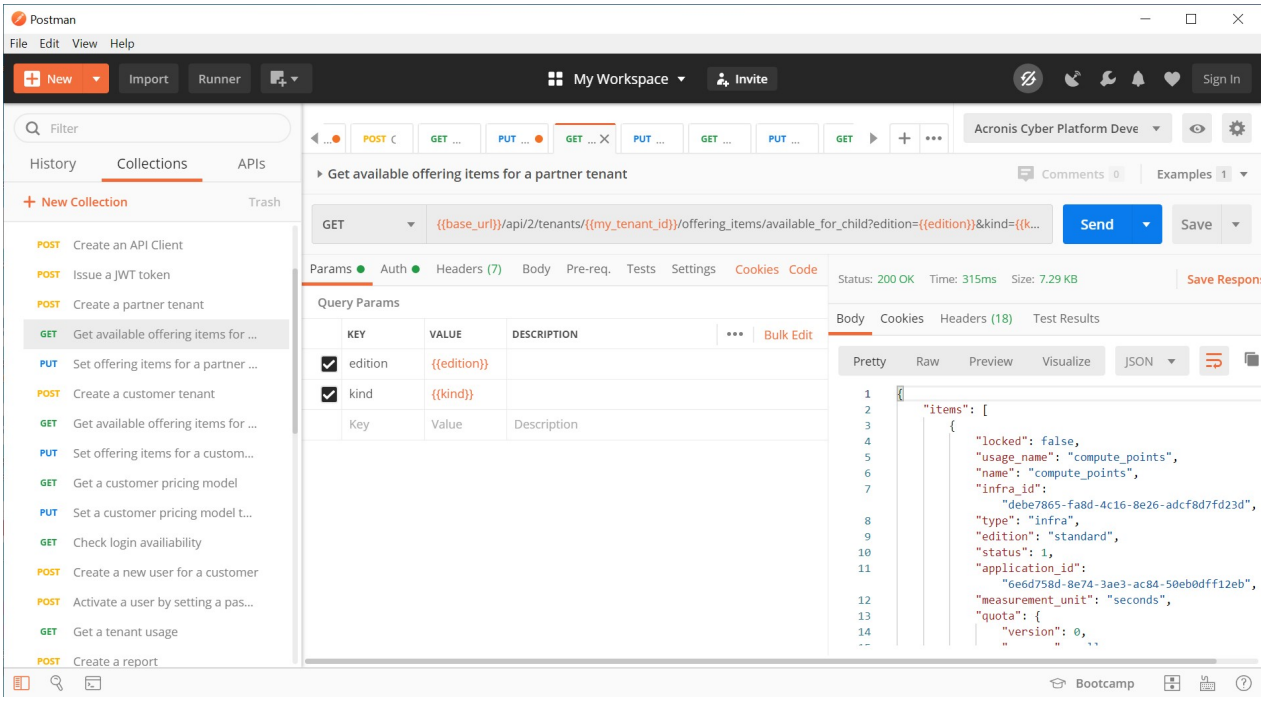
1. Be sure that you've fully finished *Issue a token to access the API* step-by-step guide.
2. Open `Create a partner tenant` request in the `Postman` and then `Body` tab. Check the JSON body. It uses `my_tenant_id` as a parent tenant id and expects to create a partner with name `MyFirstPartner`. Click `Send` button. You should receive a JSON response body as on the screenshot below.



3. If a partner with this name exists you will receive an error like on the screenshot below.

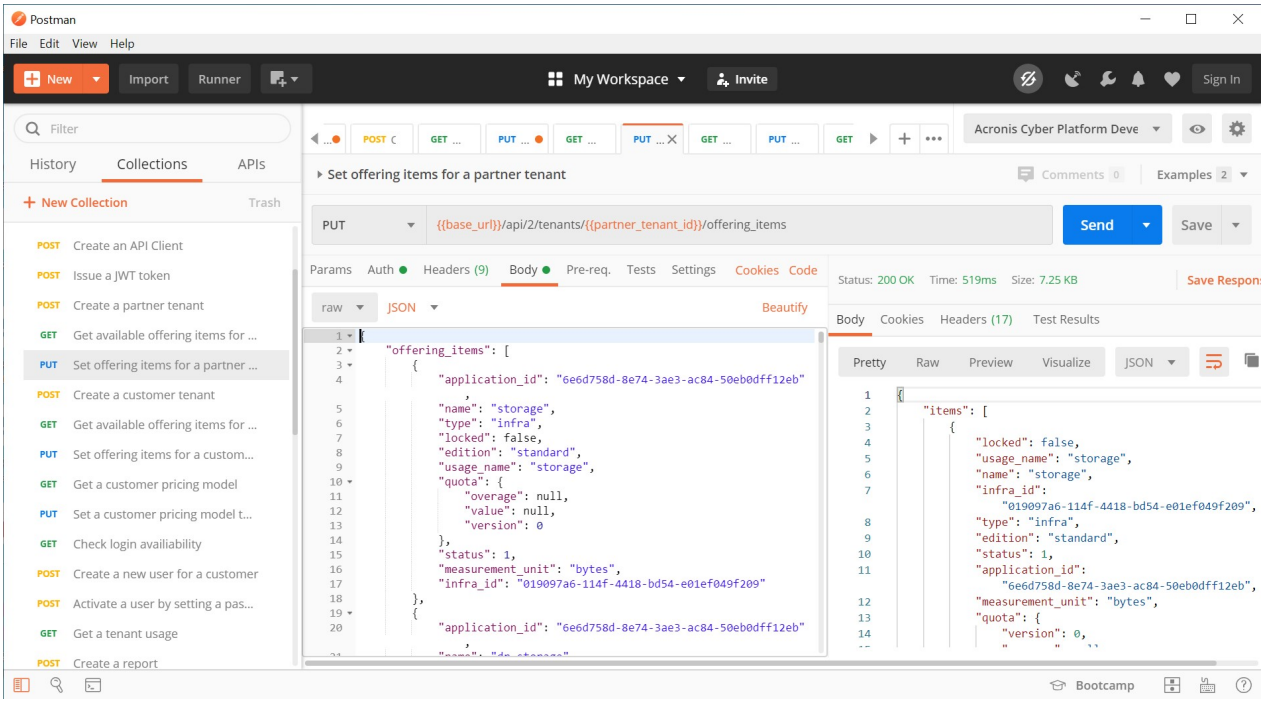


4. Add `id` value without quotas from the JSON response body to the `partner_tenant_id` variables the same way as you've done it before for other variables.
5. Open `Get available offering items for a partner tenant` request in the `Postman` and then `Params` tab. We are ready to request offering items available for the created tenant. To specify edition and kind the same named variables are used. Please, check the `Acronis Cyber Platform Development` environment `kind` variable set to `partner` and `edition` set to `standard` and click `Send` button. You should receive a JSON response body as on the screenshot below.

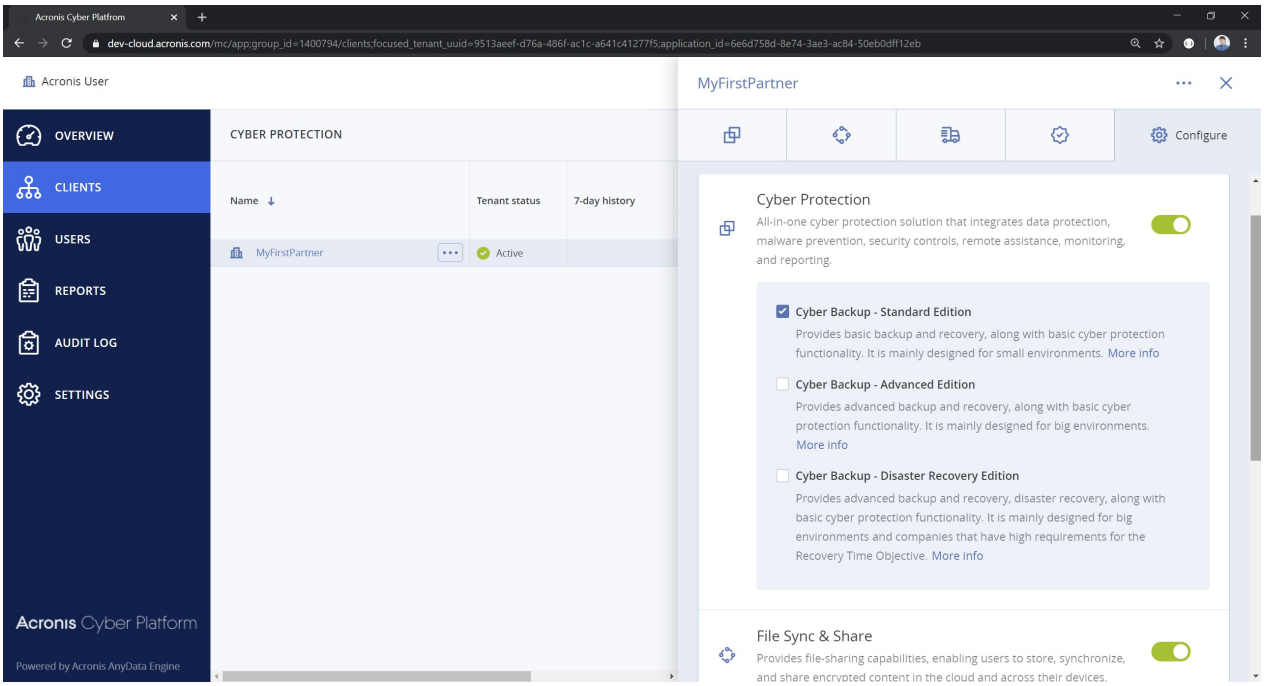


6. For simplicity purposes we enable all available offering items for a partner for standard edition. So, select the response JSON body, press right mouse button (on Windows) and from drop-down menu select `Copy`. We've copied the JSON to a buffer for further usage.
7. Open `Set offering items for a partner tenant` request in the `Postman` and then `Body` tab. You can see an example of offering items JSON body. Select all the JSON request body, , press right mouse button (on Windows) and from drop-down menu select `Paste`. Then replace the JSON root

element `items` with `offering_items`. So, now we a ready to update the partner offering items, click `Send` button. You should receive a JSON response body as on the screenshot below. It represents set offering items for the tenant.

