Kofax Equitrac
API Reference Guide

Version: 6.4.0

Date: 2023-03-11

Table of Contents

Equitrac API Overview

Equitrac API provides a REST API to perform operations in the Equitrac product. It is designed to use from various client applications that needs integration with Equitrac. Customers can use our web hosted API in an industry standard way, so the clients can use any technology and can run on any devices, and sending requests to our API does not need uncommon, custom implementation.

Installation and configuration

The Equitrac API (EQAPI) is installed along with the Equitrac Core Accounting Server (CAS) in the ControlSuite Installer. There are no additional steps needed to install EQAPI.

If you have not already installed Equitrac 6.4, refer to the ControlSuite Installation help.

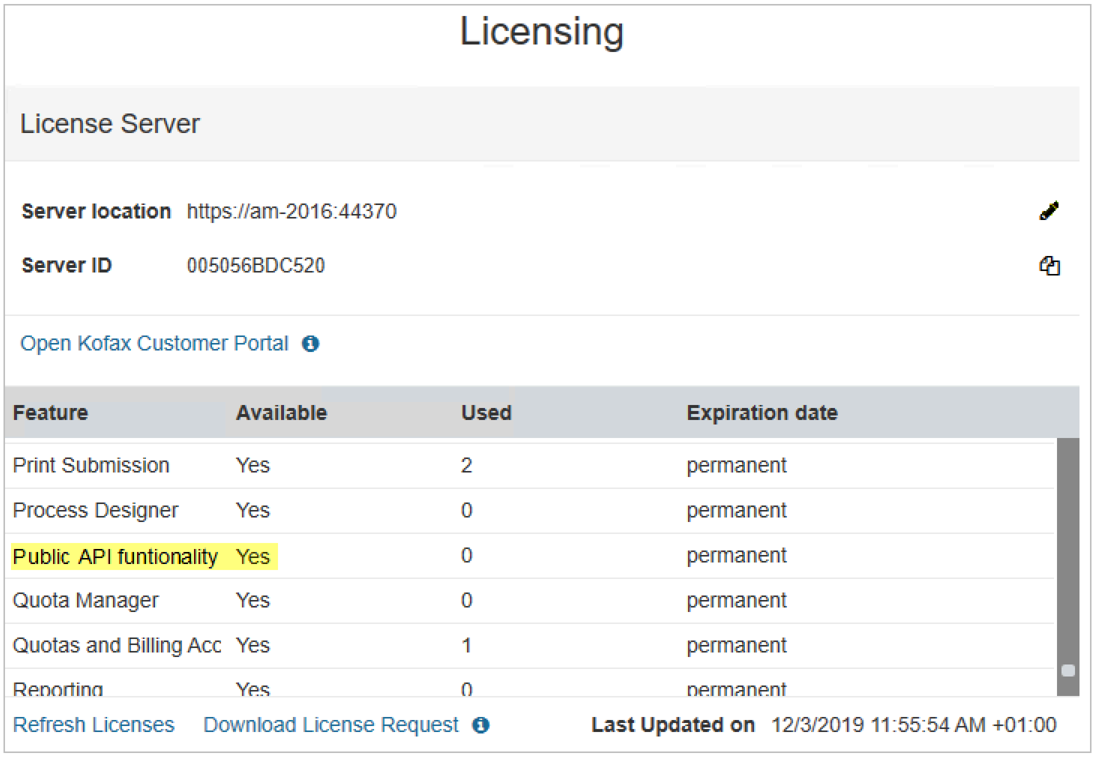
EQAPI Prerequisites:

- .NET Framework v4.6 or higher
- .NET Core runtime v2.2 or higher

After Equitrac 6.4 is installed, run the Configuration Assistant and set the following:

1. On the **Certificate Management** page, Generate a self-signed certificate or Import a custom certificate. EQAPI can only be used through a HTTPS connection.
2. On the **CS Enrollment** page, enroll the following services into the Security Framework:
   - **Equitrac API**
   - **Web Client**
   - **Core Accounting Service**

   Once the EQAPI is installed and setup, it can be accessed on the ’https://localhost:8282/equitracapi/’ URL.
3. On the **Licensing** page, ensure that the **Public API functionality** license entitlement is assigned to your system.



