**Manager Soft**

|  |  |
| --- | --- |
| **Contributors** | Gabriel Sanchez |
| **Last Updated** | 2/27/2021 |
| **Version Number** | 0.0.1 |

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Change** | **Version** |
| Gabriel | March 31, 2020 | Added personas | 0.0.1 |

**Table of Contents**

[SECTION 1: INTRODUCTION 3](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.gjdgxs)

[1.1. Purpose 3](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.30j0zll)

[1.2. Terminology 3](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.1fob9te)

[1.3. User Personas 3](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.3znysh7)

[1.4. User Stories 3](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.2et92p0)

[1.5. Time Estimations 3](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.tyjcwt)

[SECTION 2: PROCESS DESIGN 4](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.3dy6vkm)

[2.1. Prototypes 4](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.1t3h5sf)

[2.2. ORD (Object Relational Diagram(s) 4](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.4d34og8)

[2.3. NoSQL Document Diagram 4](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.2s8eyo1)

[2.4. NoSQL Data Structure 4](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.17dp8vu)

[SECTION 3: QA TESTING 4](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.26in1rg)

[3.1. QA Test Plan 4](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.lnxbz9)

[3.2. Unit Tests 4](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.35nkun2)

[SECTION 4: REFERENCES 5](https://docs.google.com/document/d/1nj1KLza97zPvFoFhProgZgd4M2QtXdfa/edit#heading=h.1ksv4uv)

**SECTION 1: INTRODUCTION**

**Purpose**

Manager Soft is a software to keep track of small parts rooms, it could be IT, Mechanic or any other small parts room. This software is going to keep track of inventory, check in, and check out and move inventory to another warehouse. When inventory goes low, an email will be sent to the admin who orders the parts.

Terminology

|  |  |
| --- | --- |
| **Name** | **Comments** |
| *MEAN Stack* | *MongoDB, Express, Angular, and Node.js* |
| *VS Code* | *Development IDE* |
| *SoapUI* | *API testing tool* |
| *MongoDB Compass* | *Database development IDE* |
| *Angular Material* | [*https://material.angular.io/*](https://material.angular.io/) |
| *Flex-layout* | [*https://github.com/angular/flex-layout*](https://github.com/angular/flex-layout) |

**User Personas**

**User 1:**



**Gabriel Sanchez,** 29 years old IT technician.

An IT technician who constantly grabs parts from the IT room and always end up wondering where the parts are located. He always complain when the inventory goes low. It is really hard for him to find a part from the IT room.

**Hobbies:** Coding, Drinking.

**Skills:** Is very technical and understand the concept of the needs from people.

**User 2:**

Bradly Dean, he is a 37 years old IT technician.



Bradley Dean keeps tracks of the inventory in the parts room, he organizes the parts and makes sure that thigs get ordered.

**Hobbies:** He likes to read.

**Skills:** Excellent computer skills.

**User 3:**



Mary Davis: A very nice IT manager.

She makes sure that the IT department has the tools to perform the job. She is on charge of the IT department and is needs to have the parts room always on top.

**Hobbies:** likes to paint and go hunting

**Skills:** Very technical

**User Stories**

As a regular user, I would like to know where the parts are, so I can spend more time doing my job instead of waste my time looking for parts.

As a regular user, I would like to log in with number, so that I don’t forget my password.

As a regular user, I would like to check in and check out inventory, so I can do my job faster.

As a regular user, I would like to see the things I have take out from the warehouse, so I can have a proof.

As a regular user, I would like to have a function to look if we have a part and the inventory available, so I don’t waste my time looking for it.

As the regular user, I would like to have a function to associate the part to a computer, worker, user or thing, so we keep track of where it goes.

As the Admin, I would like to inventory all parts with a barcode, so regular users just scan and make a check out and check in.

As the admin, I would like to have a function to move inventory form warehouse to warehouse, so we can see where the inventory is located.

As the Admin, I would like to have reports of the low inventory, so I can order more assets.

As the admin, I would like to see who takes parts out of warehouse, so I can have a proof.

As the Manager, I would like to see who takes parts of the warehouse, so I can have an idea who is really using more assets.

As the Manager, I would like to see the parts that we order more constantly, so I can have an idea of the more ideal stuff.

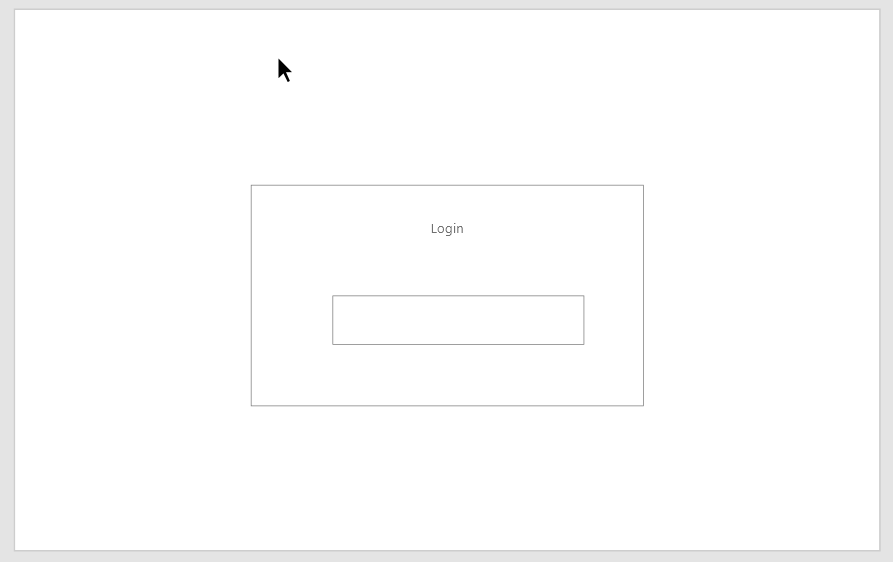
As the Manager, I would to see pictures, so I can see what I the product is.

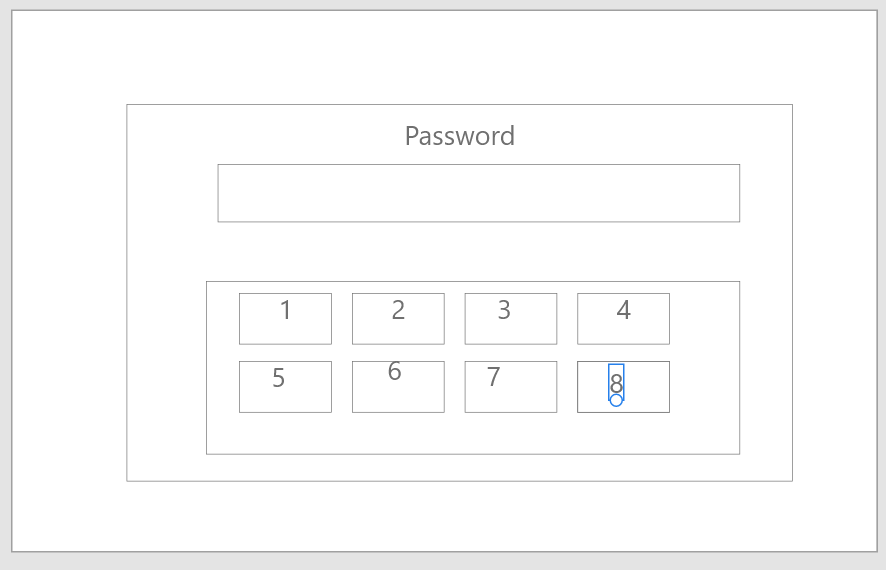
**Time Estimations**

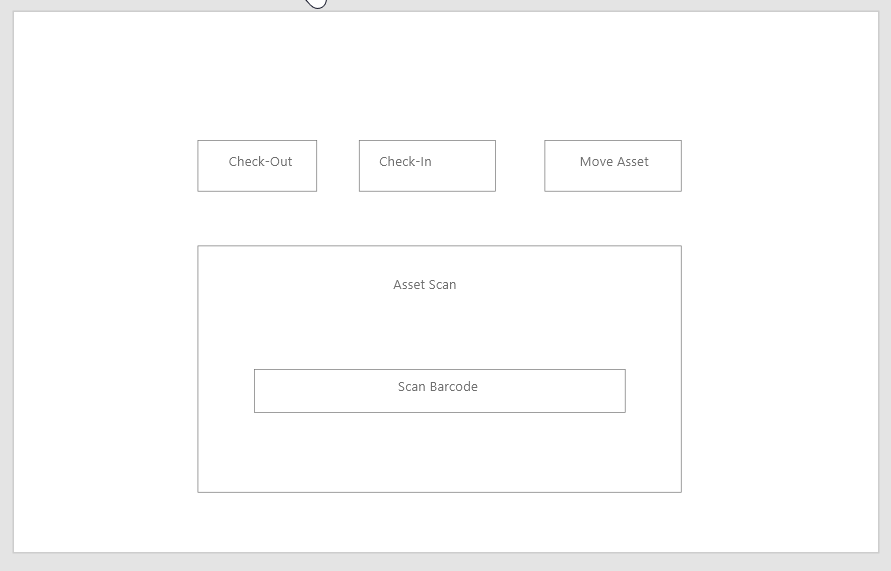
|  |  |  |
| --- | --- | --- |
| **Story** | **Hours** | **Points** |
| Set up APIs | 4 | 2 |
| Set up Schemas and Models | 2 | 1 |
| Set up Angular Material | 1 | 1 |
| User roles | 2 | 2 |
| Set up UI | 2 | 1 |
| Invoice function | 5 | 3 |
| Set up AuthGuard | 1 | 1 |