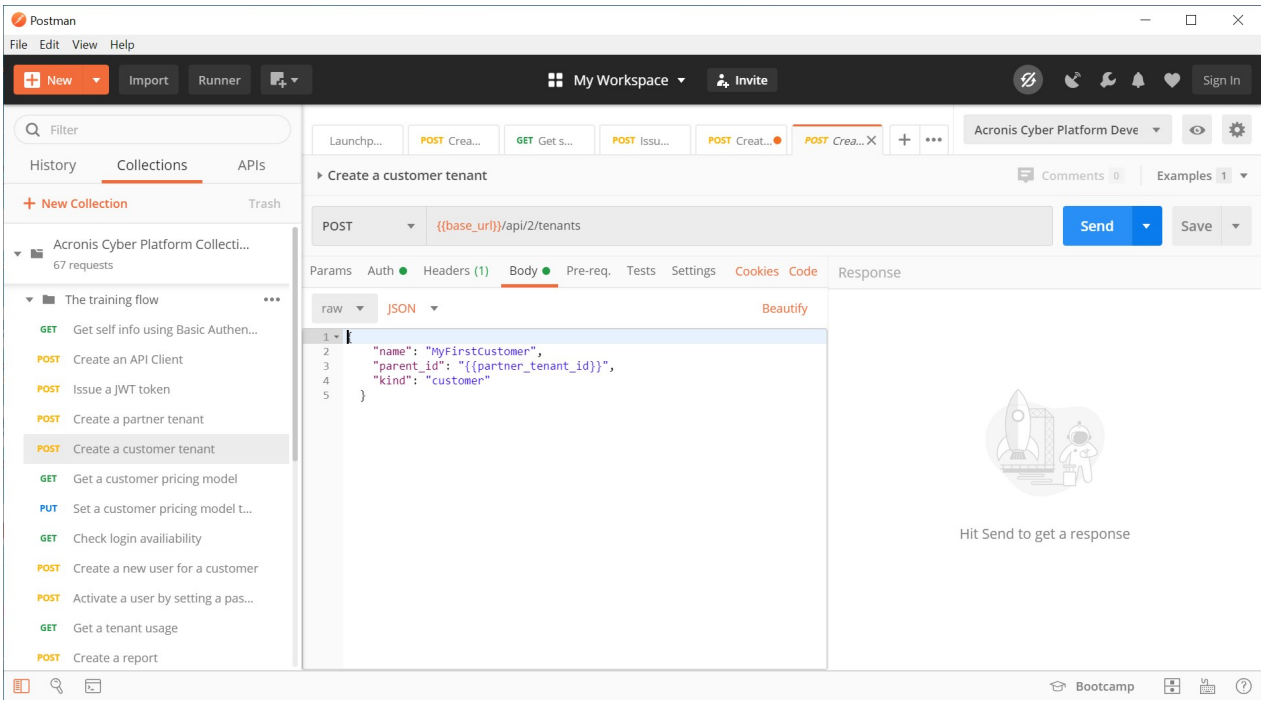


8. Open the Management Console and check that you successfully create a partner and enable all offering items for Standard Edition.

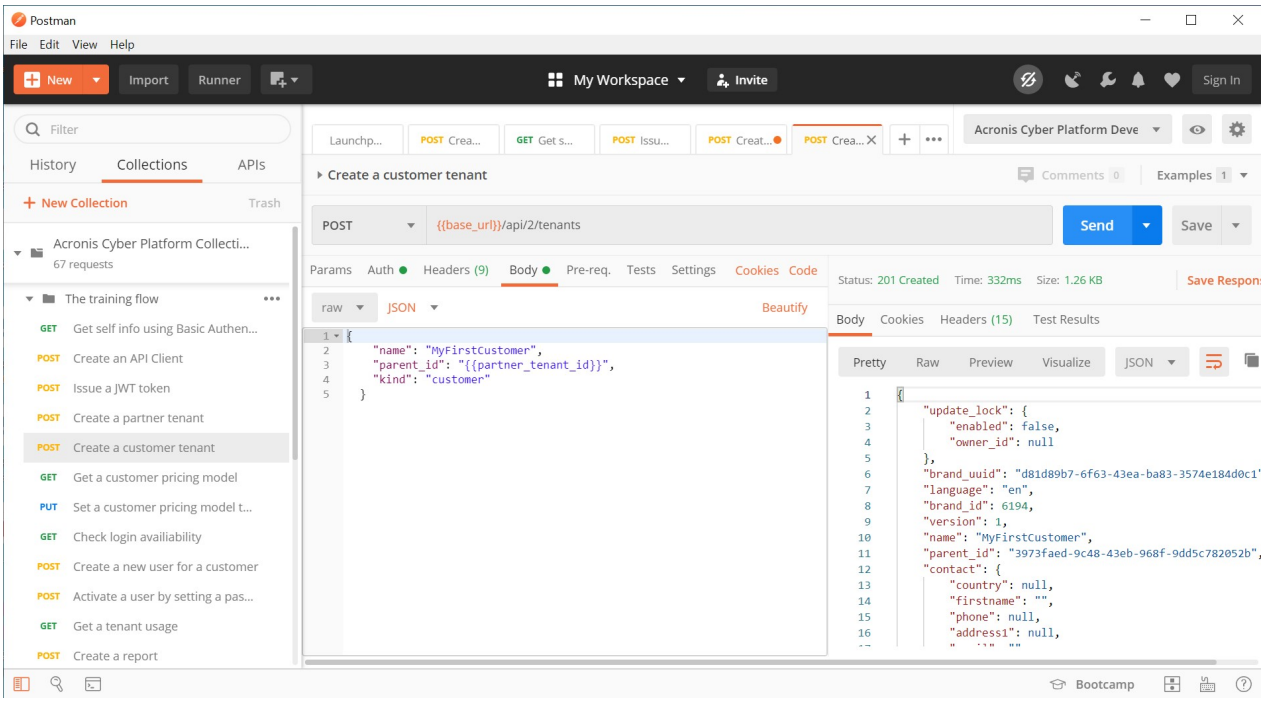


Create customer, enable all available standard edition offering items and switch to production mode

1. Open `Create a customer tenant` request in the `Postman` and then `Body` tab. Check the JSON body. It uses `partner_tenant_id` as a parent tenant id and expects to create a customer with name `MyFirstCustomer`.

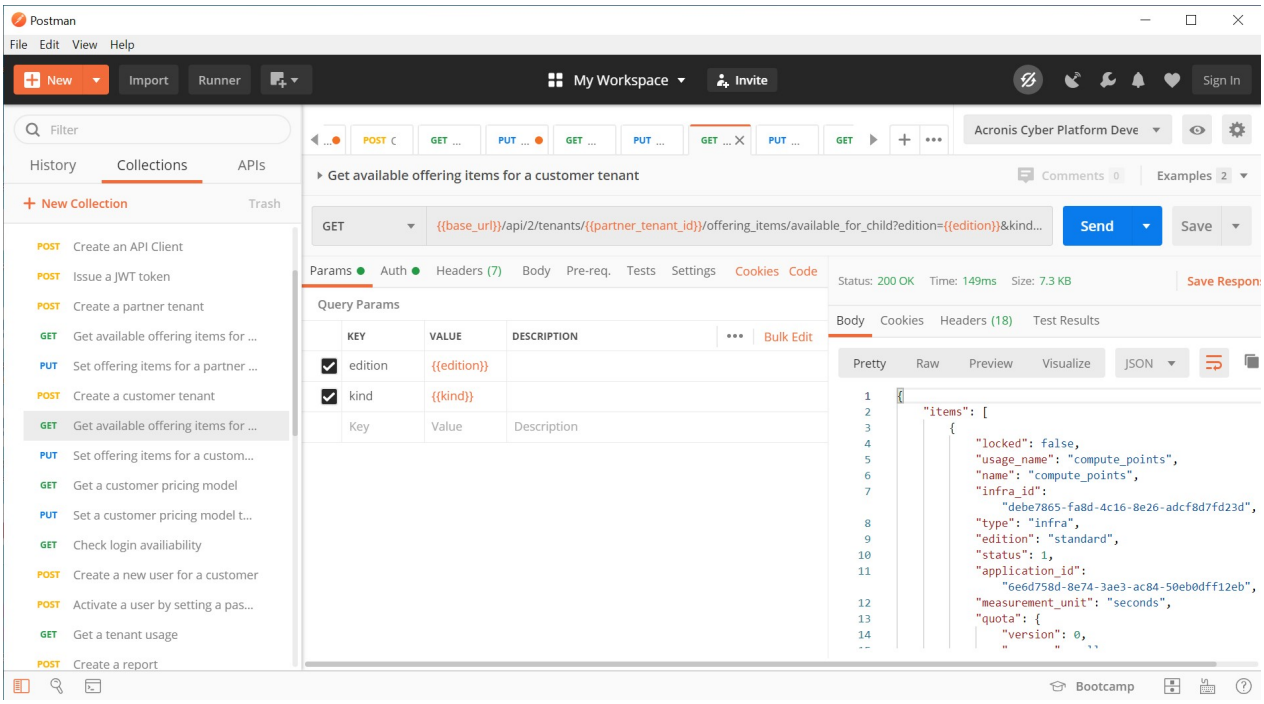


2. Click Send button. You should receive a JSON response body as on the screenshot below.

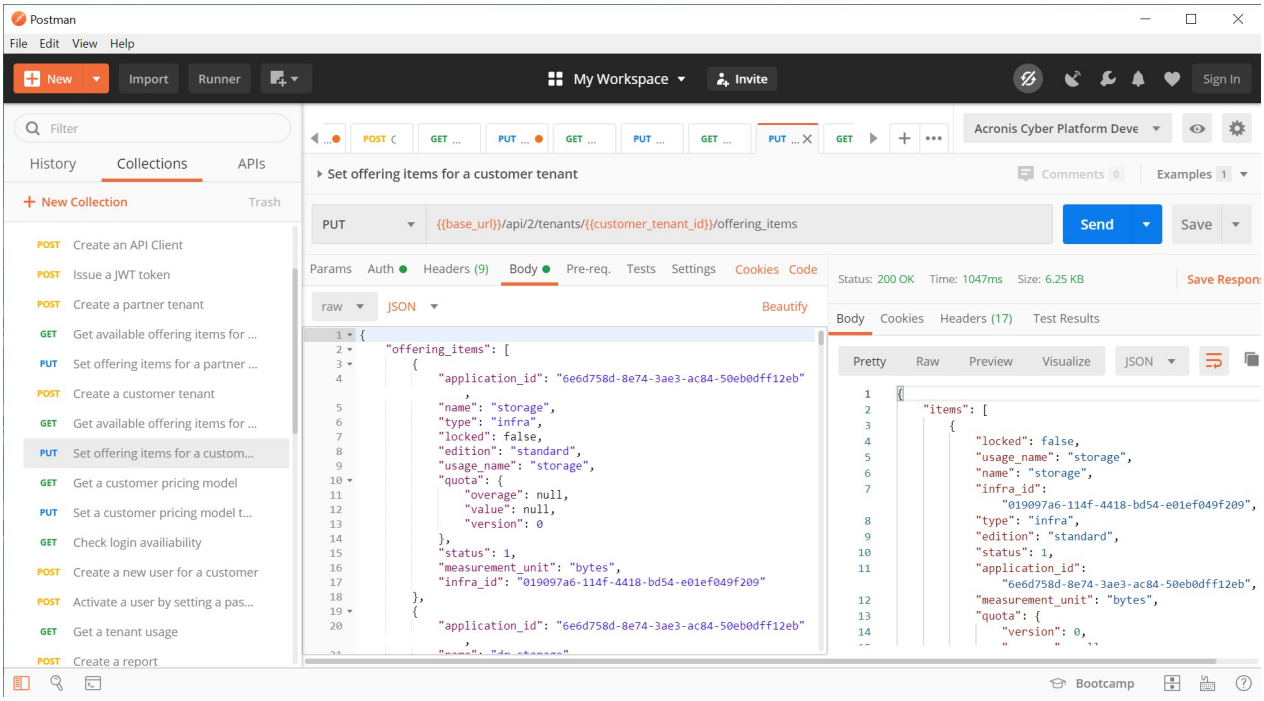


3.