If EQAPI is not licensed, all API entry points (except the Status API) returns with an 402 response code. The license status of EQAPI can be acquired with the Status API. Refer to the Kofax Knowledge Base webpage for licensing support.

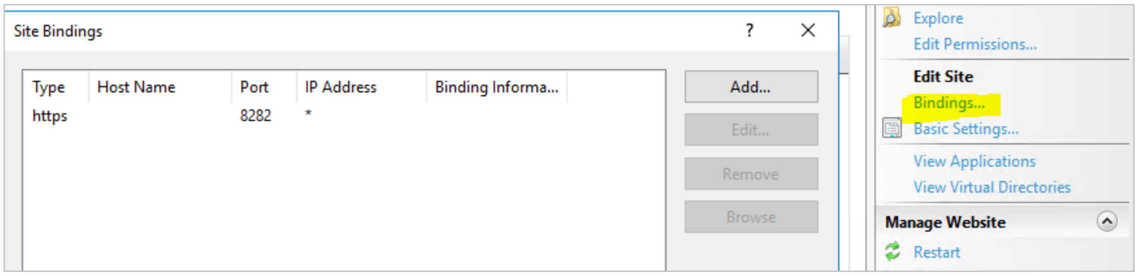
IIS configuration

EQAPI is a public API, and can be used in different network environments.

> By default EQAPI is installed in IIS and listens on port 8282 with HTTPS protocol. The IIS configuration can be changed manually to suite your environment.

The following are EQAPI IIS settings:

- Application pool: EquitracAPI with the default IIS user.
- Default user: IIS built-in user.
- Web site: Equitrac API
- Default configuration: HTTPS binding with the certificate selected in Configuration Assistant, listening on port 8282.



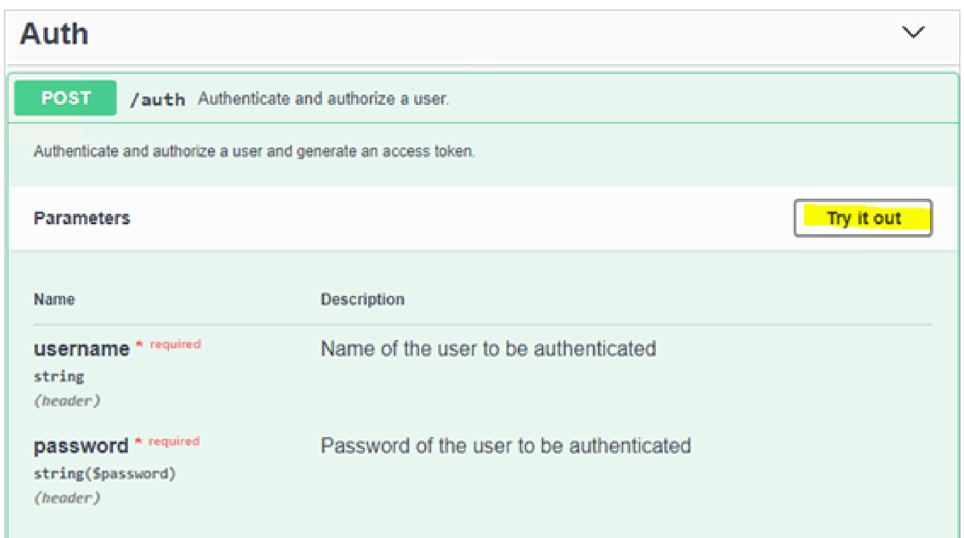
- Web application: EquitracAPI under the EquitracAPI web site.

Authenticate and authorize a user

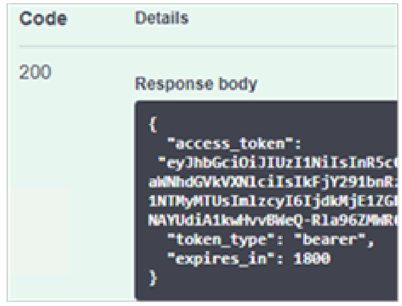
Authentication and authorization in EQAPI works according to the OAuth 2 standard. OAuth 2.0 is the industry-standard protocol for authorization by providing specific authorization flows for web applications, desktop applications and mobile devices.

To authenticate and authorize a user with Auth API, and generate an access token, do the following:

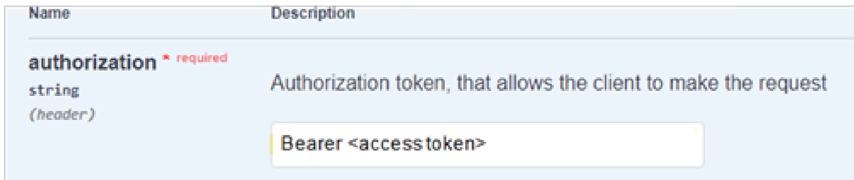
1. Send a **POST request** to the Auth API (/auth). Enter the **username** and **password** for the Windows user to be authenticated.



2. The API sends a response code a (HTTP 200), and delivers an access token with an expiration time (in seconds) in the response body. This access token contains the available roles of the user.



3. To perform an operation, send a request to the related API entry point with the access token. Set the authorization request header to **‘Bearer <access token>’**.



The access token is valid for 30 minutes before it expires. If the authorization request is made after the access token expires, send a Post request again to the Auth API to get another access token to use.

Setting access permissions

The EQAPI requires special roles to perform certain actions in Equitrac.

To set up roles in Equitrac, do the following:

1. Launch the EQ Web Client application or enter https://EQWebClient.hostname/EQWebClient in the address bar and log in to Equitrac Web System Manager.
2. In System Configuration, select **Global Configuration Settings > Security and Authentication > Access Permissions**.
3. Assign the Windows or domain groups to the following roles used by EQAPI:
   - **Admin**: With this role all operations can be performed.
   - **Accounts**: With this role, accounts (users, departments, billing codes) can be managed (create, retrieve, update, delete). It is a subset of the ‘Admin’ role.
   - **Cashier**: With this role the money balance of the accounts can be adjusted. It is a subset of the ‘Admin’ role.
4. Click the **Edit** icon for the group link you want to configure.
5. In the **Select Group** dialog box, choose the group to have access to the Equitrac functions, and click **OK** to save the changes.

API reference

EQAPI has a web page where all API entry points are documented and can be played with them. It is called the Swagger page, see in the next section. Every operations are well documented and self-explanatory there, including the possible parameters and the possible results.”

ApiError

All non-success responses contain ApiError object(s). This object provides a unique business error code for possible error cases and a textual message.

```
ApiError ⌄ {
    description:          Represents an error from all possible error cases
    code                  integer($int32)
                          Available values :
                          NoError = 0,
                          UserInvalidCredential = 1001,
                          UnAuthorized = 1002,
                          Forbidden = 1003,
                          NameAlreadyExists = 2001,
                          NameNotFound = 2002,
                          AccountInvalidCredential = 2003,
                          InsufficientFunds = 3001,
                          InsufficientColorQuota = 3002,
                          ChargeAccountIsLocked = 3003,
                          AccountIsLocked = 3004,
                          ParameterIsInvalid = 4001,
                          ServiceCommunicationError = 5001,
                          ServiceConnectionError = 5002,
                          Unlicensed = 6001,
                          UrlMismatch = 7001,
                          ParameterTypeMismatch = 7002,
                          HttpsOnly = 7003,
                          UnsupportedApiVersion = 8001,
                          UnspecifiedError = 8002

                          Enum:
                           > Array [ 20 ]
    message               string
                          Textual message to increase understanding the error situation
}
```

The business error code can be used to automate the processing of the responses. The textual message is a human readable description of the error code.

All business like errors are delivered under the response code 400 (Bad Request). Business like error means that the EQAPI infrastructure is healthy, but some regular business process generates an error. For example, if the user wants to add an account which already exists, the service will return a “The given name already exists!” error.

Example for possible business error codes (with human readable texts):

```
[
  {
    "code": 2001,
    "message": "The given name already exists!"
  },
  {
    "code": 4001,
    "message": "The amount should be a number!"
  },
  {
    "code": 4001,
    "message": "A parameter was missing or has invalid format or value!"
  }
]
```

There are other errors which are not strictly related to the business operation, but they are also contain an ApiError object in the response.

