How to Test Client Passwordless Login

Passwordless is a functionality which allows login into the TSH app by using a simple link. This link contains a query string called ticket and the system validates and verifies the ticket to know if it’s valid and it contains an user’s information.

The ticket should be created by using the endpoint (in Postman)

*https://together1api.togetherseniorlife.com/api/v1/pwdless/ticket/user/{{user1Id}}* where *{{user1Id}}* is a valid user ID (Users IDs in [this document](https://docs.google.com/spreadsheets/d/1D3Yo8oNHWtY3qCpRryKT0nwVvUSM-WTCx3GzjEkuCsw/edit#gid=24294604)).

After the ticker is generated you could use it to navigates into the TSH app by building an url like this:

<https://together1.togetherseniorlife.com/session/upcoming?ticket=hez3dV2nqTRae7hDVoSHpLucqCFJy6OtCiuQA0R3kDbWJ55cBS4PZ7AZ0m5D8Qjf>

Notice you can mix the the passwordless parameter with the ***forceTime*** parameter described [here](https://docs.google.com/document/d/1TAZOiGsgnV8Ae9bVy9543ayexXfc9YzQ1shYsGvkUZs/edit#) to get something like:

<https://together1.togetherseniorlife.com/session/upcoming?ticket=hez3dV2nqTRae7hDVoSHpLucqCFJy6OtCiuQA0R3kDbWJ55cBS4PZ7AZ0m5D8Qjf&forceTime=2020-10-09T17:30:00.000Z>

More information about how to generate ticket please [stomas@togetherseniorhealth.com](mailto:+stomas@togetherseniorhealth.com) or [diego.castro@evolvingweb.ca](mailto:diego.castro@evolvingweb.ca)

**Note**: Instructors cannot use passwordless tokens to join classes, they must login using their username/password. However instructors (or any user with username/password) can login using the login option from the menu dropdown. Simply use the URL above (Or click on the “Sessions” link at the top of the page) and append ?forceTime=<utctime> at the end of the URL (without the ticket).