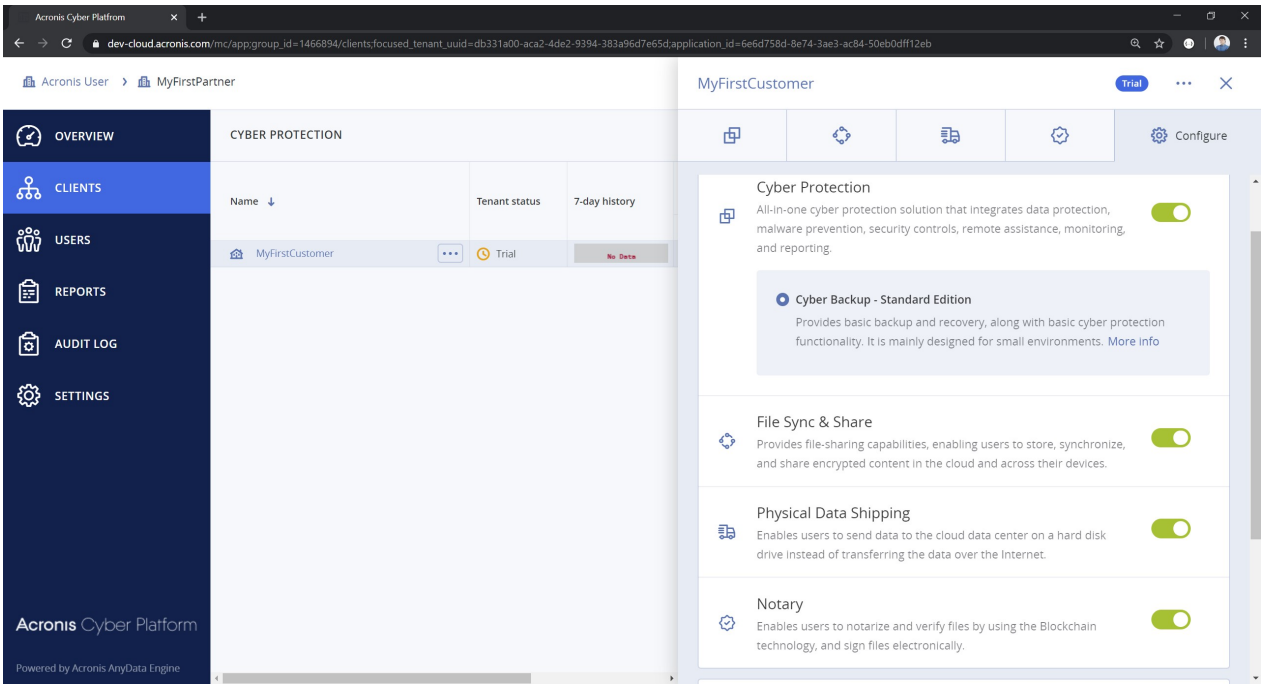
4. Open `Get available offering items for a customer tenant` request in the `Postman` and then `Params` tab. We are ready to request offering itemAdd `id` value without quotas from the JSON response body to the `customer_tenant_id` variables the same way as you've done it before for other variables.s available for the created tenant. To specify edition and kind the same named variables are used. Please, check the `Acronis Cyber Platform Development` environment `kind` variable set to `customer` and `edition` set to `standard` and click `Send` button. You should receive a JSON response body as on the screenshot below.



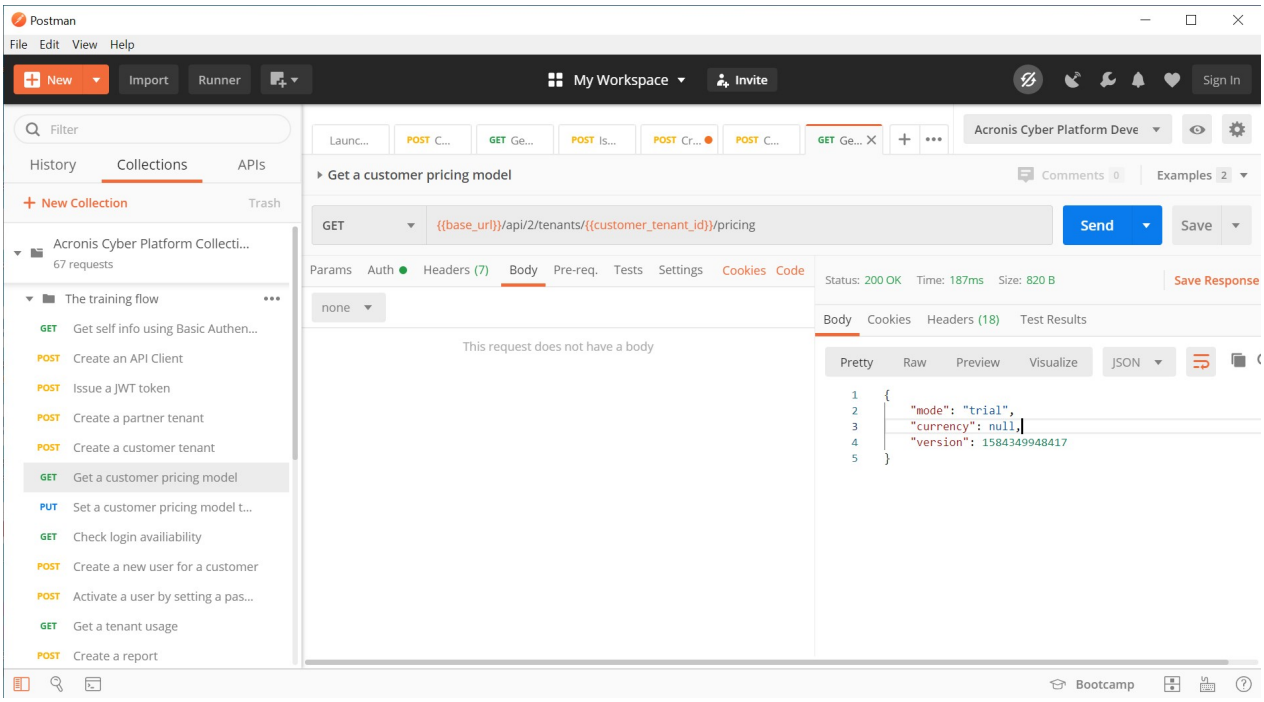
5. For simplicity purposes we enable all available offering items for a partner for standard edition. So, select the response JSON body, press right mouse button (on Windows) and from drop-down menu select `Copy`. We've copied the JSON to a buffer for further usage.
6. Open `Set offering items for a customer tenant` request in the `Postman` and then `Body` tab. You can see an example of offering items JSON body. Select all the JSON request body, , press right mouse button (on Windows) and from drop-down menu select `Paste`. Then replace the JSON root element `items` with `offering_items`. So, now we a ready to update the partner offering items, click `Send` button. You should receive a JSON response body as on the screenshot below. It represents set offering items for the tenant.



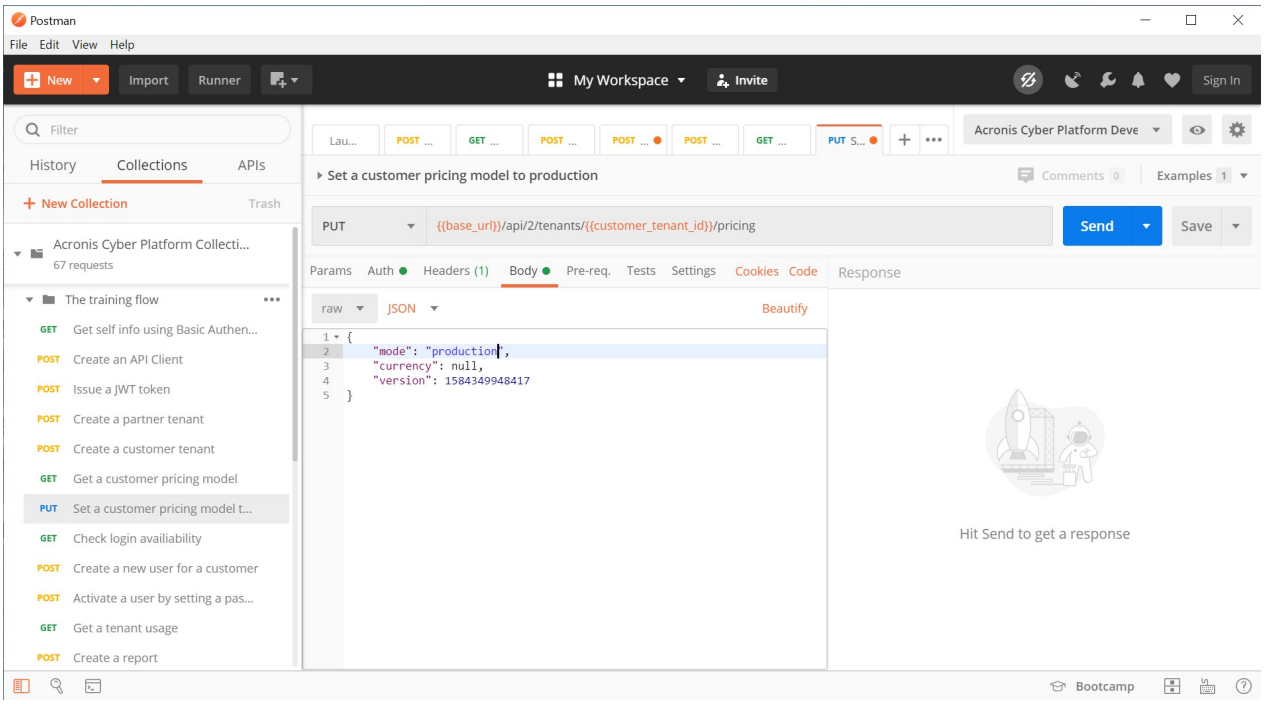
7. Open the Management Console and check that you successfully create a customer and enable all offering items for Standard Edition.



8. By default a customer is created in `Trial` mode. To switch it to `Production` we need to change pricing for the tenant.
9. Open `Get a customer pricing model` request in the `Postman` and click `Send`. You should receive a JSON response body as on the screenshot below. It represents the current pricing mode for the customer.

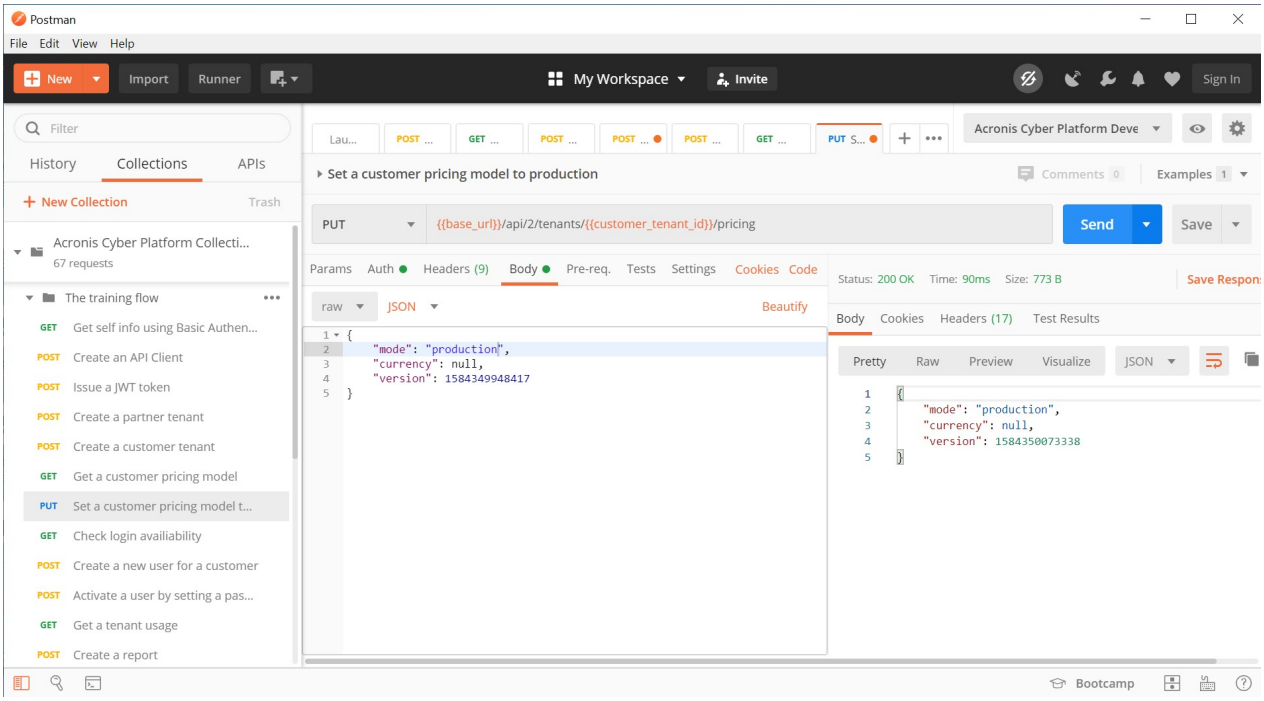


10. Select the response JSON body, press right mouse button (on Windows) and from drop-down menu select `Copy`. We've copied the JSON to a buffer for further usage.
11. Open `Set a customer pricing model to production` request in the `Postman` and then `Body` tab. You can see an example of offering items JSON body. Select all the JSON request body, , press right mouse button (on Windows) and from drop-down menu select `Paste`. Then replace the JSON `mode` element value `trial` with `production`. So, now we a ready to update the partner offering items.

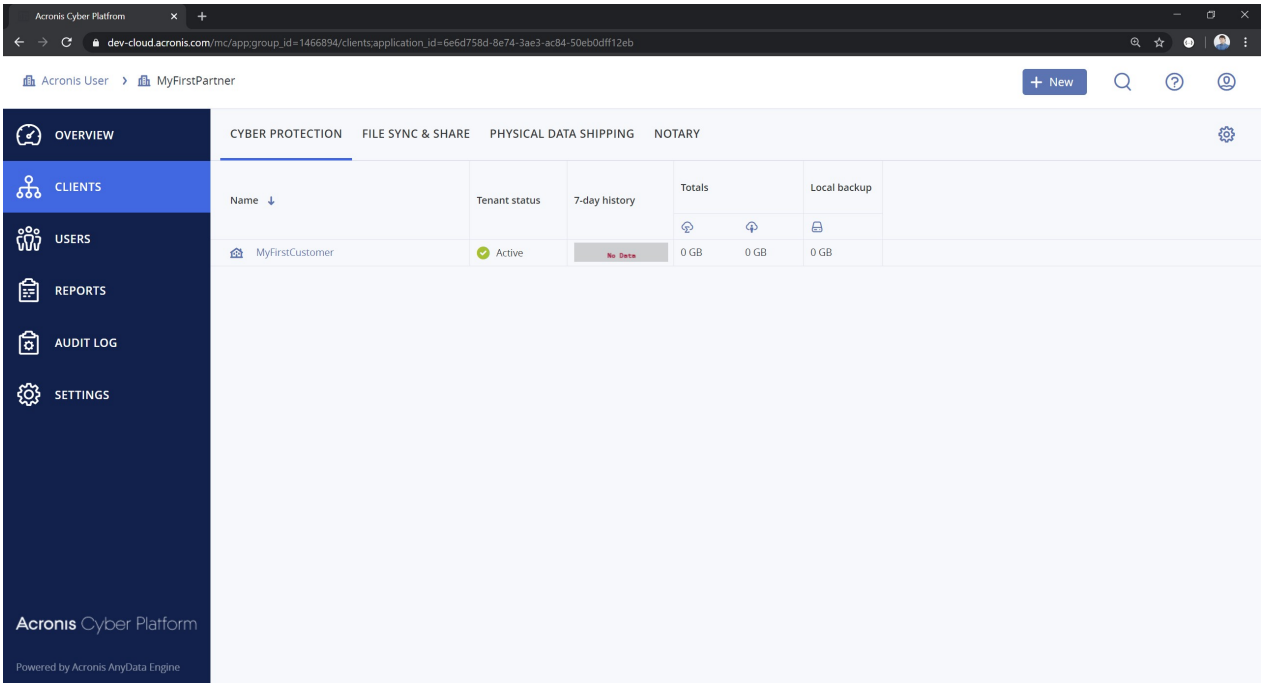


Please, be aware, that this switch is non-revertible.

12. Click Send button. You should receive a JSON response body as on the screenshot below. It represents the current pricing mode and you can see that it's `production`.

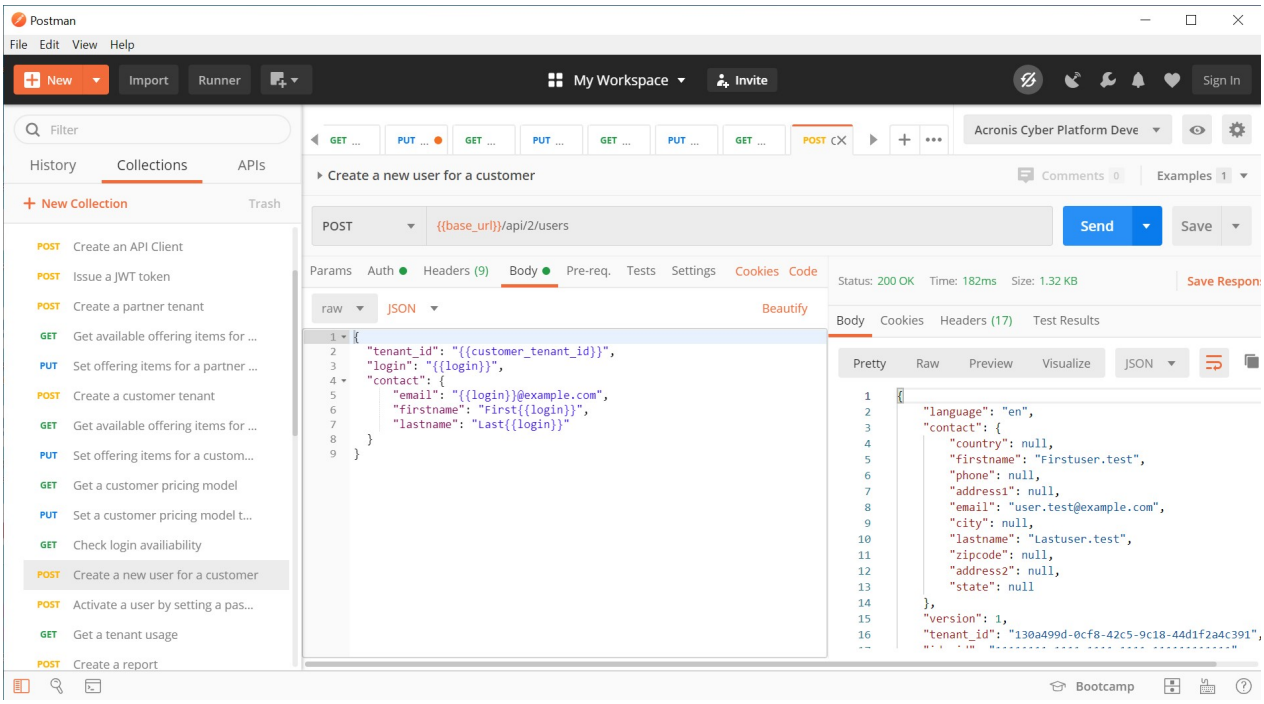


13. Open the Management Console and check that you successfully switch a customer to production.

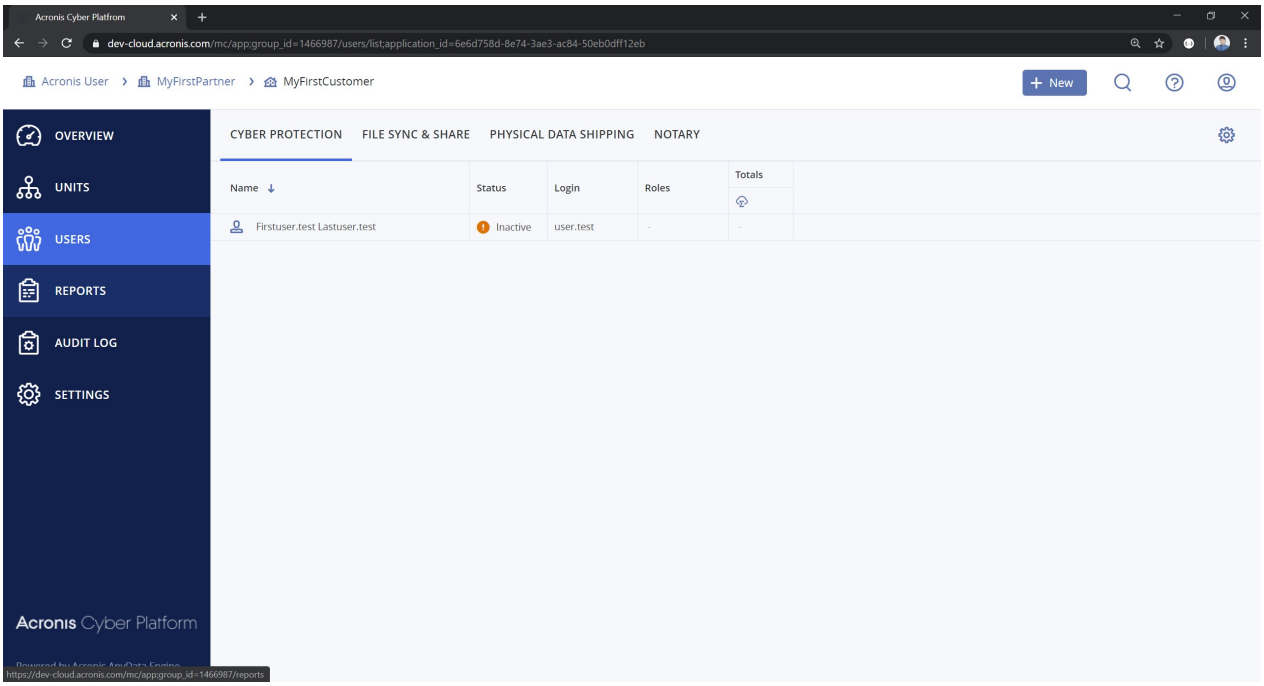


Create user, activate them by setting a password and enable backup services

1. We a ready to create a user for the created customer. Open `Check login availability` request in the `Postman` and then `Params` tab. It's required to have unique login for a user. So, before creating a user we need to check login availability.
2. Please, check the `Acronis Cyber Platform Development` environment `login` variable set to the login you want to use.
3. Click `Send` button. If the login is available you receive empty response body and `204` response code.
4. In unsuccessful case, please, change the `Acronis Cyber Platform Development` environment `login` variable to another value and repeat *step 3*.
5. Open `Create a new user for a customer` request in the `Postman` and then `Body` tab. You can see a parametrized JSON request body to create a user for a customer with tenant id equal to `customer_tenant_id` and login - `login`. As well we use fake e-mail and fist and last names. Click `Send` button. You should receive a JSON response body as on the screenshot below. It represents set the created user.



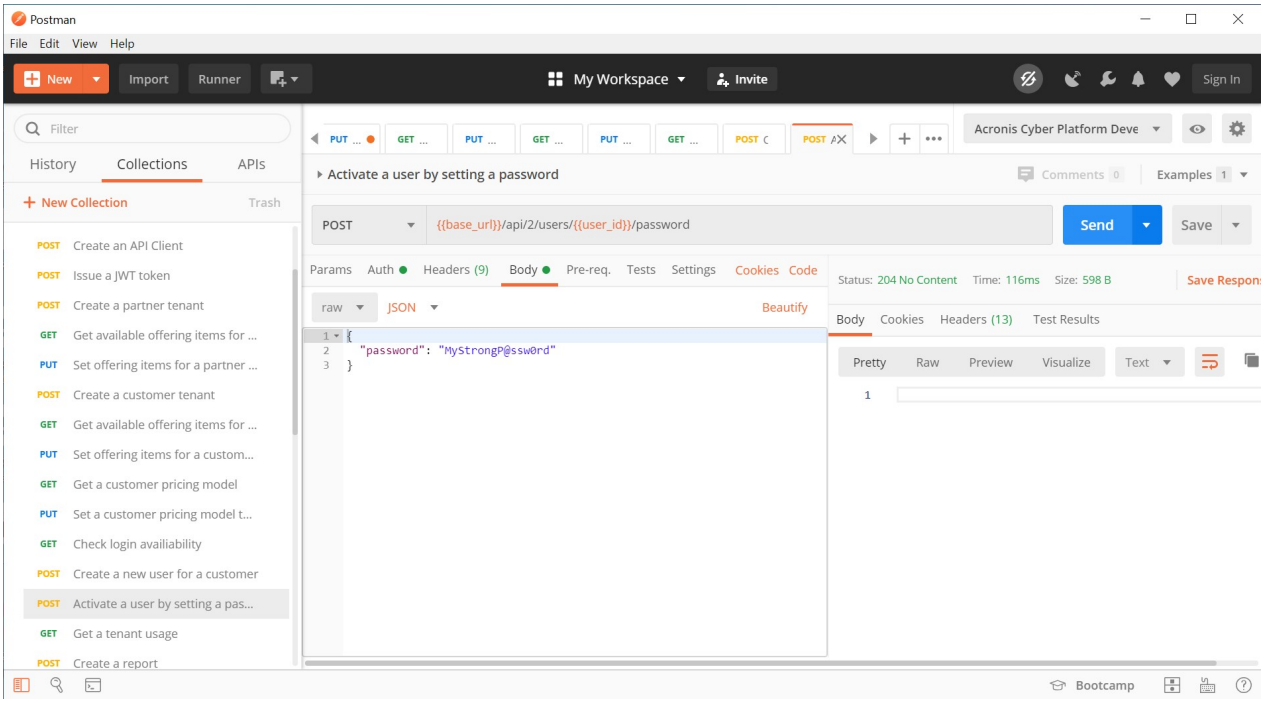
6. Add `id` value without quotas from the JSON response body to the `user_id` variables the same way as you've done it before for other variables.
7. Add `personal_tenant_id` value without quotas from the JSON response body to the `user_personal_tenant_id` variables the same way as you've done it before for other variables.
8. Open the Management Console and check that you successfully create a user.



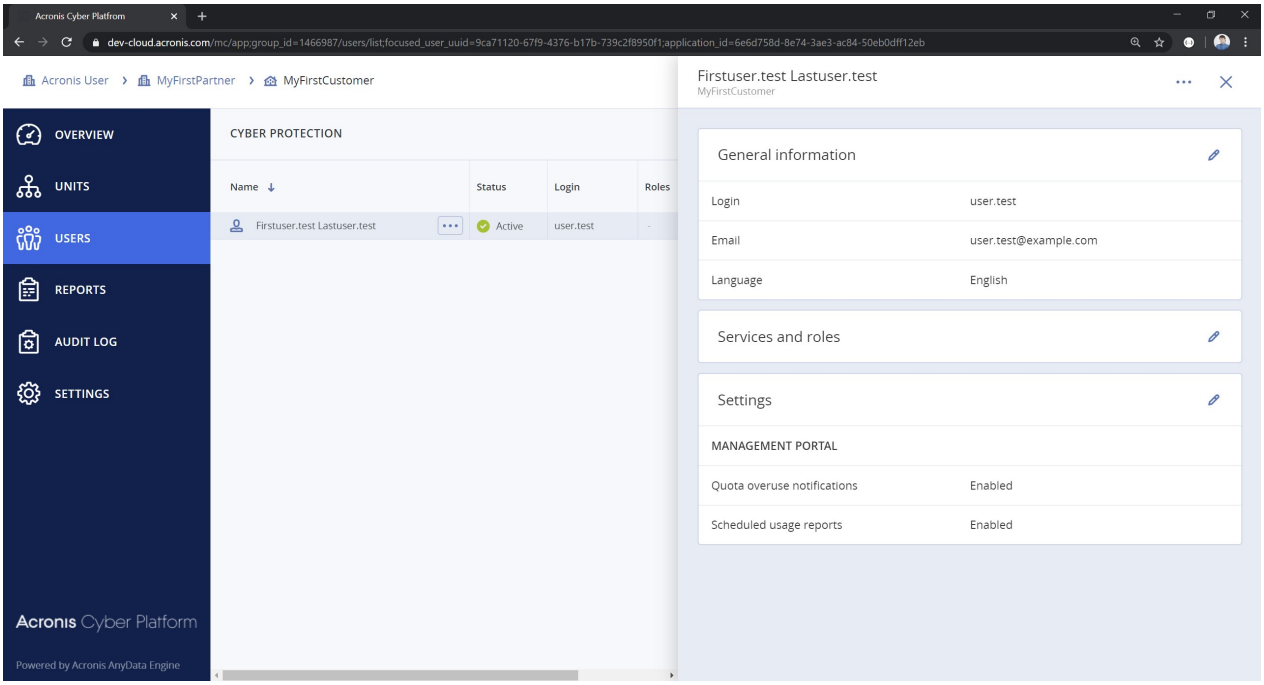
> **A created user is not active. To activate them we can either send them an activation e-mail or set them a password.**

> **The sending of an activation e-mail is the preferable way, as in this case a user can set their own password by themselves. We use a set password way for demo purposes and a fake e-mail is used.**

9. Open `Activate a user by setting a password` request in the `Postman` and then `Body` tab. You can see a simple JSON request body to set a user password. Click `Send`.



10. If the the call is successful you receive empty response body and `204` response code.
11. Open the Management Console and check that you successfully activate the user.