Example for “non business” errors (in this case for HTTP 403):

403

Insufficient role

Example Value | Model

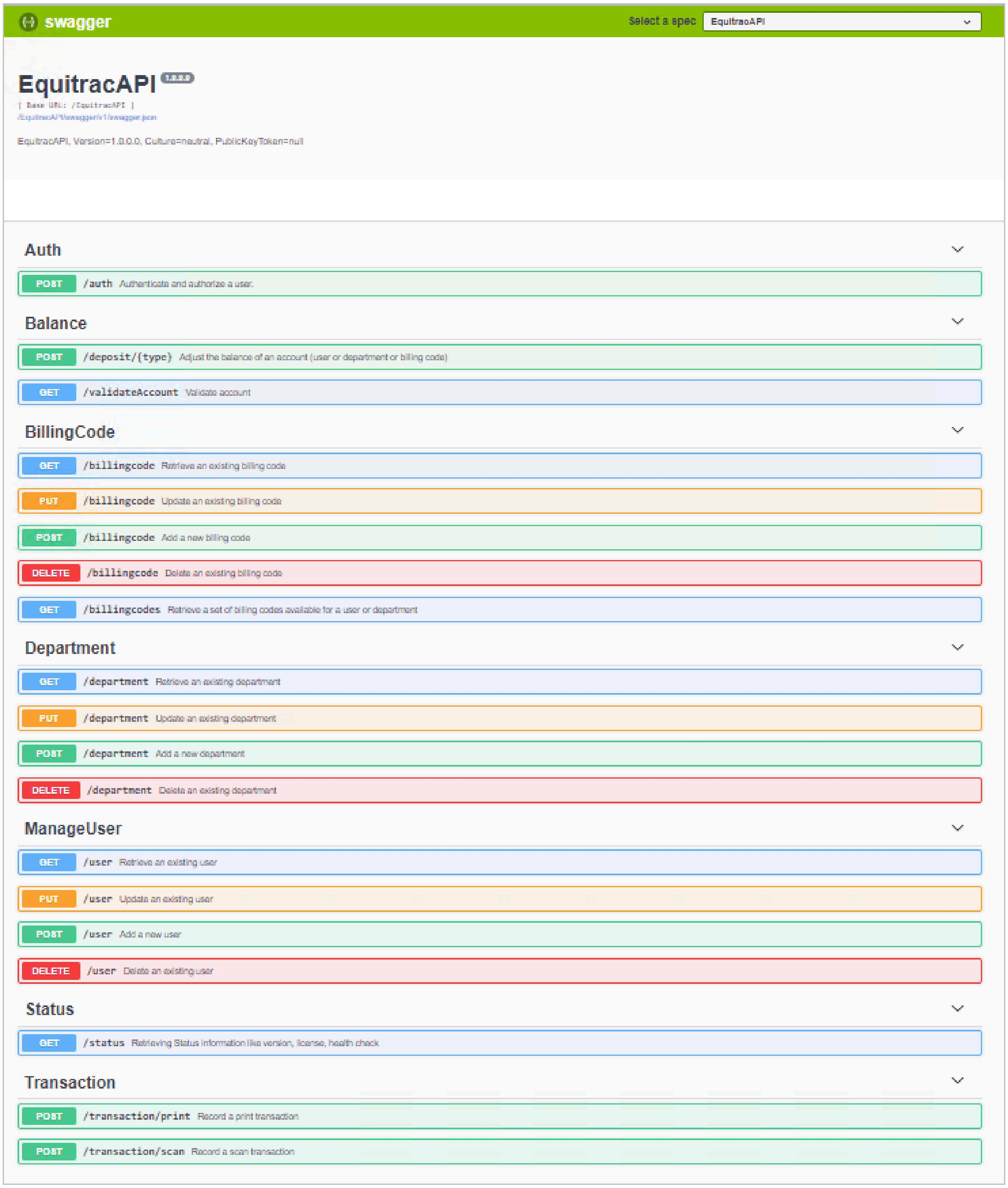
application/json

```
{
  "code": 1003,
  "message": "This user does not have role to perform this operation!"
}
```

Swagger page

EQAPI has a Swagger web page with a collection of all API entry points, where you can see the input and response parameters, and send requests to them.

The Swagger page is accessed at the EQAPI root URL. (Default is ‘https://localhost:8282/equitracapi’).



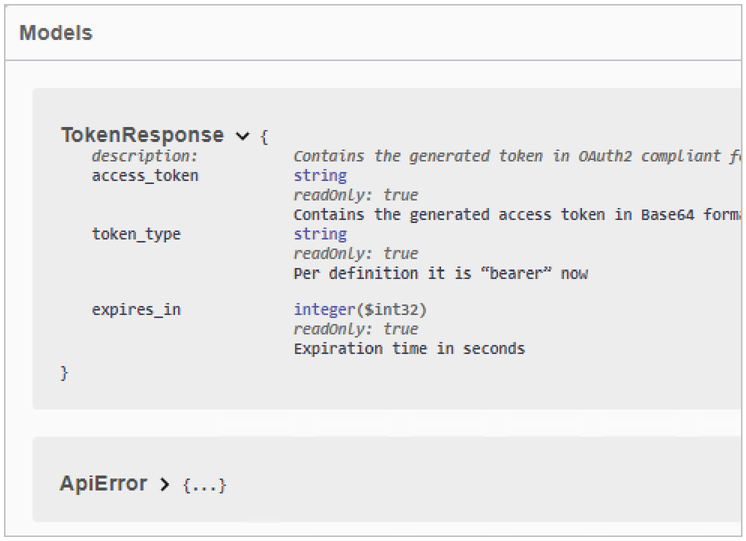
For every API entry points there are description about the operation, the input parameters and all possible responses. There are example values for the parameters for reference. Use the **Model** view to see the detailed description about the parameter fields. The mandatory fields are marked with a red asterisk.

Parameters

Example Value | Model

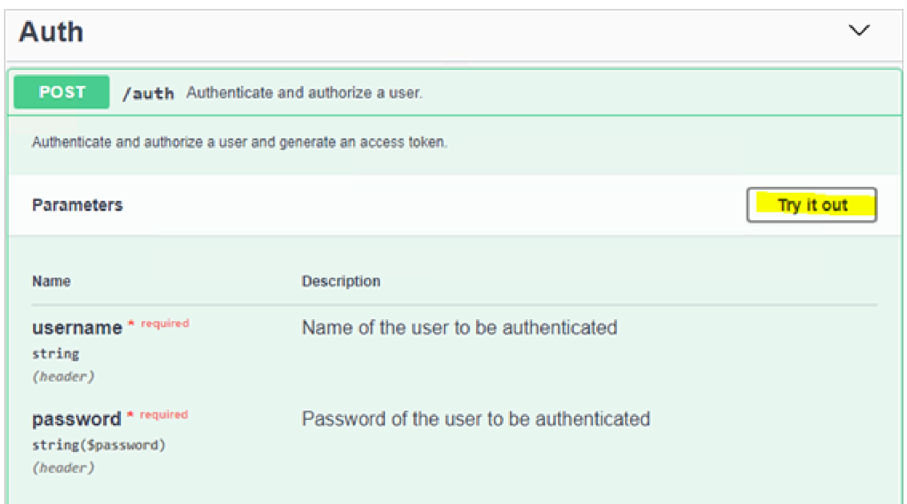
```
ManageUserDto ∨ {
    description:            Properties for a manage user operation
    accountName*            string
                            maxLength: 255
                            Account name
    fullName                string
                            maxLength: 100
                            Full name
    eMailAddress            string
                            maxLength: 100
                            Email address
    balance                 number($double)
                            Balance
    hardLimit               number($double)
                            Hard limit (the lowest balance allowed)

    primaryPin              string
                            maxLength: 255
                            Primary PIN
    secondaryPin            string
                            maxLength: 255
                            Secondary Pin
    alternatePrimaryPin     string
                            maxLength: 255
                            Alternate primary Pin
    homeDre                 string
                            maxLength: 255
                            Home DRE
    colorQuota              integer($int32)
                            Set this to -1 for unlimited.

    colorPageCount          integer($int32)
                            Color page count
    locked                  boolean
                            Is the account locked?
    department              string
                            maxLength: 100
                            Department
    scanHomeFolder          string
                            Scan home folder
}
```

All API models are described at the bottom of the page in the **Models** section.

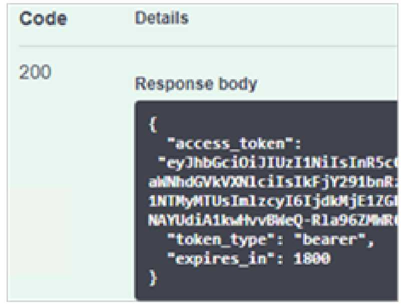


As was previously mentioned, every operation needs an access token with proper roles. Handling of the access token is automated on the Swagger page, and there is no need for any manual copy operation:

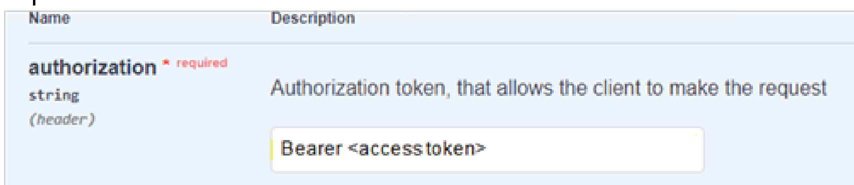
1. First authenticate the user with the Auth API.



2. Click the Try it out button to initiate the authentication. If the given credentials are valid, after the Execute button clicked, the API sends a response code (HTTP response code is 200), and delivers an access token with an expiration time (in seconds) in the response body.



3. This token will be stored on the page automatically and is copied to the field where it is needed later by other operations.



API descriptor

The API descriptor is a JSON file which is downloadable from the Swagger page. This file can be used to generate a source code for programming languages (such as Java, C#, C++) to access the EQAPI easily.

1. Open the Swagger page. (Default is ‘https://localhost:8282/equitracapi’).
2. Click on the JSON link under the EquitracAPI heading (for example, /EquitracAPI/swagger/v1/swagger.json) to display the JSON content.
3. Download the JSON descriptor file.

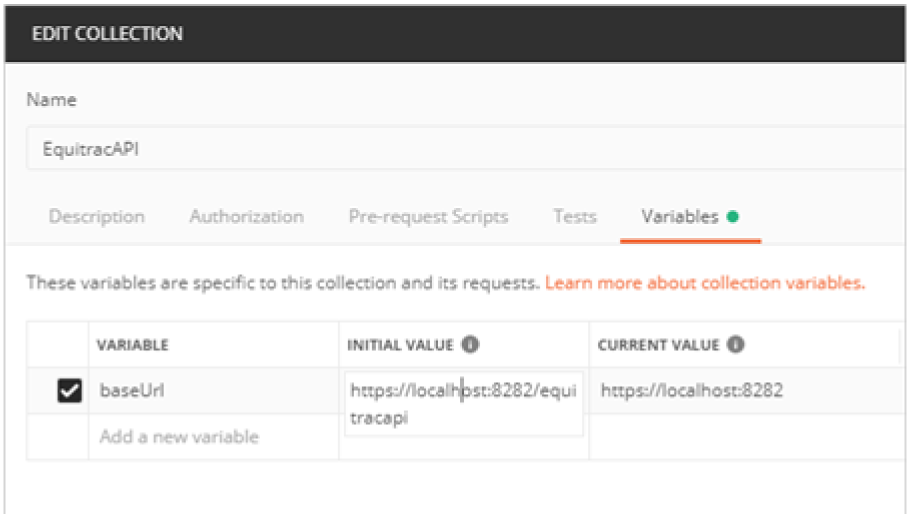
Postman profile

With the EQAPI, a Postman collection is deployed and can be imported into the Postman API application to make requests to EQAPI. The Postman collection is in **<Installation folder of ControlSuite>/Equitrac/EquitracAPI/ EquitracApi.postman_collection.json**.