**SECTION 2: PROCESS DESIGN**

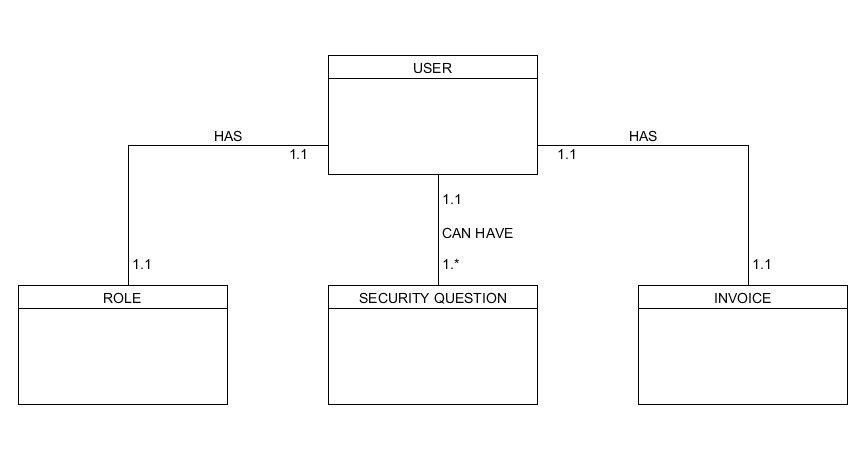
**2.1. Prototypes**



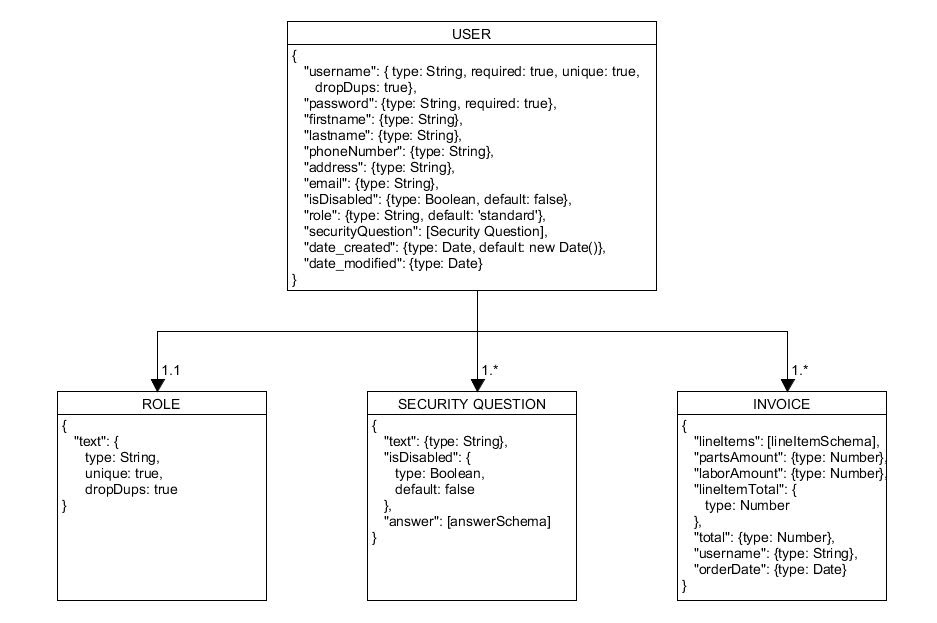


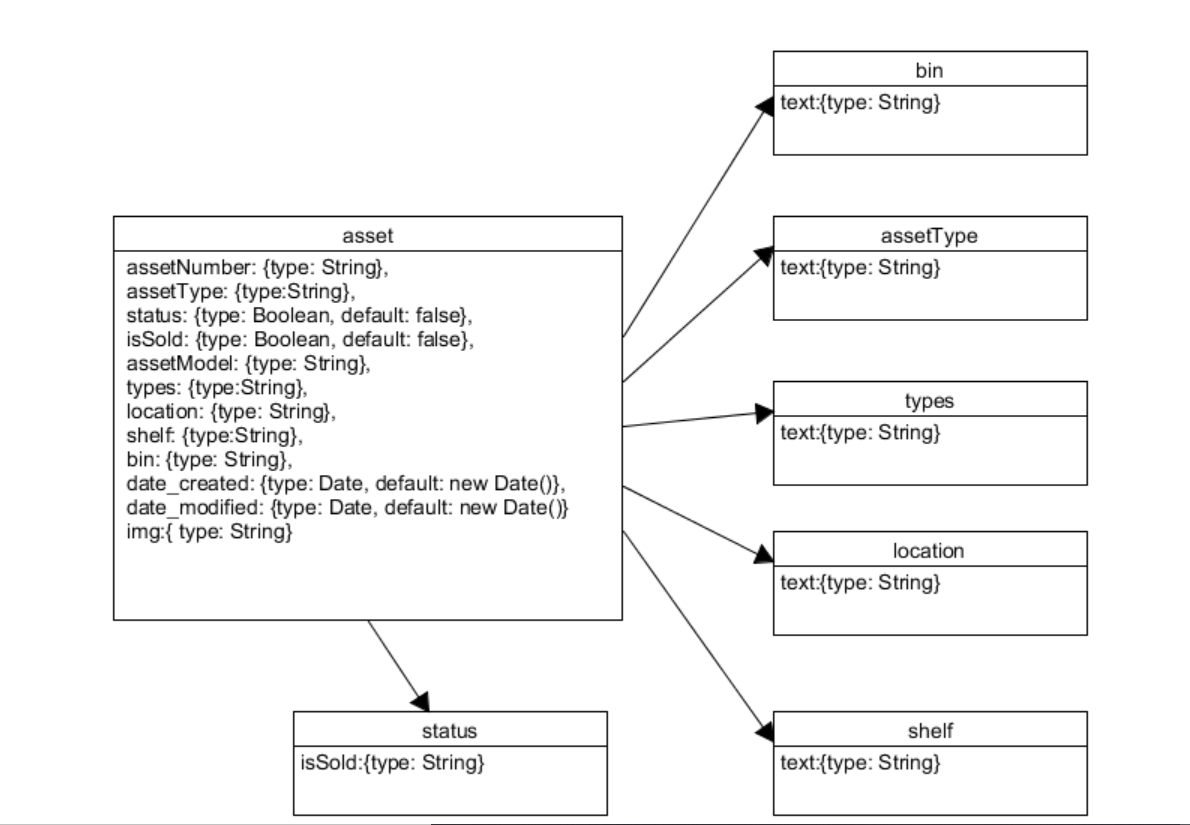


**2.2. ORD (Object Relational Diagram(s)**



**2.3. NoSQL Document Diagram**





**2.4. NoSQL Data Structure**

 //In user

{

“username”: “eriblue”,

“password”: “jdiYkkbhh8KMynhj”,

“firstname”: “Erick”,

“lastname”: “Blue”,

“phoneNumber”: “402-229-2233”,

“address”: “3990 L st”,

“email””: “erickblue@gmail.com”,

isDisabled: false,

“role”: {

“text”: “standard”

},

“date\_created”: “04/01/2020”,

“date\_modified”: “04/02/2020”,

“securityQuestion”: [

                 {

“text”: “What is your favorite color?”,

isDisabled: false,

“answer”: [

“questionId”: “20398ij30923”,

“text”: “Red”

]

}

],

 “invoice”: [

  {

         “lineItems”: [

                     {

“title”: “example”,

                     “price”: 109

},

],

“partsAmount”: 35,

“laborAmount”: 33,

“lineItemTotal”: 109,

“total”: 177,

“username”: “eriblue”,

“orderDate”: “04/02/2020”,

}

]

}

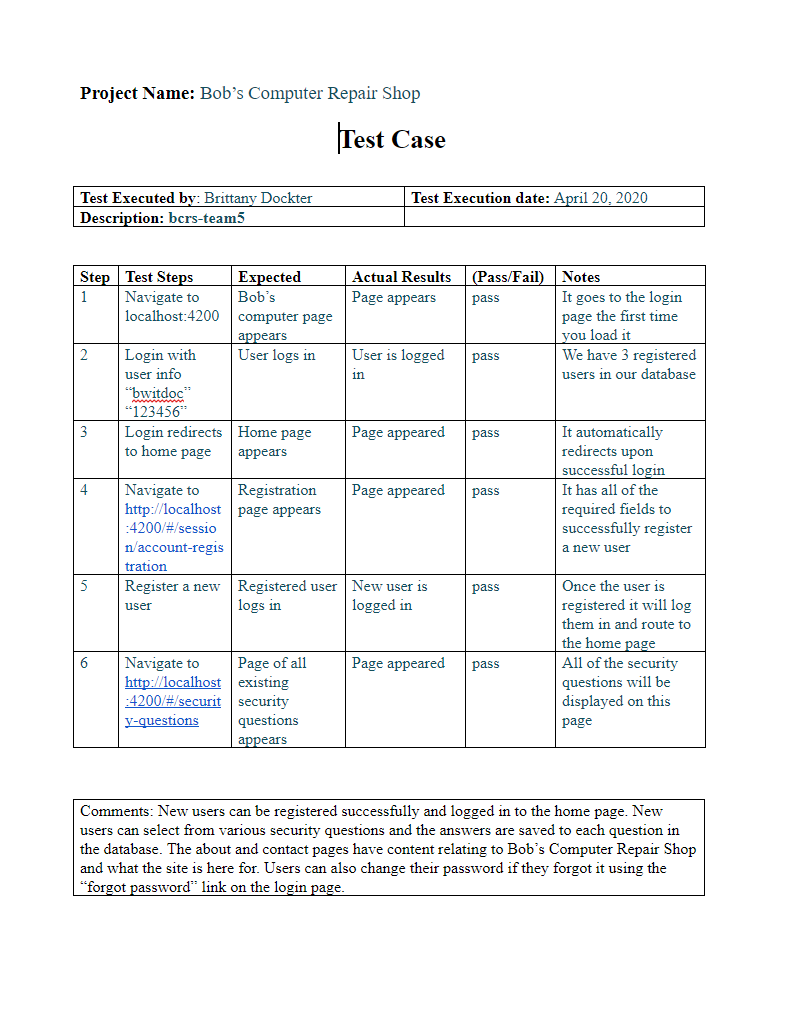
**SECTION 3: QA TESTING**

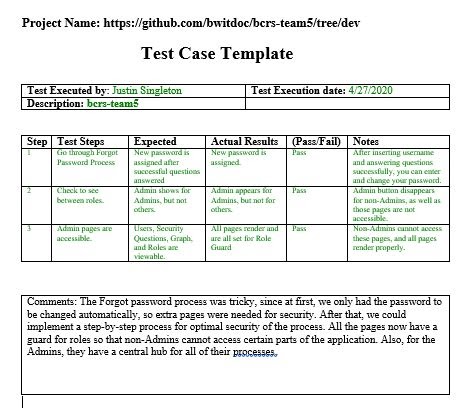
**3.1. QA Test Plan**

**Project Name:** https://github.com/bwitdoc/bcrs-team5

**Test Plan**

|  |  |
| --- | --- |
| **Tests Executed by**: Brittany Dockter | **Tests Execution date:** April 2020 |
| **Description:** The tests that will be conducted will be small Unit Tests, to make sure that the functionality of the site works as intended. That is to include rendering of all pages, sign-in, register, account access, admin access, item purchasing, invoicing, etc. | |



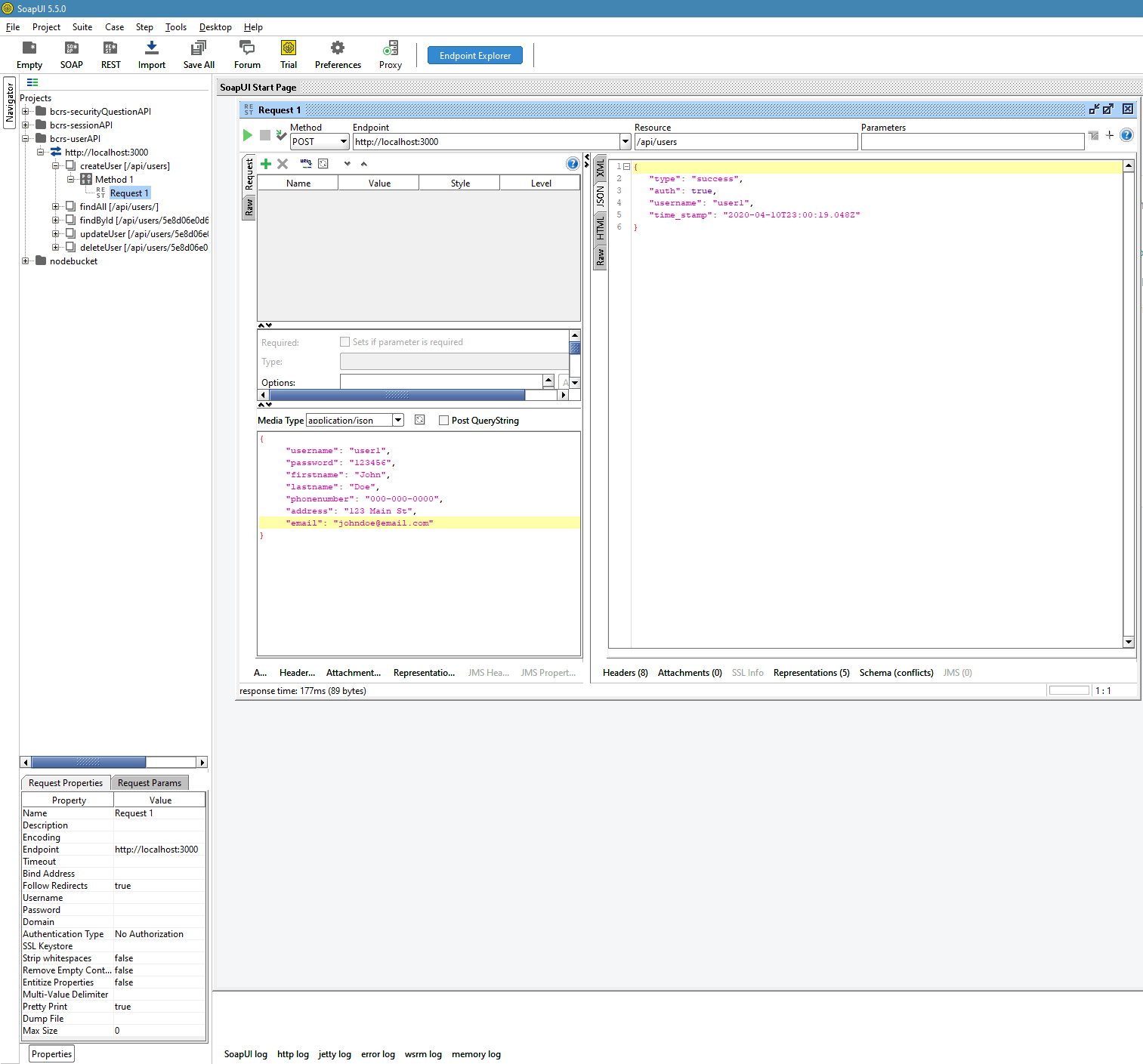


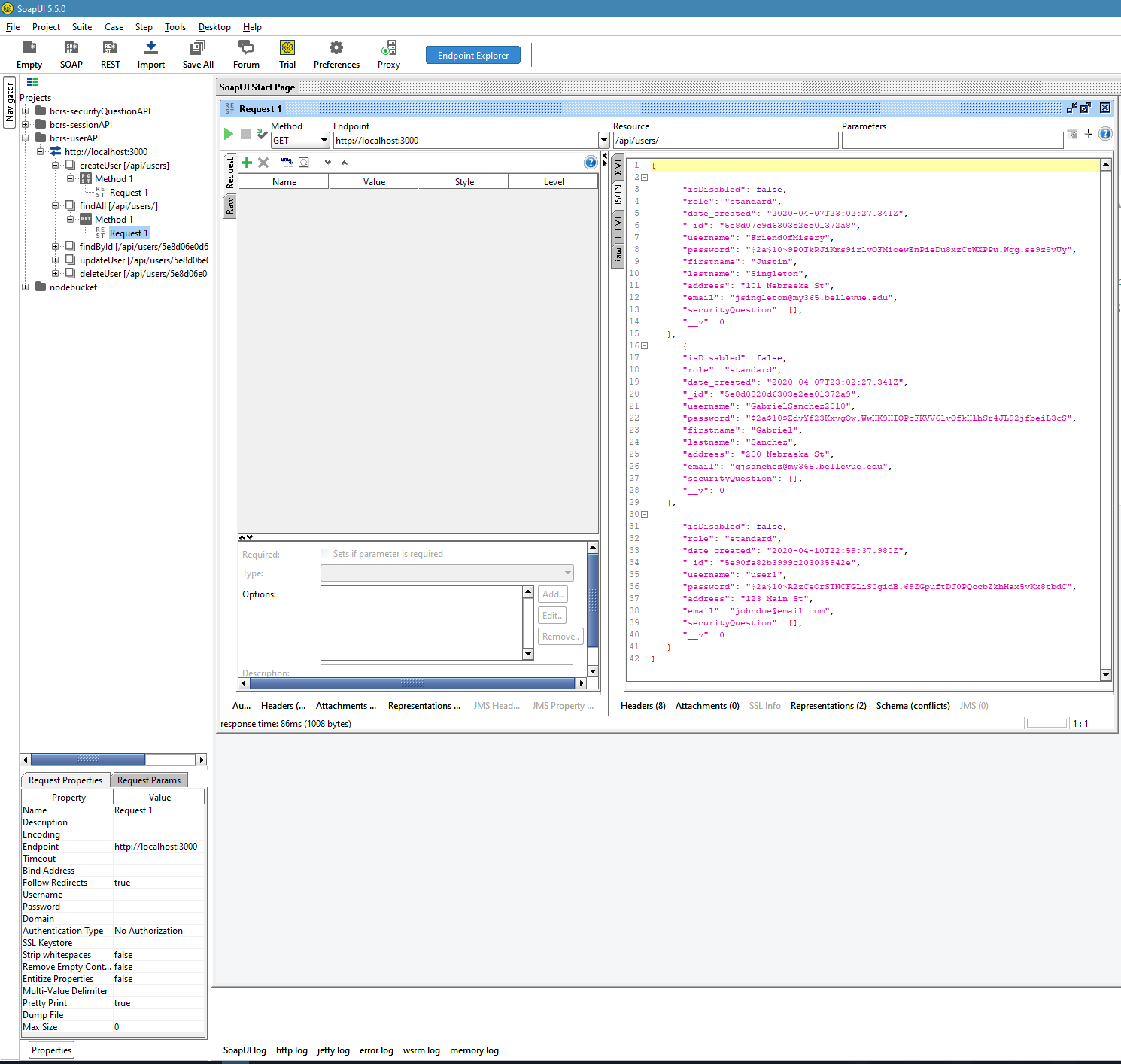
|  |  |
| --- | --- |
| **Test Executed by**: Gabriel Sanchez | **Test Execution date:** 04/28/2020 |
| **Description:** bcrs-team5 |  |