> **The created user has no roles assigned. It means it can't use any service. To enable services/applications you need to assign an appropriate role to a user. In next steps you will create a bash script to assign the created user `backup_user` role to enable backup services.**

> **All operations with the user account roles are located under the `/users/{{user_id}}/access_policies` endpoint.**

> **To build a JSON to assign a role for a user `id` and user `personal_tenant_id` need to be known.**

> **You can find more information regarding JSON format in the API documentation https://developer.acronis.com/doc/platform/management/v2/#/http/models/structures/access-policy.**

12. Select `Activate a user by setting a password` in the Postman collection, click right mouse button and select `Duplicate`.
13. Open `Activate a user by setting a password Copy` and rename it to `Assign a user backup_role`.
14. Change request URL from `{{base_url}}/api/2/users/{{user_id}}/password` to `{{base_url}}/api/2/users/{{user_id}}/access_policies` and click `Save`.
15. Switch method type from `POST` to `PUT`.
16. Open `Body` tab and replace

```
{
  "password": "MyStrongP@ssw0rd"
}
```

with

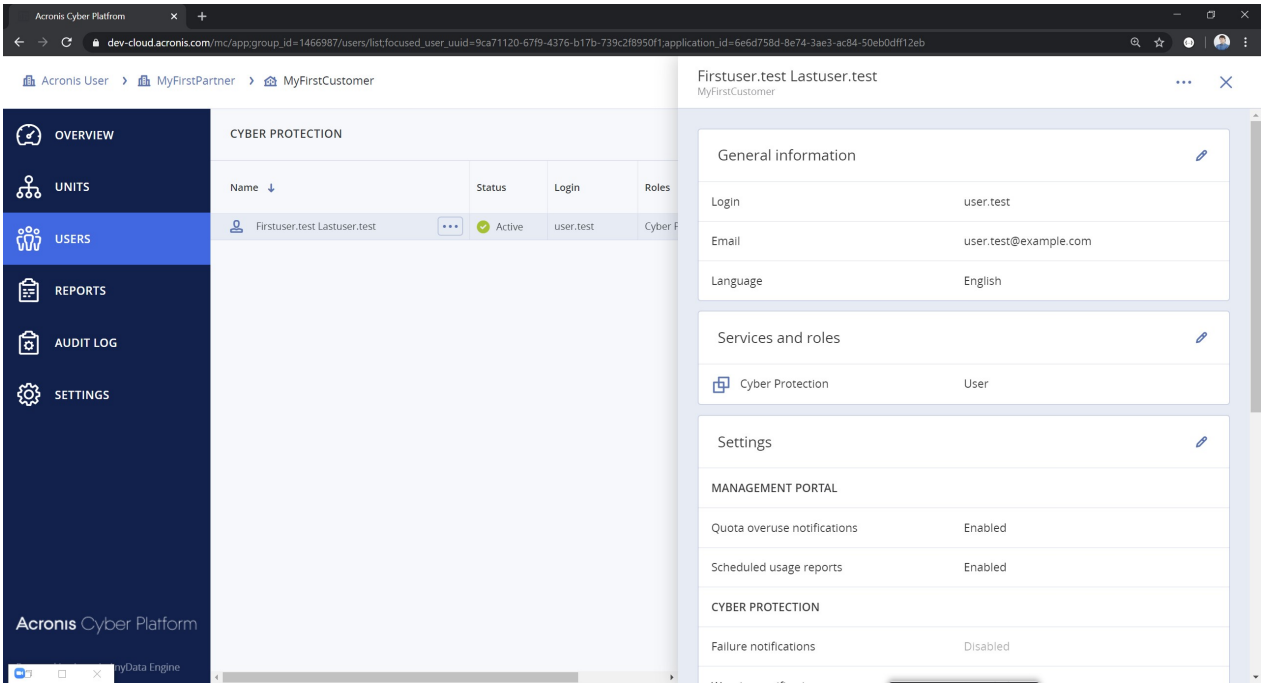
```
{
 "items": [
     {
      "id": "00000000-0000-0000-0000-000000000000",
      "issuer_id": "00000000-0000-0000-0000-000000000000",
      "role_id": "backup_user",
      "tenant_id": "{{user_personal_tenant_id}}",
      "trustee_id": "{{user_id}}",
      "trustee_type": "user",
      "version": 0
     }
     ]
}
```

17. Click `Send` button. You should receive a JSON response body as on the example below. It represents roles array for the user.

```
{
    "items": [
        {
            "tenant_id": "5702c92f-c486-4a7f-937c-7c05ca827532",
            "role_id": "backup_user",
            "version": 0,
            "trustee_id": "9ca71120-67f9-4376-b17b-739c2f8950f1",
            "trustee_type": "user",
```

```
            "issuer_id": "00000000-0000-0000-0000-000000000000",
            "id": "00000000-0000-0000-0000-000000000000"
        }
    ]
}
```

18. Open the Management Console and check that you successfully add the backup role to the user.



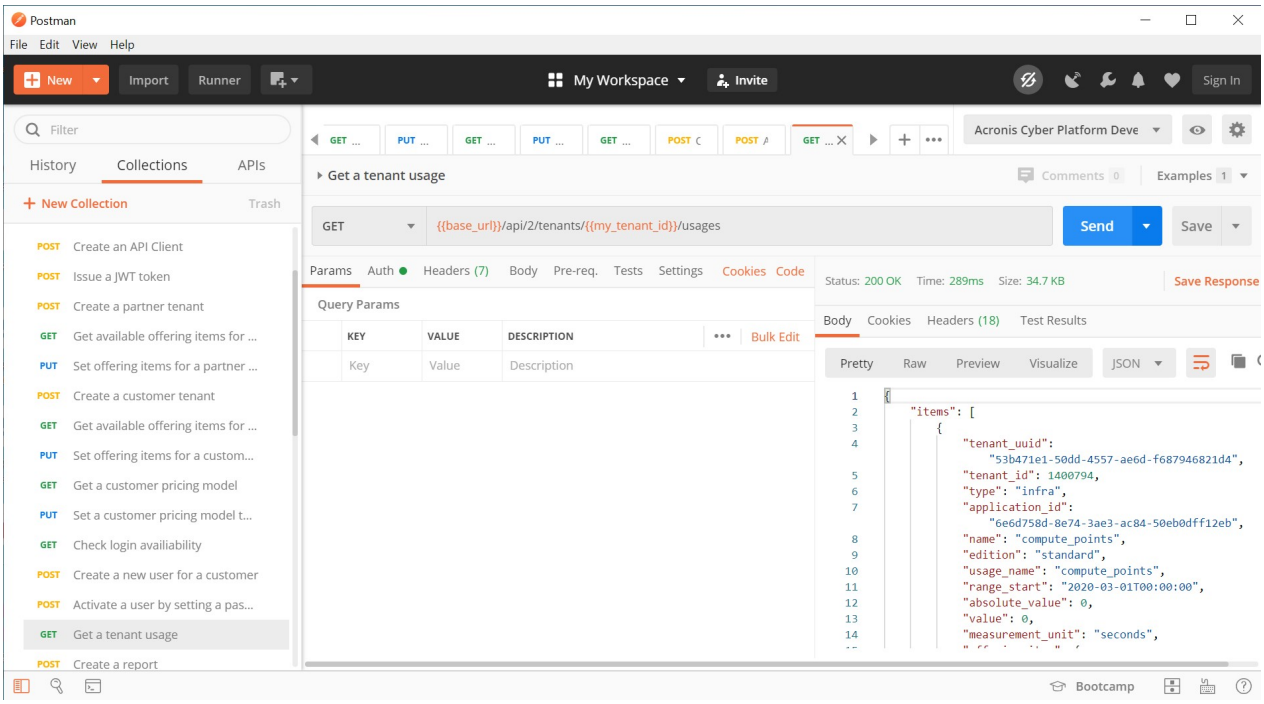
Exercise 4: Get a tenant usage

Implementation details

A very common task is to check a tenant's usage. It's a simple task. We just need to make a GET request to `/tenants/{{tenant_id}}/usages` end-point, as result we receive a list with current usage information in JSON format.

Step-by-step execution and checks

1. Open `Get a tenant usage` request in the `Postman` and then click `Send`. You should receive a JSON response body as on the screenshot below. It represents usages array for the `my_tenant_id` tenant.



The information about a service usage of the tenant, provided by the `/tenants/{{tenant_id}}/usages` endpoint, is updated on average every 5-6 hours.

It might be very useful to store usage information for further processing.

For billing purposes it's expected that reporting capabilities are used which are described in details in the next exercise.

Exercise 5: Create and download simple report

Implementation details

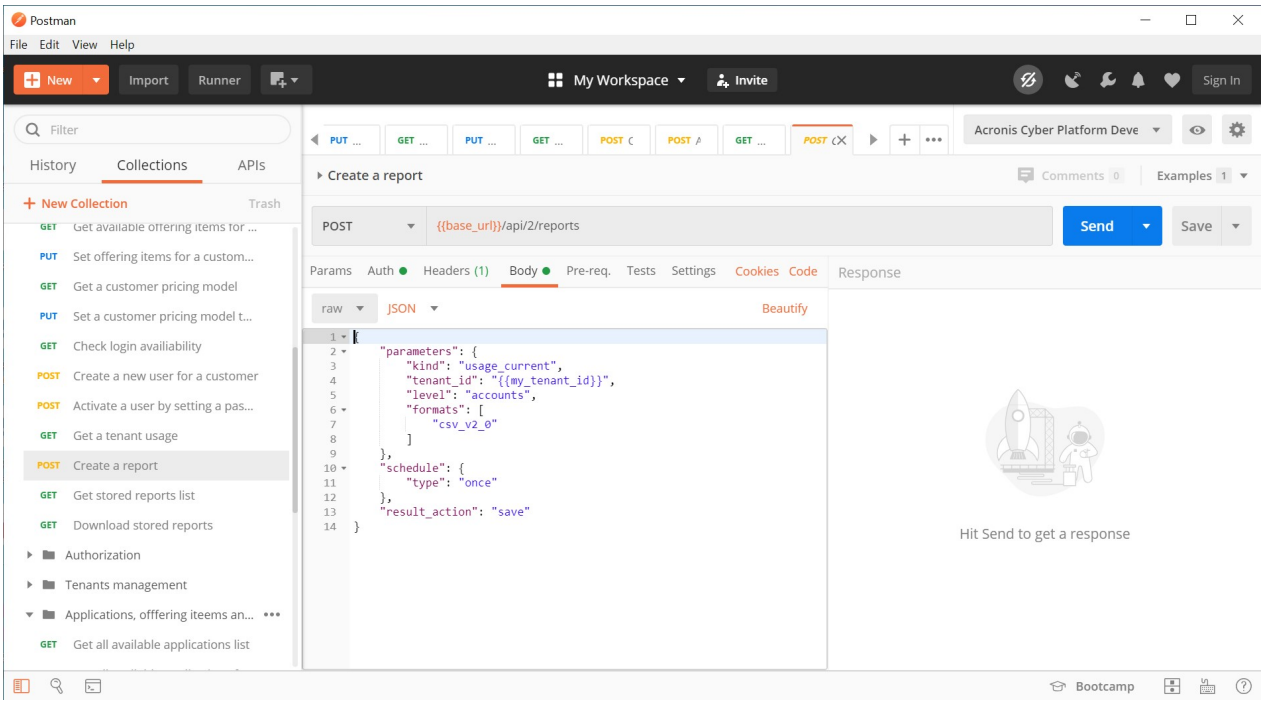
The reporting capability of the Acronis Cyber Cloud gives you advanced capabilities to understand usage. In the following simple example, we create a one-time report in csv format, and then download it. To check other options, please, navigate to the Acronis Cyber Platform documentation.

To create a report to save, we build a body JSON and make a POST request to `/reports` end-point. Then we look into stored reports with specified `{{report_id}}` making a GET request to `/reports/{{report_id}}/stored` endpoint.

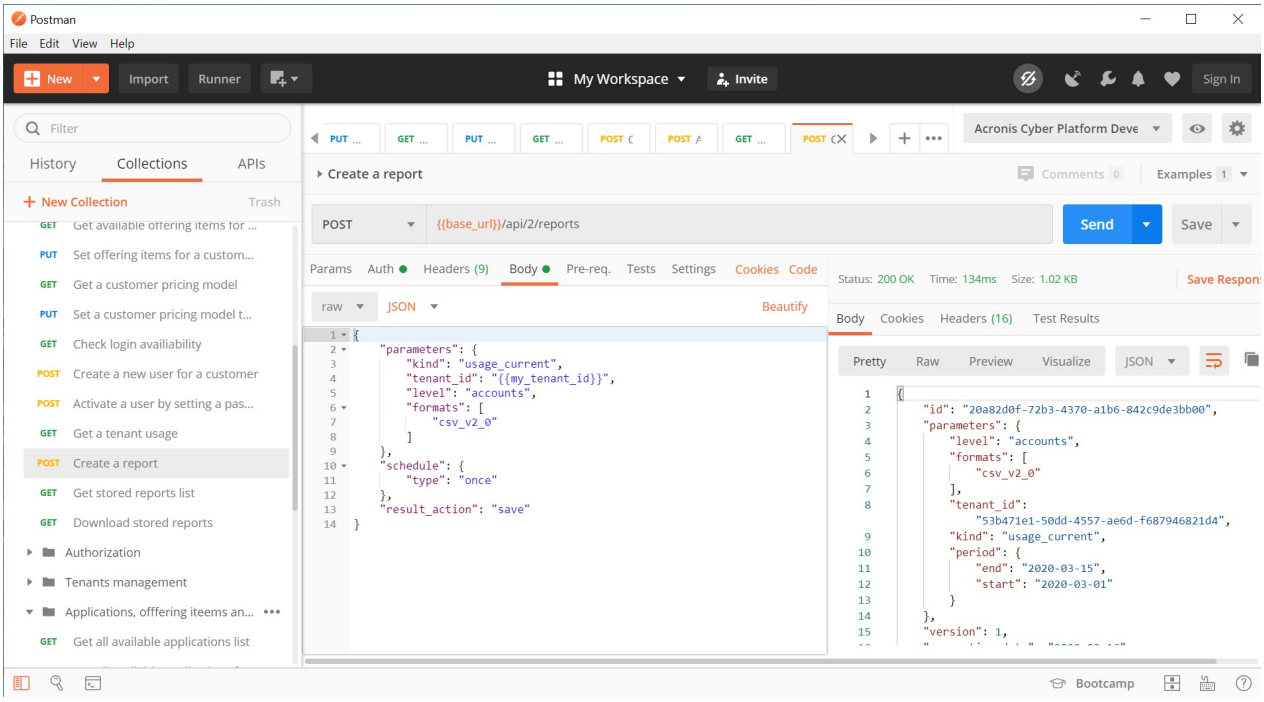
And finally, we download the report created using a GET request to `/reports/{{report_id}}/stored/{{stored_report_id}}`.

Step-by-step execution and checks

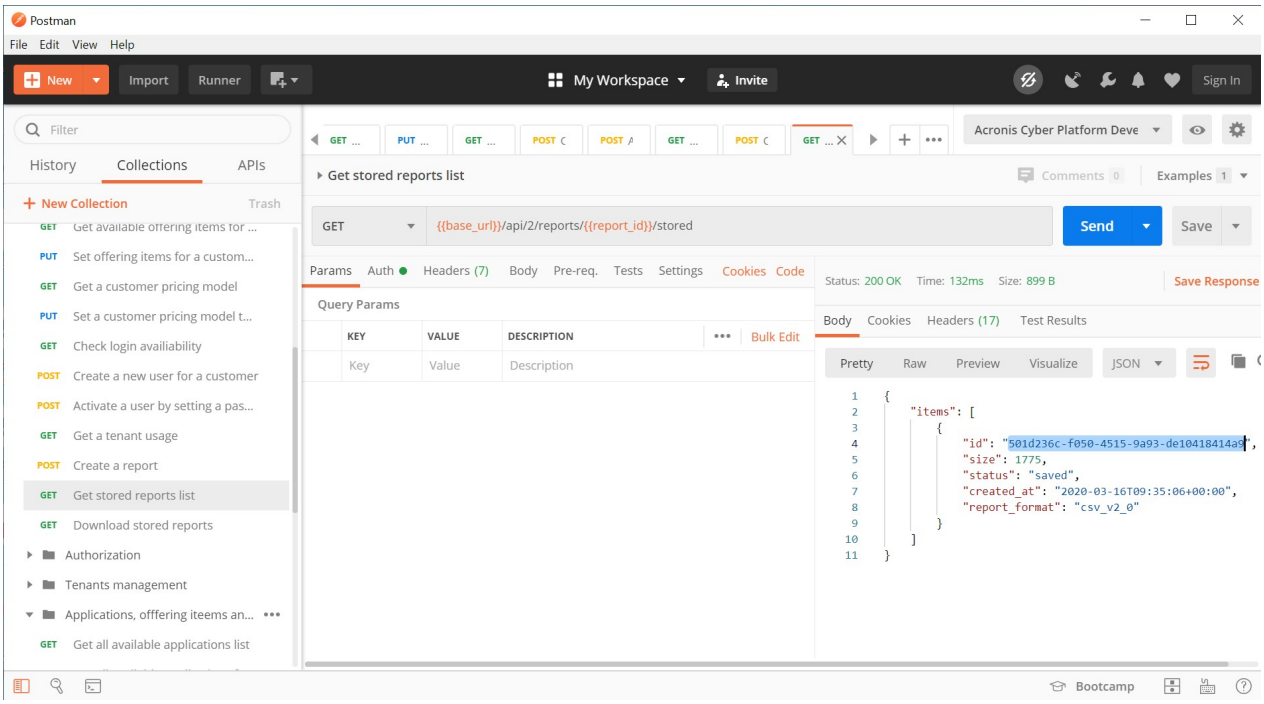
1. Open `Create a report` request in the `Postman` and then `Body` tab. You can see a parametrized JSON request body to create a `usage_current` report, executesonce time with `accounts` level of details for the tenant `my_tenant_id`.



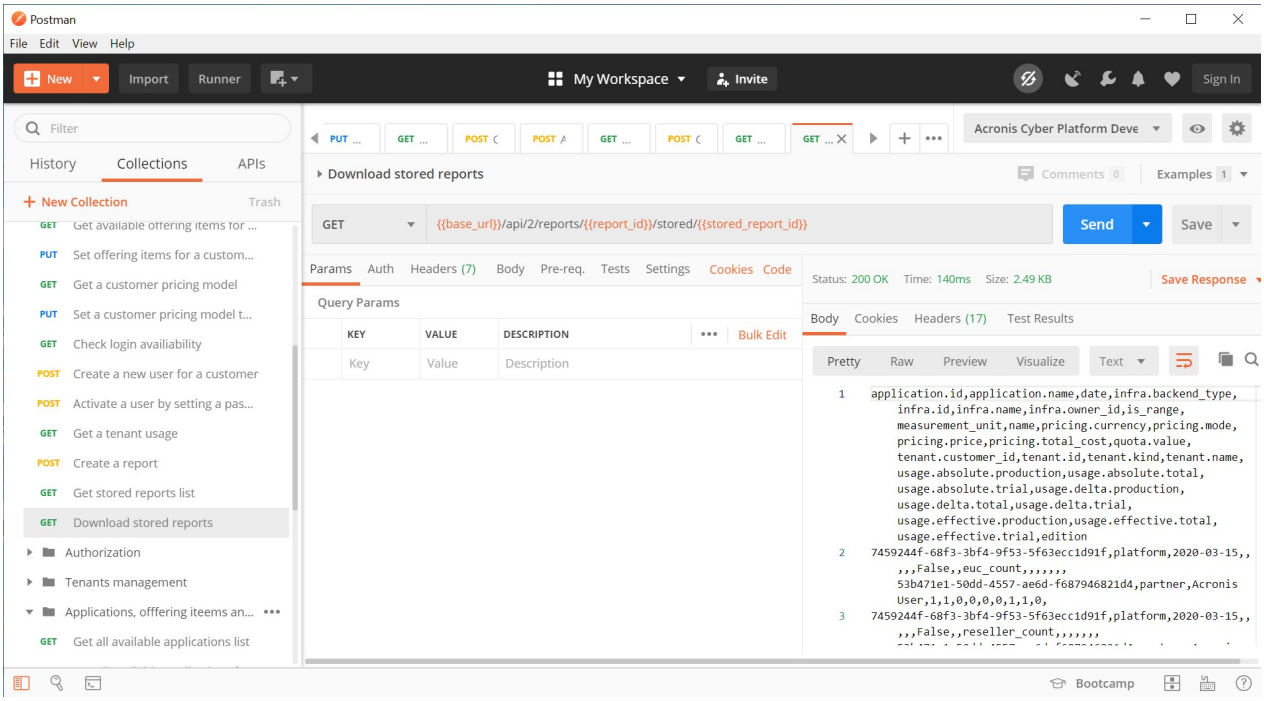
2. Click `Send`. You should receive a JSON response body as on the screenshot below. It represents a newly created report for the `my_tenant_id` tenant.



3. Add `id` value without quotas from the JSON response body to the `report_id` variables the same way as you've done it before for other variables.
4. Open `Get stored reports list` request in the `Postman` and click `Send`. You should receive a JSON response body as on the screenshot below. It represents a stored report information for the created report.



5. Add `id` value without quotas from the JSON response body to the `stored_report_id` variables the same way as you've done it before for other variables.
6. Open `Download stored reports` request in the `Postman` and click `Send`. You should receive a CSV response body as on the screenshot below. It the stored report the created report.



Exercise 6: Add marks to your API calls for better support

Implementation details

It's technically possibly to identify your API calls as they are connected to your API Client. But still it's required a lot of efforts and hard to find in your Audit log at the Management Portal for your. Thus to better support your development effort it would be a great idea to identify your integrations and API calls somehow. Traditional way to do it in a RESTFul word is using the `User-Agent` header.

There are common recommendations how to build your `User-Agent` header:

```
User-Agent: <product>/<product-version> <comment>
```

For example, for our hands-on lab, you can use:

```
User-Agent: Training/1.0 Acronis #CyberFit Developers Business Automation Training
```

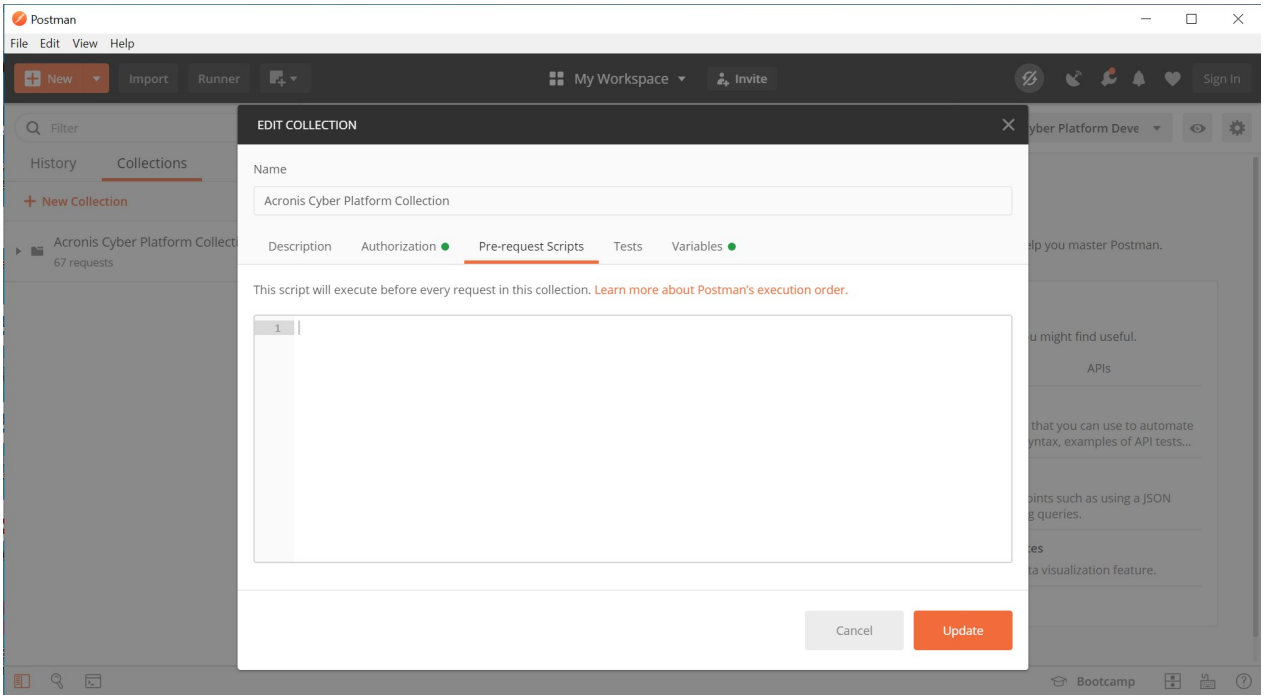
To implement it using Postman, we need just add the header to each Postman call using API. It can be implemented adding the following code to the *Pre-request Scripts* of the imported postman collection:

```
pm.request.headers.add({key: 'User-Agent', value: 'Training/1.0 Acronis #CyberFit Dev
```

> ⚠ **Please, for a real integration, use your real integration name, a specific version and suitable comments to simplify your support.**

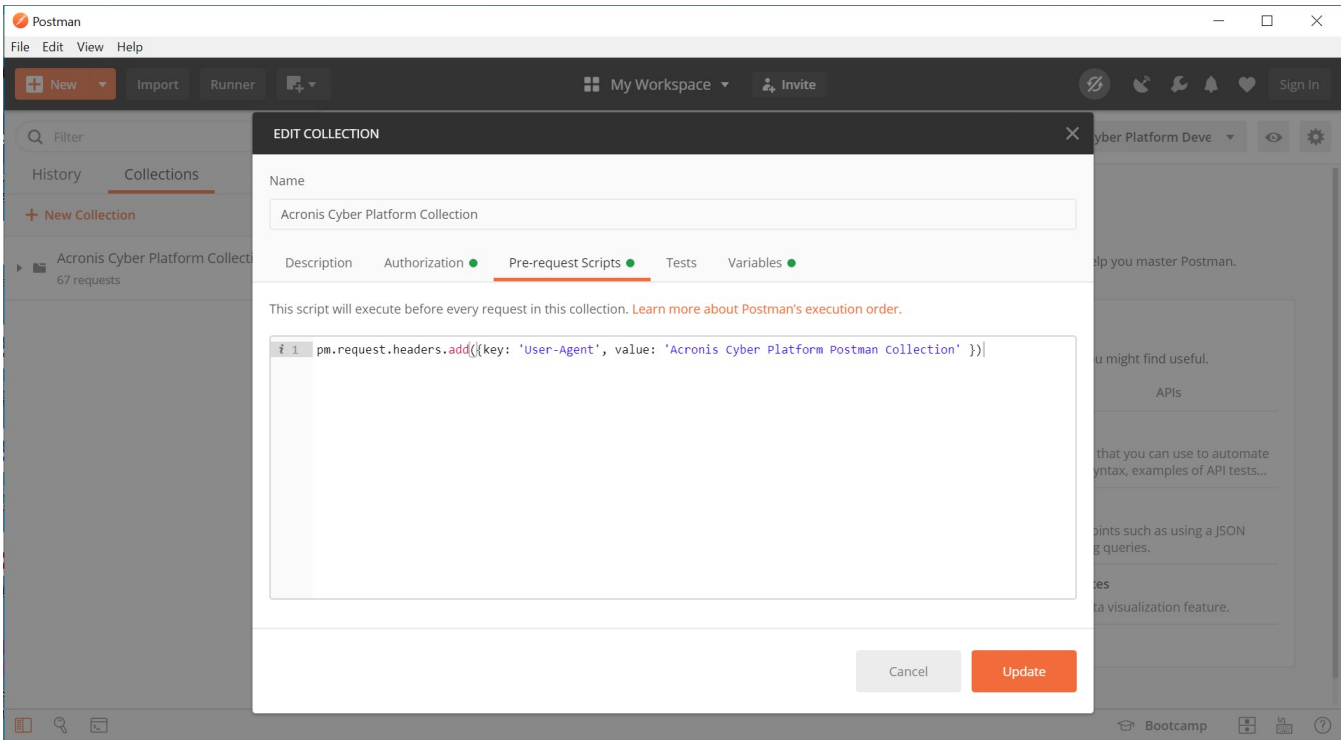
Step-by-step execution and checks

1. Open settings for the imported collection and open *Pre-request Scripts* tab.



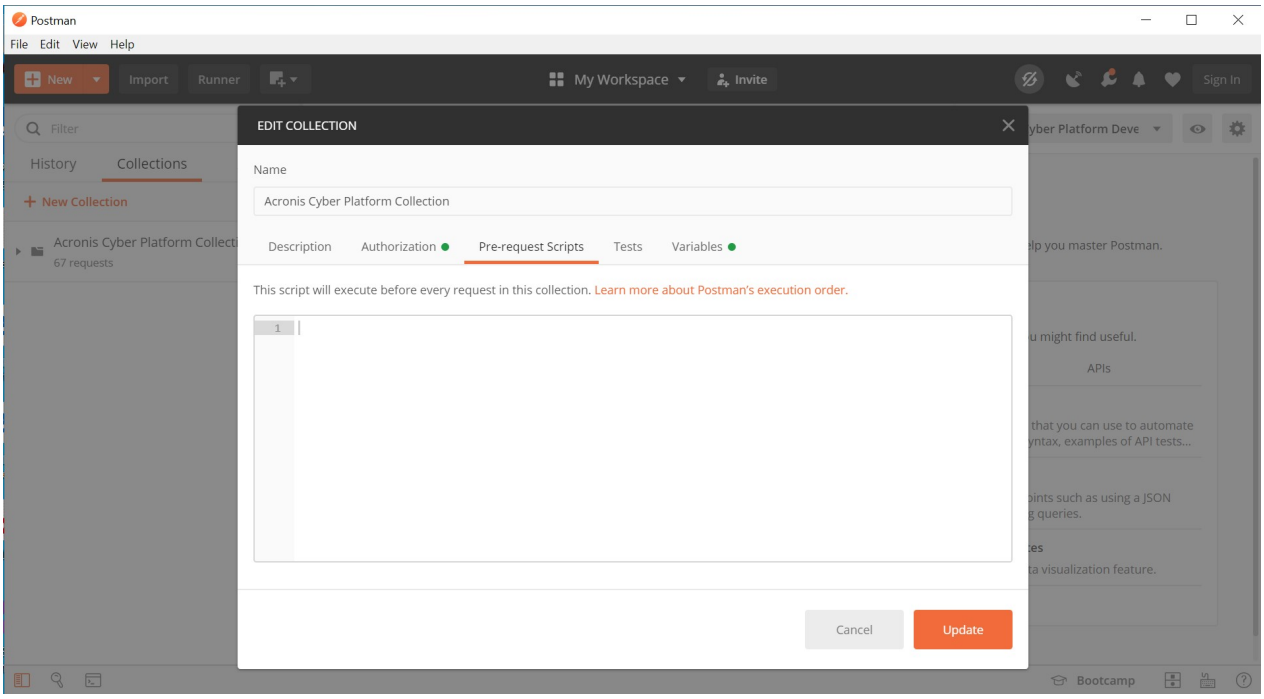
2. Enter into the tab the following code

```
pm.request.headers.add({key: 'User-Agent', value: 'Training/1.0 Acronis #CyberFit Dev
```



And click `Update` button.

3. So now let's check that it works.
4. Open `Create an API Client` and click `Send`.
5. Open the Management Console and check Audit log.



Summary

Now you know how to use base operations with the Acronis Cyber Platform API:

1. Create an API Client for the Acronis Cyber Platform API access
2. Issue a token for secure access for the API
3. Establish a simple procedure to renew/refresh the token
4. Create a partner and a customer tenants and enable offering items for them.
5. Create a user for a customer tenant and activate them.

6. Enable services for a user by assigning a role.
7. Receive simple usage information for a tenant.
8. Create and download reports for usage.

Get started today, register on the Acronis Developer Portal and see the code samples available, you can also review solutions available in the Acronis Cyber Cloud Solutions Portal.