**Demo workflow for working with user data.**

When you run the application you selects predefined user. That simulates application behavior concerning user roles (role is a set of permissions). At the moment you able to pick one of 3 users: anonymous, registered (normal) and administrator. User ID stores in environment variable NODE\_USER\_ID. You must render the website only in case when all auth steps finished without errors:

**1. Get user**

[GET] /user/:id

«:id» - is an ID of user that you can pick-up from NODE\_USER\_ID. You can read «docs/UserAPI.docx» if you want to learn UserAPI more detailed.

**Expected behavior:** you will receive User object

**2. Get user role and JUNK token (JUNK means that is a test data and test behavior)**

**These requests MUST be runnen PARALLEL and AFTER request #1**

**2.1. Get role (2.1 and 2.2 are parallel requests)**

[GET] /role/:id

«:id» - iis an ID of user that you can pick-up from NODE\_USER\_ID. You can read «docs/UserAPI.docx» if you want to learn UserAPI more detailed.

**Expected behavior:** you will receive Role object that contains all permissions of the user.

**2.2. Get token (2.1 and 2.2 are parallel requests)**

[GET] /user-token/:id

«:id» - is an ID of user that you can pick-up from NODE\_USER\_ID. You can read «docs/UserAPI.docx» if you want to learn UserAPI more detailed.

**Expected behavior:** you will receive String[6] data.

**3. Check data (validate token and user, request it only after #2.1 and #2.2)**

[GET] /user-validate/:id/:token

«:id» - is an ID of user that you can pick-up from NODE\_USER\_ID. You can read «docs/UserAPI.docx» if you want to learn UserAPI more detailed.

«:token» - result from 2.2. request.

**Expected behavior:** you will receive «true» or «false». «True» means that user is found and you able to render HOME page. «False» means that something happens bad and you should not to render website normally.