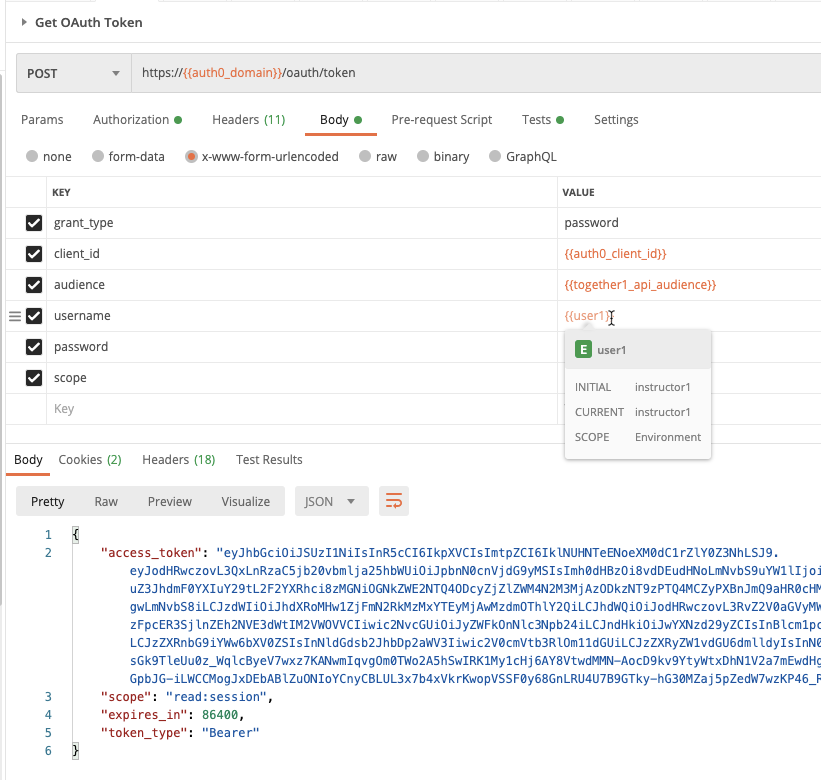
# 

# Generating Tokens, Tickets and Getting Dates Using Postman to get an API

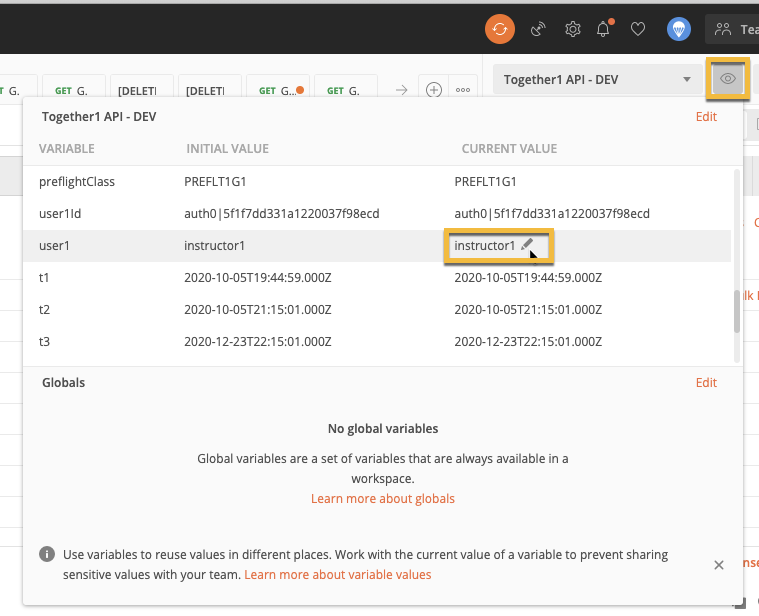
There is a Postman collection that is set up for testing and development purposes. That collection has scripts that copies values from some requests into variables that can be used in subsequent requests.

**Get OAuth Token**

For example, to generate a token that can be used to authenticate for any API call, use the “Get OAuth Token” request:



The token is generated for the user whose name is in the {{user1}} parameter (instructor1 in this example). To change the variable, edit the appropriate environment variable:

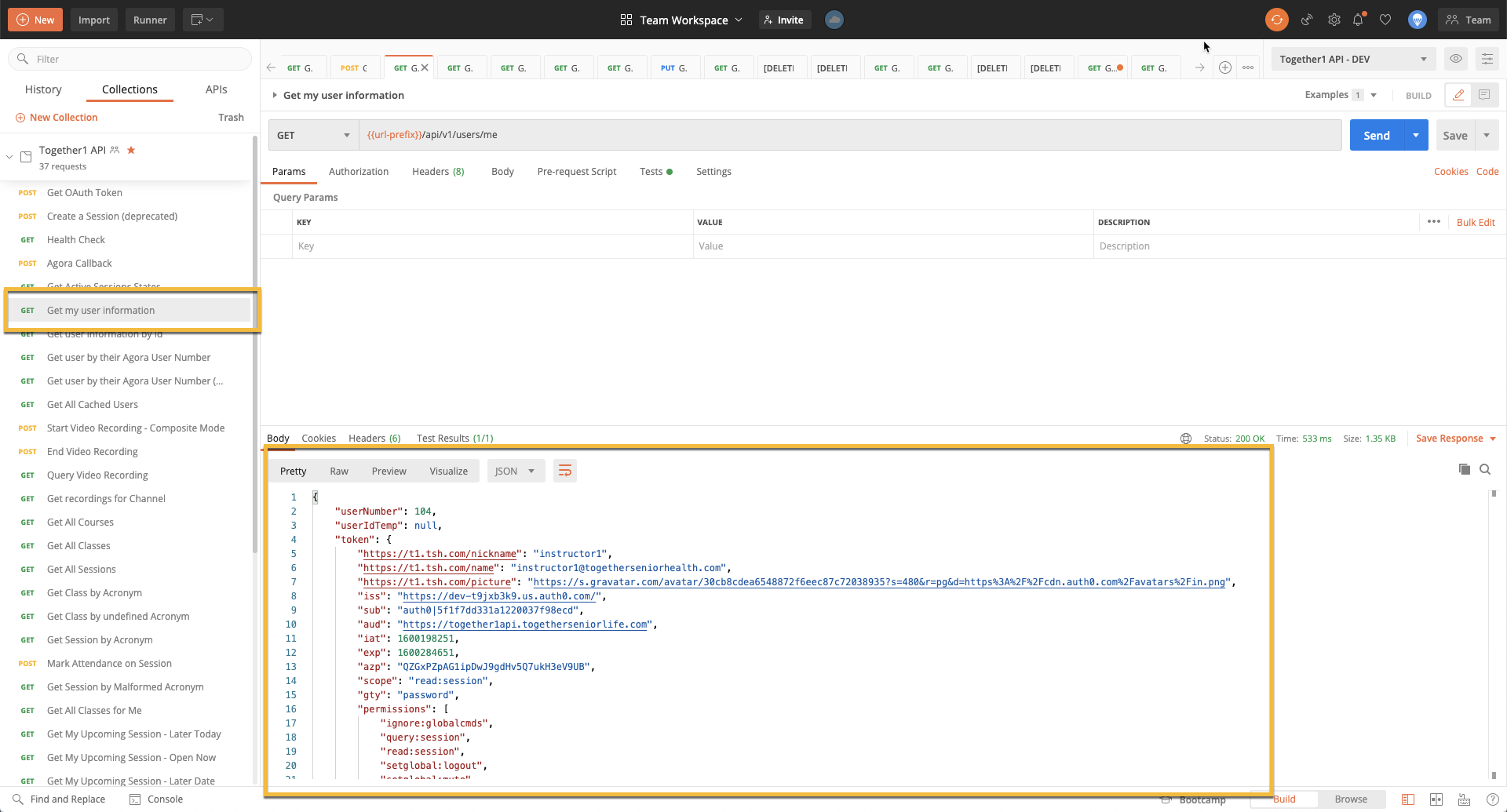


**Note**: Be sure to always change the “Current Value” - this will ensure that the value you use is not changed for other developers! (always use the “eye” icon to edit data)!

Calling this request will set the token returned to a variable named {{token}} to contain an authentication token for the user you specified. This token is then used to authenticate the user in subsequent Postman calls.

**Get my user information**

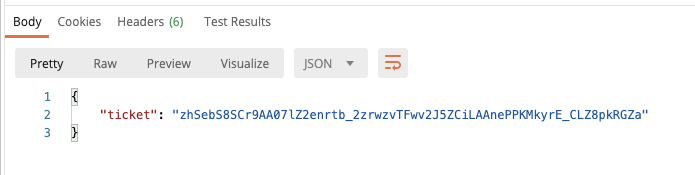
Next, you can sanity check the current user using the “Get my user information” request to see the user information in the result:



**Generate a Ticket for a User**

To generate a ticket for testing of a specific user, follow these steps:

1. Get the User’s ID from [This spreadsheet](https://docs.google.com/spreadsheets/d/130xwXi1y-zsGiO_N-T9zvONtSZ9lw1tiEFP1UUSABjo/edit?usp=sharing) (for example, ‘auth0|5f2aef95257b000038e487f7’ for instructor2)
2. Set the user’s ID to the current value of the {{user1Id}} variable
3. Call the request ‘Generate a Ticket for a User”. This will generate a ticket in the result:



1. Copy/paste the ticket for the value of the forceTime query string variable.

**Generate JWT Token from Ticket**

Note that after “Generate a Ticket for a User” the value of the ticket is in the {{ticket}} variable, which can then be used to generate a token by calling “Generate JWT Token from Ticket”. For fun, try calling “Get My User Information” to see the values within the token. You should notice that the token from the ticket contains fewer permissions than the token generated by “Get OAuth Token”

**Get All Classes for Me**

Another useful request for testing is the “Get All Classes for Me” request, which will get the classes for the current user, and also sets some Postman variables from the first class assigned for the user. The following variables are set:

// T1: Time before the first session

// TOpen: Time during first session

// T2: Time just after the first session closes

// T3: Time just after the last session closes

These variables are used to force the current date in the test cases whose names begin with “Get My Upcoming Session”. You can also copy/paste the corresponding variable values to reproduce the test case in the browser. For example, the value of {{topen}}, if used for ‘forceTime’ on the URL, will cause the landing page to believe that there is a session that you can join:

