Testing the System is done with Postman.

1. Import the collection SystemTestBanqiPostman.json.
2. Create an environment in Postman so that variables can be stored.
3. Start the server without a database connected (this will ensure that production isn’t affected).
4. Have Postman run the tests.
5. Ensure that each succeeds.

If you would like to do UI system testing, you need to load UISystemTestPrepBanqi.json into postman with an environment with url and port set to where the server is being hosted. See the instructions below to do that.

Create Collections

1. When you first open **Postman**, it will prompt for what template to load from.
2. Select a **Collection**, or close it if you already have a collection.
3. It will show up on the left.

Create Environments

Next, you'll want to create an environment so that you can use variables in your request, such as "url" to point to the location of your server.

1. In the top right, click on the cog wheel to the right of the eye ball.
2. Click Add, and name it.
3. You can create variables here
4. In most locations, to use the variable surround the variable with two curly brackets:
5. {{variable\_name}}

Create Requests

Next, you'll want to add requests to your collection.

1. Click the three dots on your collection located on the left, and click **Add Request**.
2. You can set the HTTP request type: GET, POST, PUT, etc.
3. You can set the body content under the **Body** tab.
4. You can click **Send** to send the request and view the response

Create Tests

Next, you'll need to create tests. This is under the **Tests** tab for a request.

Here's an example:

// Get responseBody and convert to JSON  
var jsonData = JSON.parse(responseBody);  
  
// Test that response is 200  
pm.test("response is ok", function () {  
 pm.response.to.have.status(200);  
});  
  
// Test that the data sent back matches what was expected.  
pm.test("status is good", function () {  
 pm.expect("expectedValueOfData").to.equal(jsonData.dataReturedByServer);   
});  
  
// Here's how you set an environment variable to be used by future requests.  
// This is a great way to "chain" requests which depend on the response  
// of the previous request.  
postman.setEnvironmentVariable("someVariableToStoreInfo", jsonData.informationForNextRequest);

Run Tests

Once you have a set of requests, you can run them in order as a test.

1. Click the arrow above the three dots on the collection.
2. Click **Run**, and the **Collection Runner** will open.
3. You can select a **Collection** and an **Environment**.
4. Click **Run Test** to run

Sources and Further Reading

Postman Docs on writing tests:

[https://learning.getpostman.com/docs/postman/scripts/test\_scripts/ (Links to an external site.)Links to an external site.](https://learning.getpostman.com/docs/postman/scripts/test_scripts/)

Postman Docs on using variables / environments

[https://learning.getpostman.com/docs/postman/environments\_and\_globals/variables/ (Links to an external site.)Links to an external site.](https://learning.getpostman.com/docs/postman/environments_and_globals/variables/)