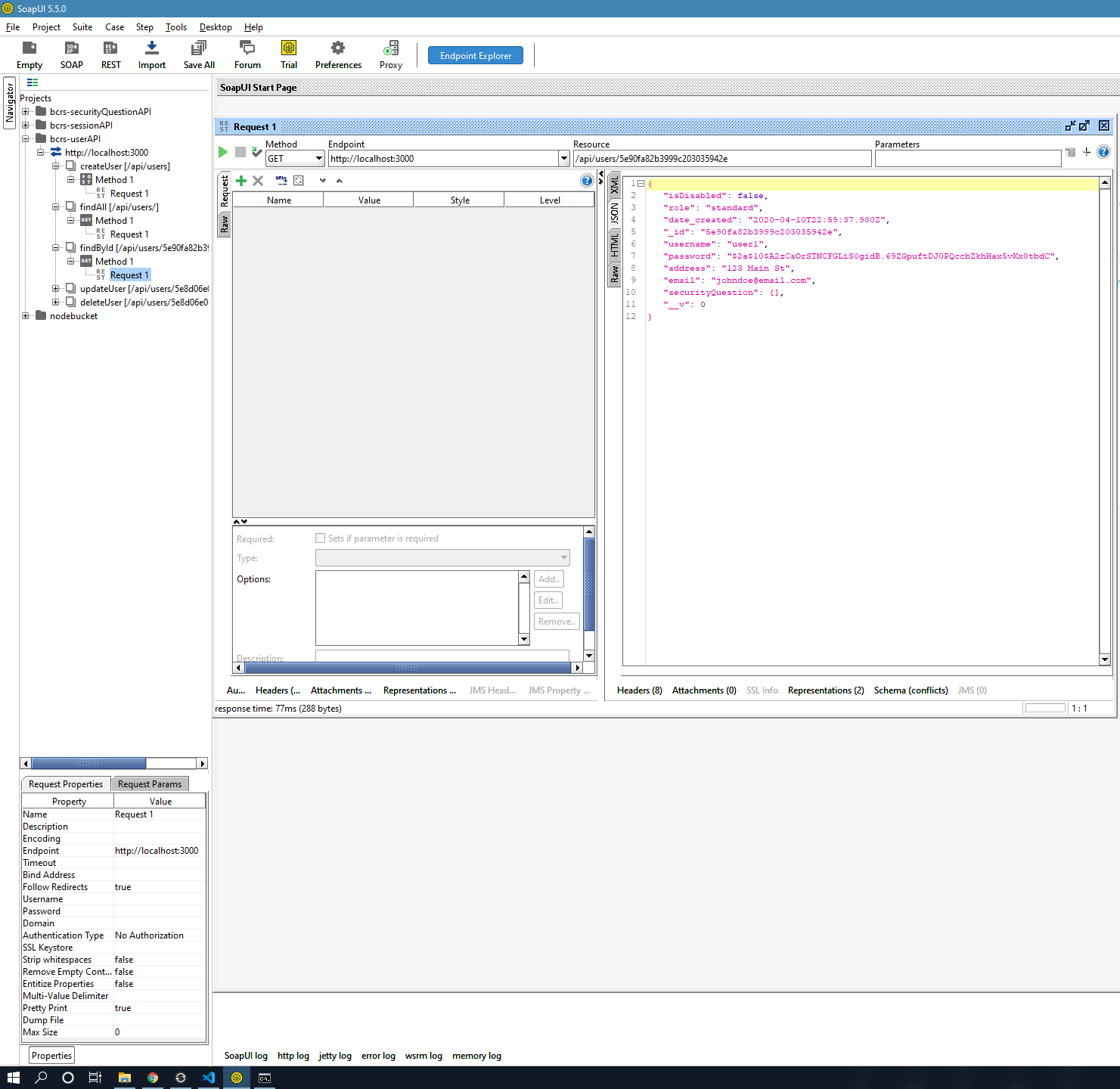
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to  http://localhost:4200/#/signin  Locally | Bob’s Computer Repair appears | Page appeared | Pass | We can login with the credentials. |
| 2 | Login with user info  Username and Password | User can login. | User is logged in | Pass | The three main users can login to the page. |
| 3 | Login redirects to the home page | Home page appears | Page appeared | Pass | It automatically redirects upon successful login |
| 4 | Navigate to http://localhost:4200/#/session/account-registration | Registration page appears | Page appeared | Pass | It has all of the required fields to successfully register a new user |
| 5 | Register a new user | Registered user logs in | New user is logged in | Pass | Once the user is registered it will log them in and route to the home page where the user can choose services |
| 6 | Navigate to  <http://localhost:4200/#/security-questions> | Page of all existing security questions appears | Page appeared | Pass | The user can select 3 questions and create an answer. |
| 7 | Forgot password function | New password is assigned after the user successfully answers the security questions. | New password page appears | Pass | When the user successfully answers all the three security questions, the system allows the user to set a new password. |
| 8 | Check the roles | Admin accounts shows different functions than the standard user | Admin functions appears but not for others. | Pass | The Admin user can create new roles. |
| 9 | Service-repair service | A page with all the services and prices should appear. | Page appeared with all the services available. | Pass | When you select a service and you click submit, a modal page appears with an invoice information. |

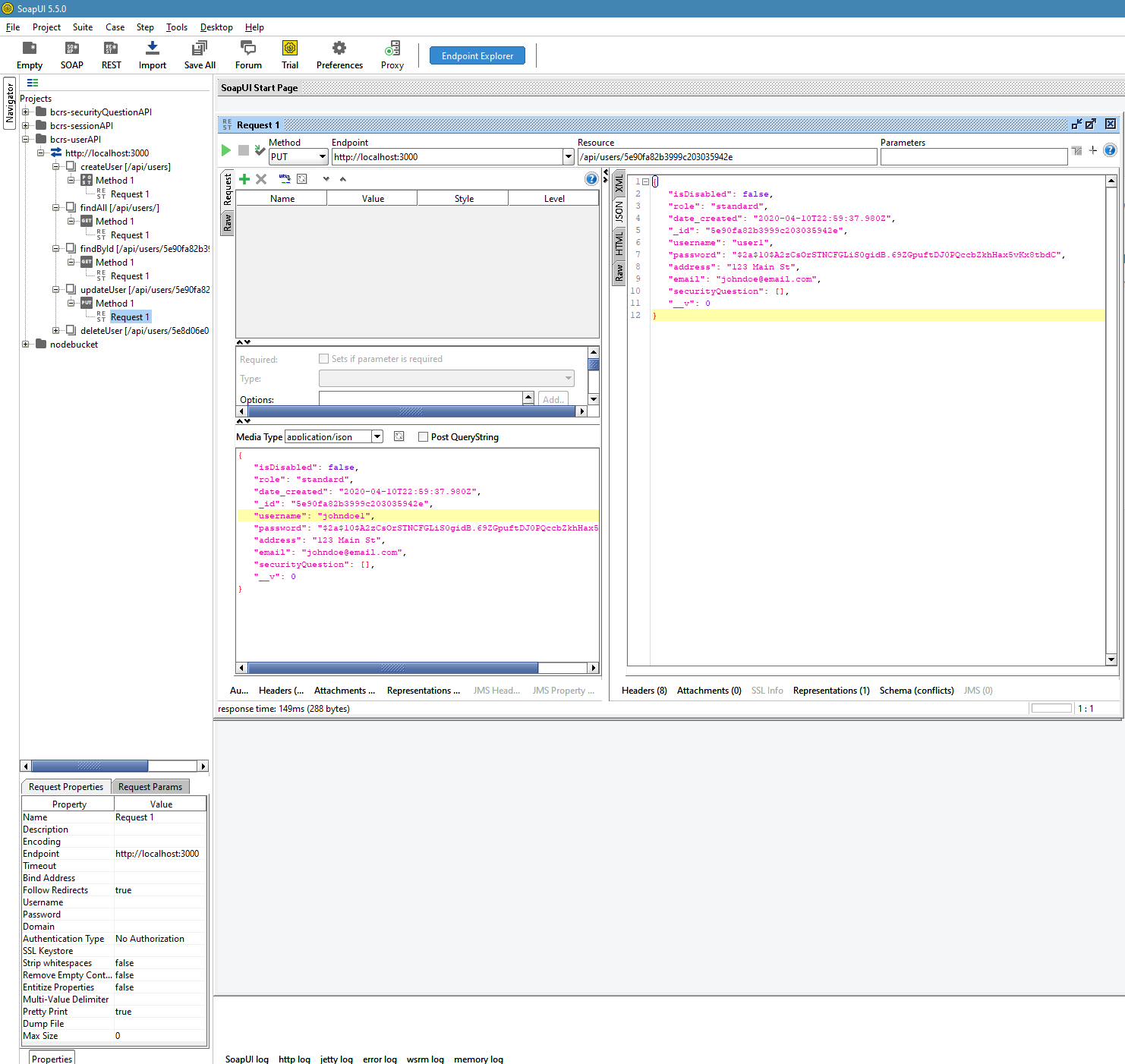
|  |
| --- |
| **Description:** All the tests above were passed, the website Bob’s computer repair is fully working as expected**.** A user or a customer can create an account, create security questions, and rest the password. Admins have more functions available; they can create security questions, get information from the services, and create new roles. |