To use the Postman collection, do the following:

1. Open the Postman application, and import the collection (EquitracApi.postman_collection.json).
2. If Postman is used on another machine, set the EQAPI root URL:

    a. Click **Edit** in the EquitracAPI collection menu.

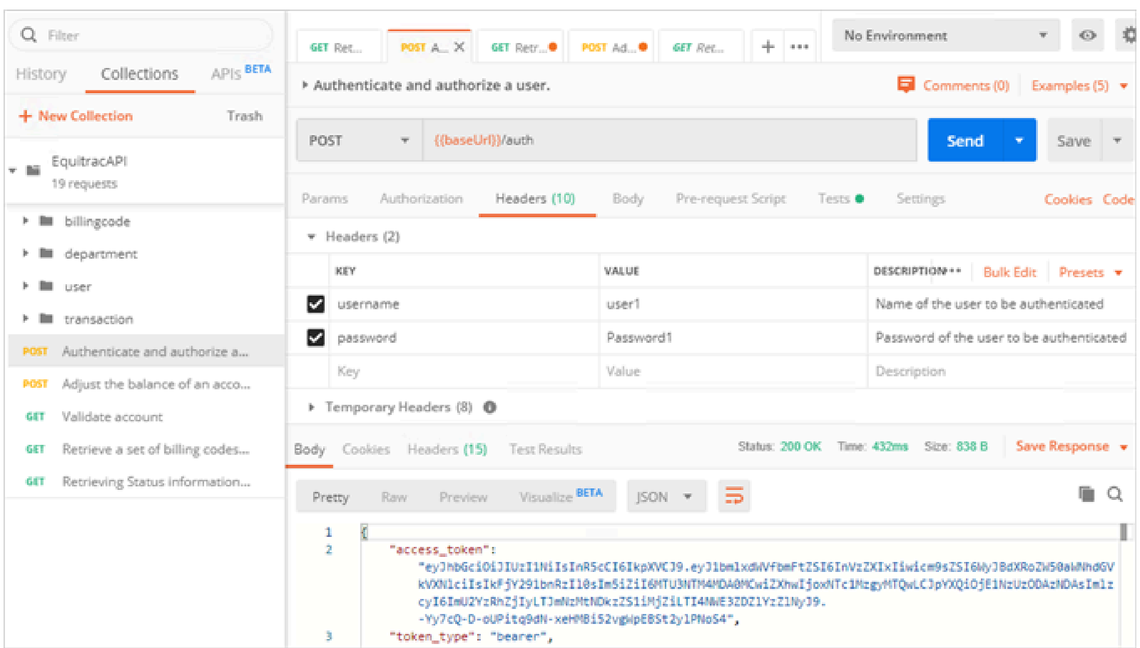


   b. Change the current value of the **baseUrl** variable to the network address of EQAPI.

   c. Click **Update**.

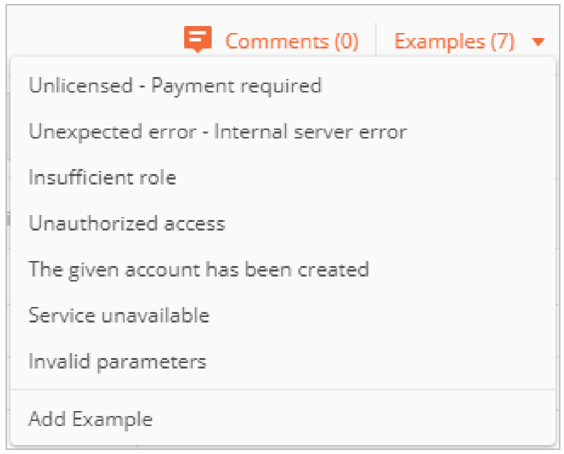
3. This Postman collection will use the access token automatically.

   a. Send a request to the Auth API with valid user credentials. Once the credentials are validated, the access token appears in the response body. This access token will be used automatically for all subsequent operations.



   b. Perform any other operation. The access token is used automatically.

4. There are many examples for each operation.


Comments (0)
Examples (7)
Unlicensed - Payment required
Unexpected error - Internal server error
Insufficient role
Unauthorized access
The given account has been created
Service unavailable
Invalid parameters
Add Example