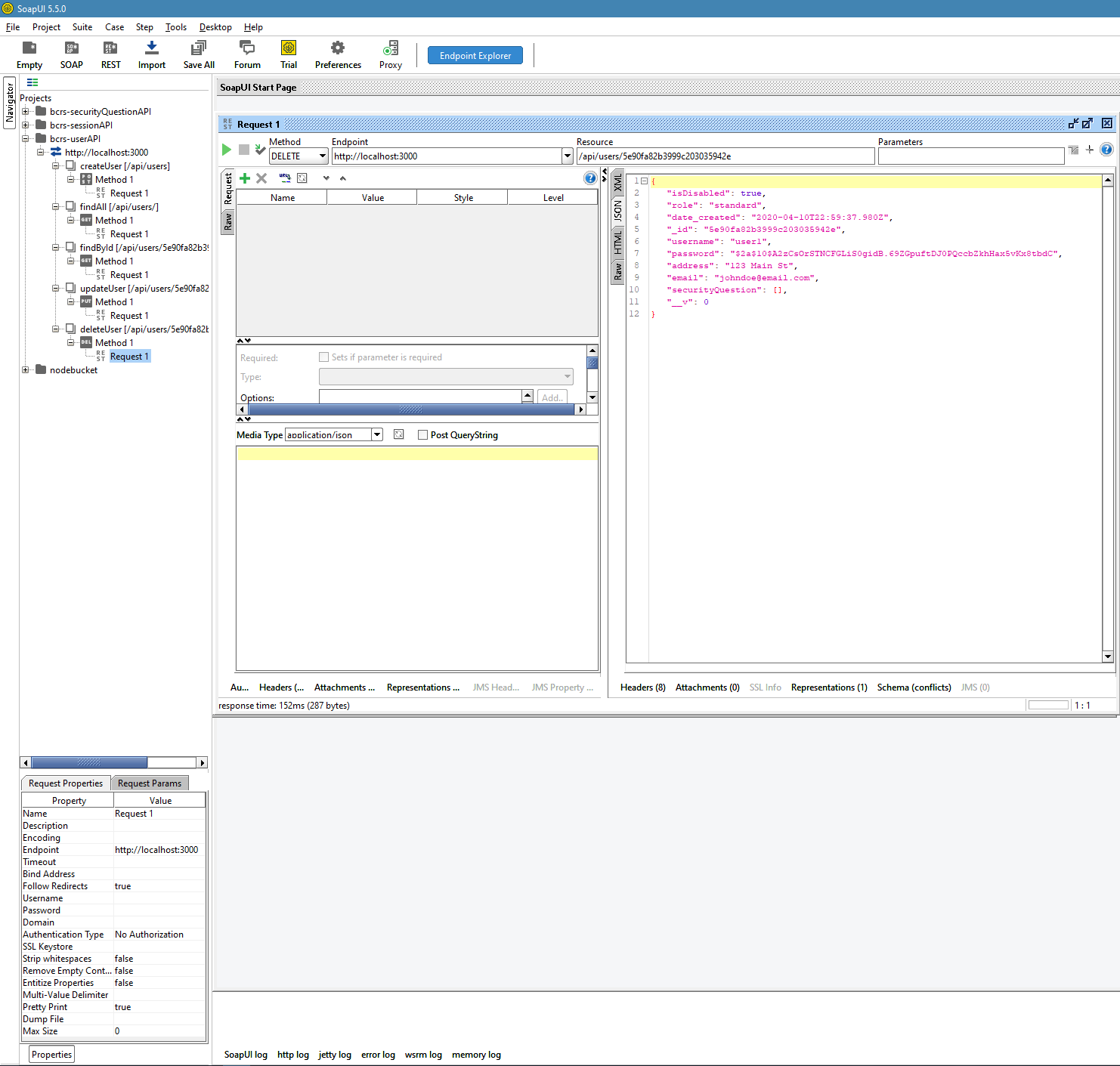
**3.2. Unit Tests**

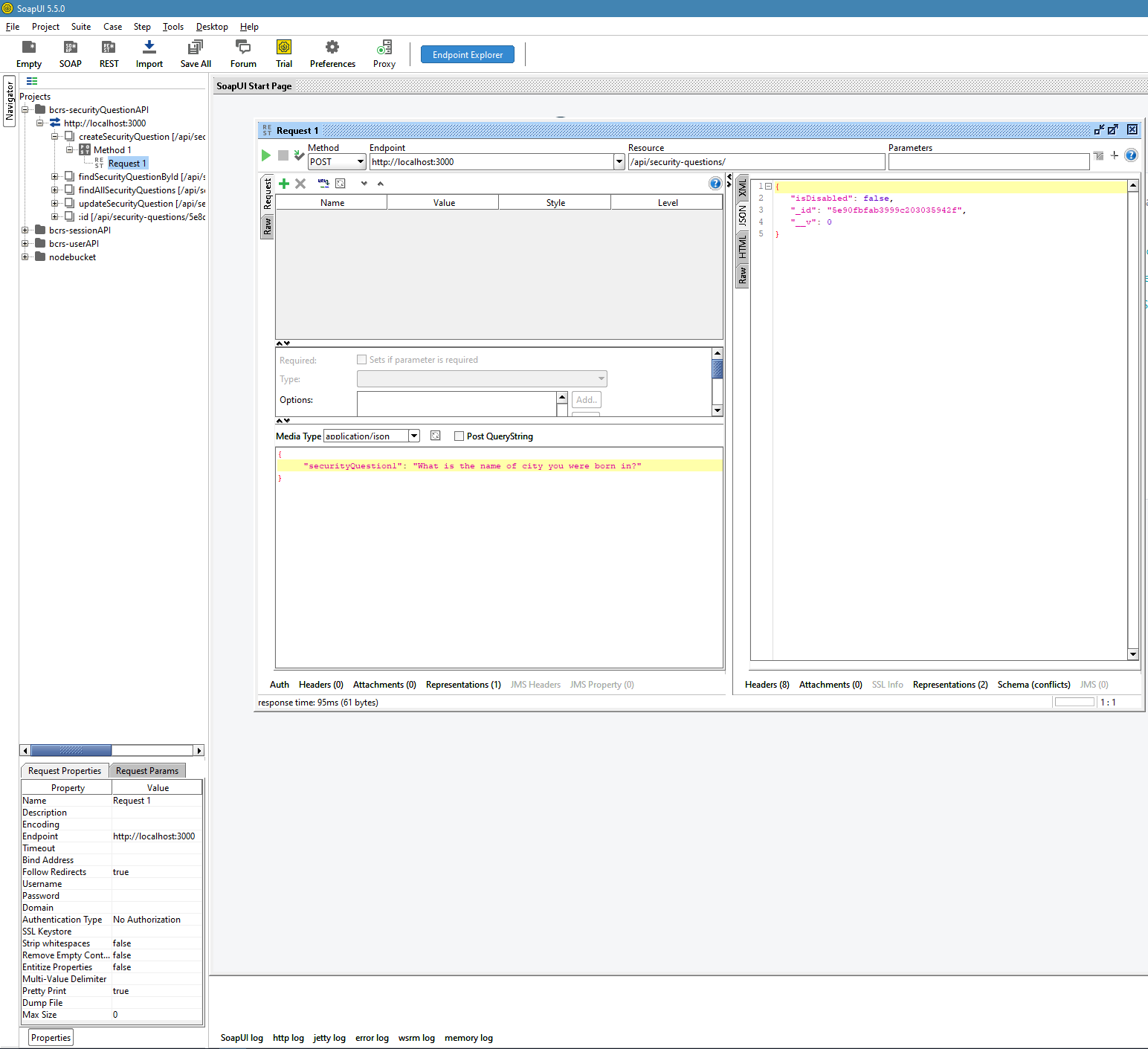


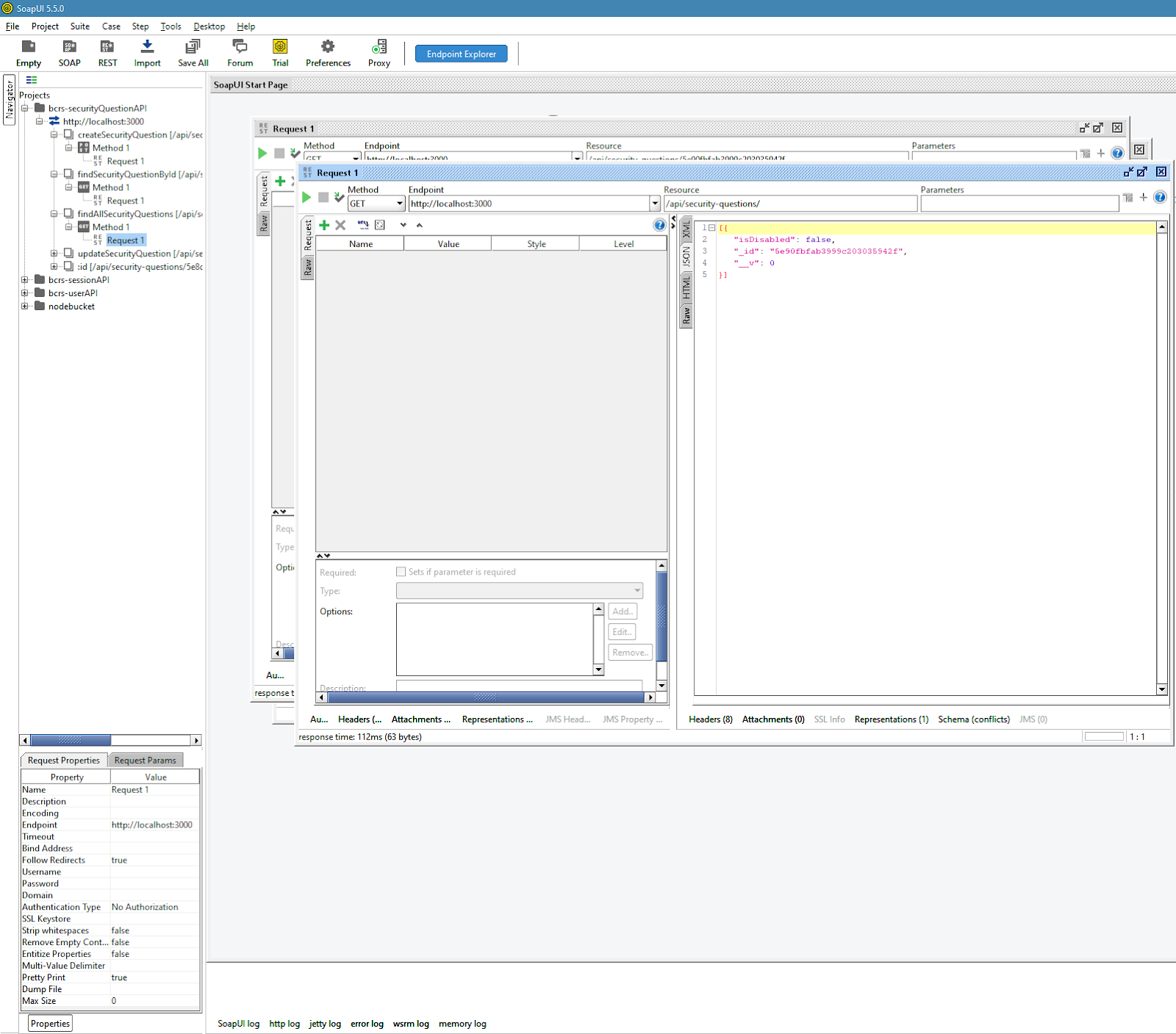


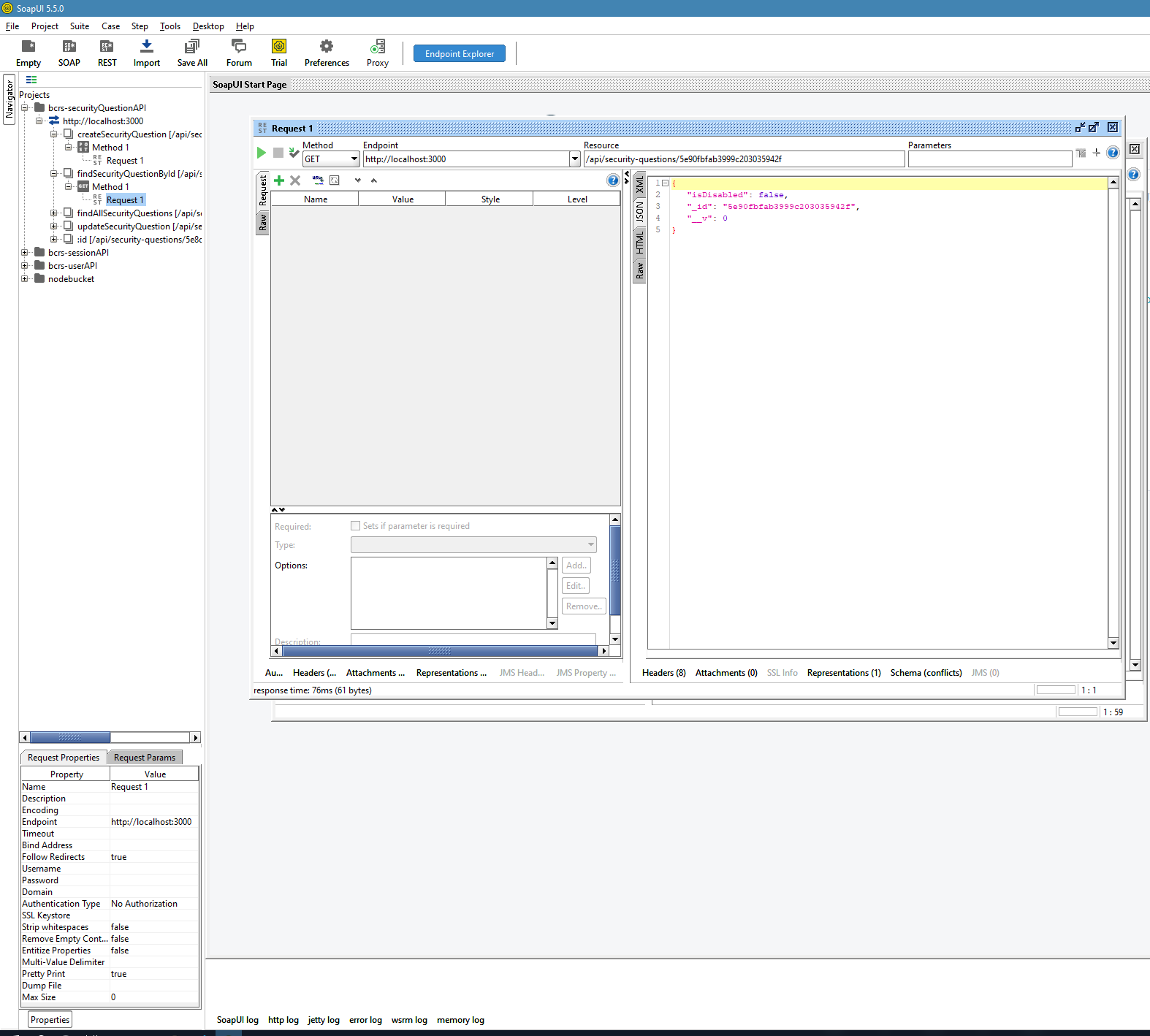


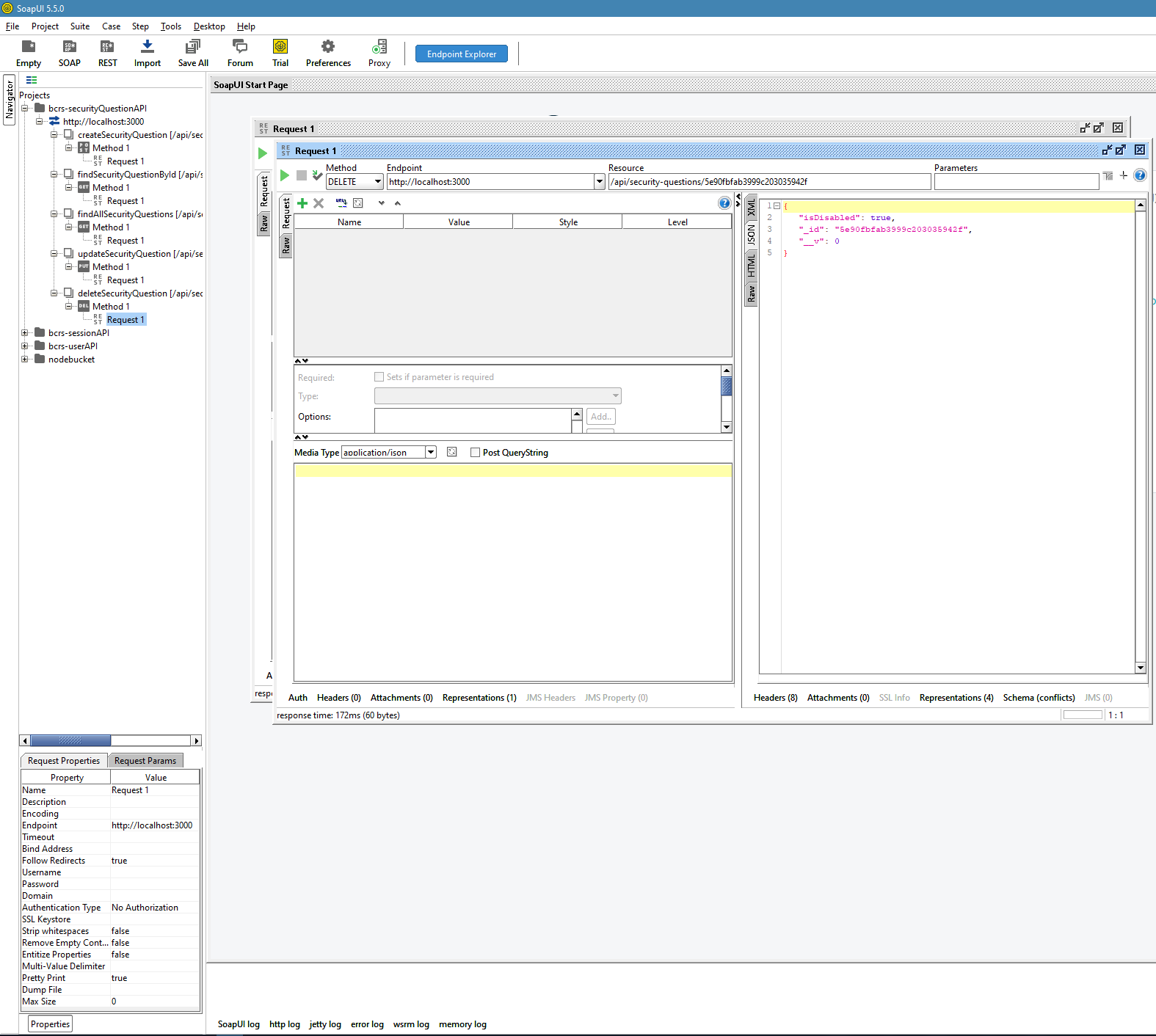


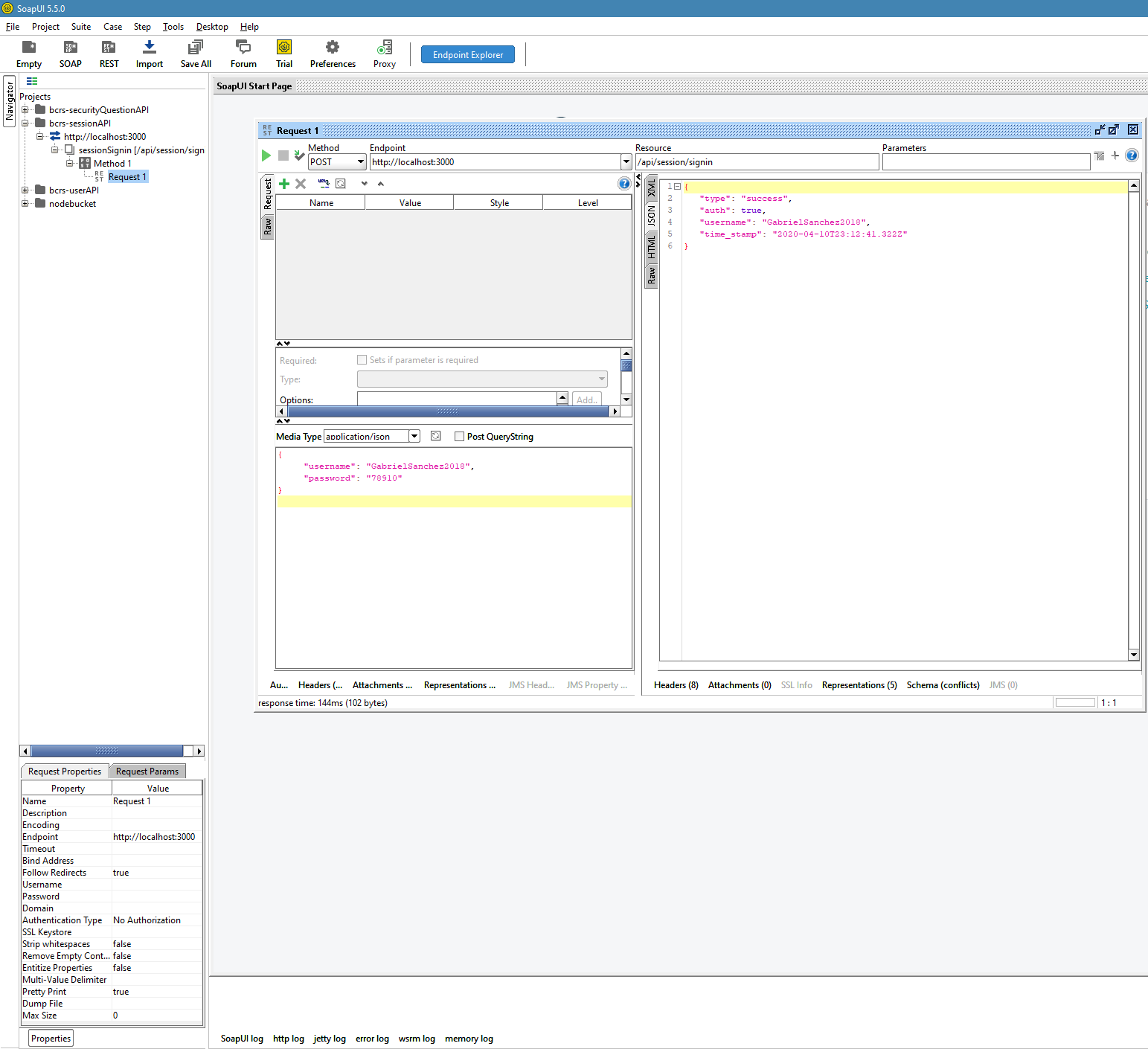


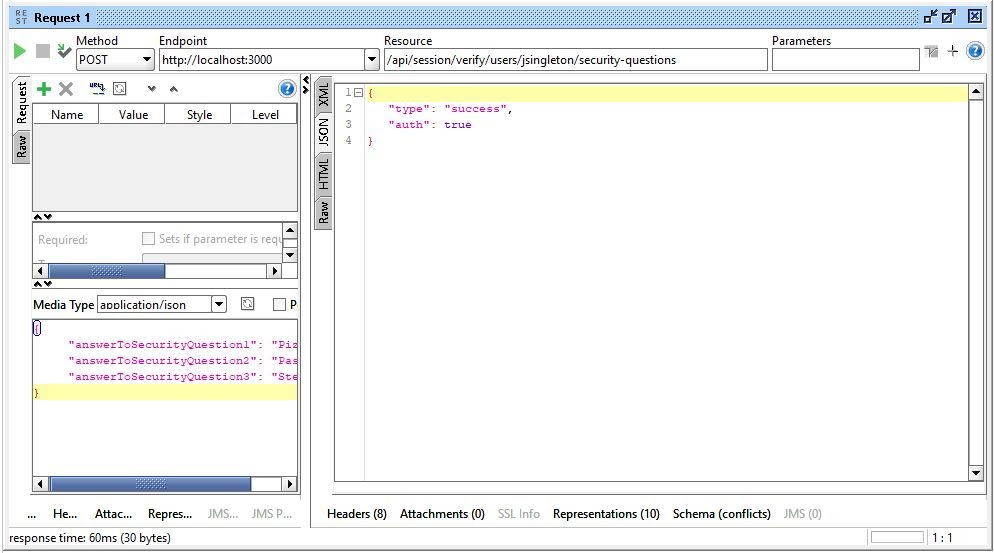
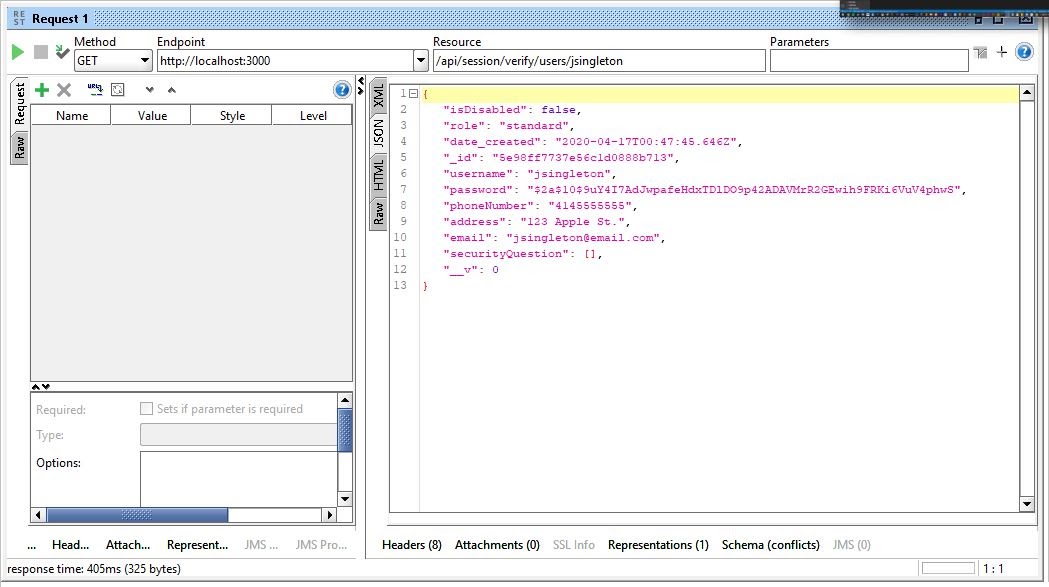
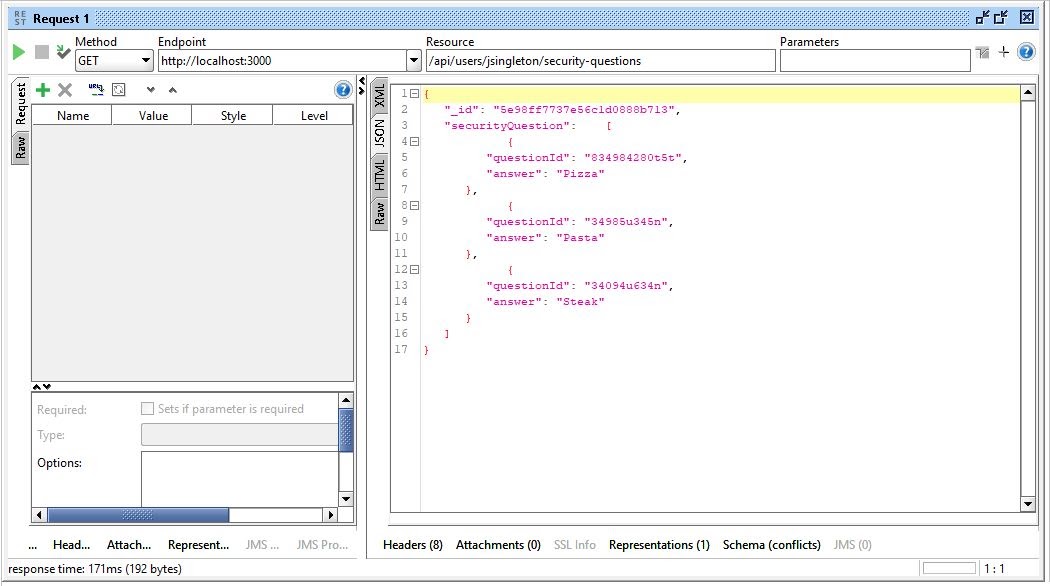
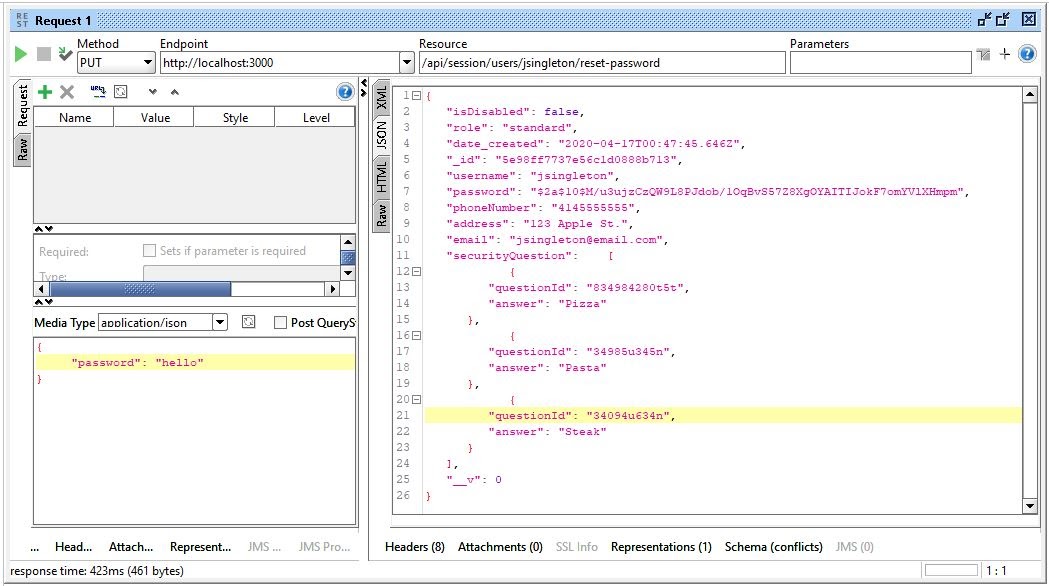
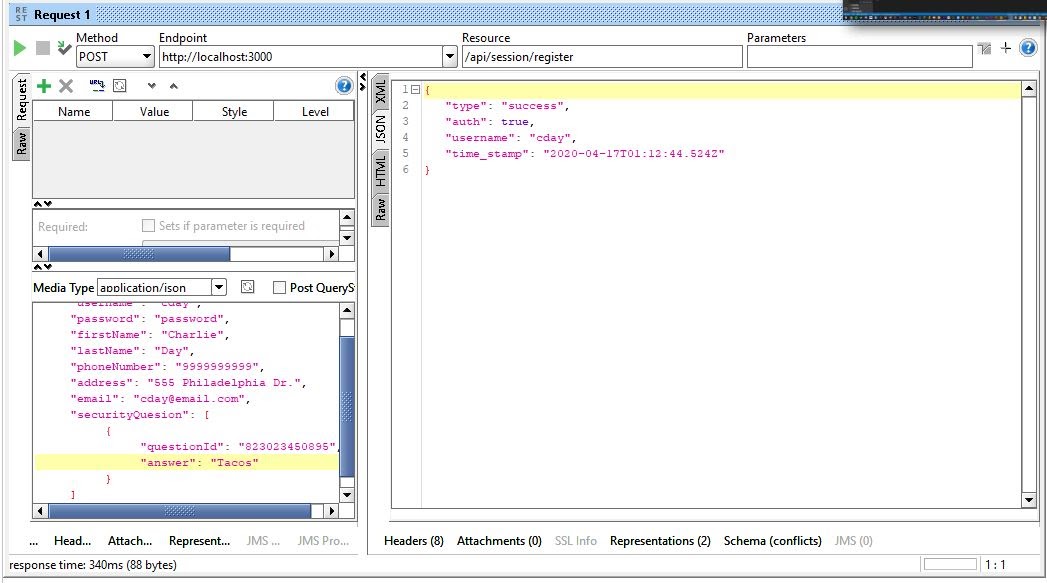
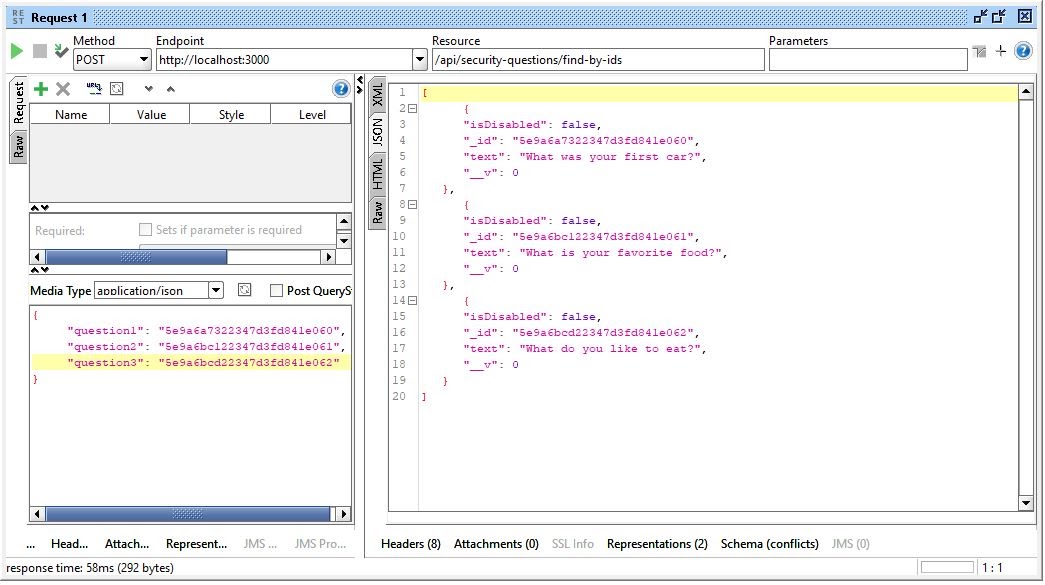












**SECTION 4: REFERENCES**