

QA Fundamentals and Manual Testing Exam Prep I



Part I - Idea Center Web Application

You are provided with a web application, the **Idea Center**, along with a specification of the requirements. The primary objective of the exam is to ensure the App functions as expected. This involves **creating test cases for each use case** provided. It also includes the detection and detailed documentation of any **identified bugs**. You have to document all your work in the provided **Test Management and Bug Tracker Template**.

Software Requirements

1. Introduction

1.1. Purpose

The objective of this document is to provide a description of the Idea Center application (also referred to as Idea Center or The App). It will present an overview of the key functionalities.

1.2. Scope

This document presents high-level descriptions of the basic functionalities of the Idea Center, such as user registration, login, profile editing, idea creation and management. It does not cover any special user (Administrator) functionalities.

2. Overall Description

2.1. System environment

The Idea Center hosts two primary actors: **unregistered/non-logged** users and **registered/logged** users. Both can access their respective parts of the application via the internet on the following url:

<http://softuni-ga-loadbalancer-2137572849.eu-north-1.elb.amazonaws.com:83/>

2.2. Key Features

Unregistered/non-logged users have access only to the Home page of The App, with the option to either **SIGN UP FOR FREE** or **SIGN IN**. On the other hand, **registered/logged** users have access to the **main functionalities**, including **Profile editing**, **Idea creation**, and **Idea management** functionalities.

2.2.1. Home Page for unregistered / non-logged users

The **Home page** is the main gateway into the Idea Center. It contains a Navigation Menu on top (also referred to as **Navbar**) with **"SIGN IN"** and **"SIGN UP FOR FREE"** buttons on the left and back to Home page link on the right. The page also features a **Carousel with three slides** - **"Be part of our community"**, **"Already have an account?"** and **"Enjoy our site!"**. These slides change at random intervals, but users can also manually cycle through them. Two of the slides contain a call-to-action buttons prompting users to **"SIGN UP FOR FREE"** or **"SIGN IN"**.



2.2.2. Sign Up Page

The **Sign Up page** is accessed from the Home page. It contains form fields for entering a **Username**, **Email**, **Password**, and **Repeat password**. The form also includes a **checkbox for agreeing to Terms of service**, a hyperlink to **view the full Terms of service document** and a **Register button to submit the form**. The acceptable requirements for each field are specified as follows:

Username: A 2-30 character field, accepting all character types.

Email: Requires a valid email, 6-254 characters long.

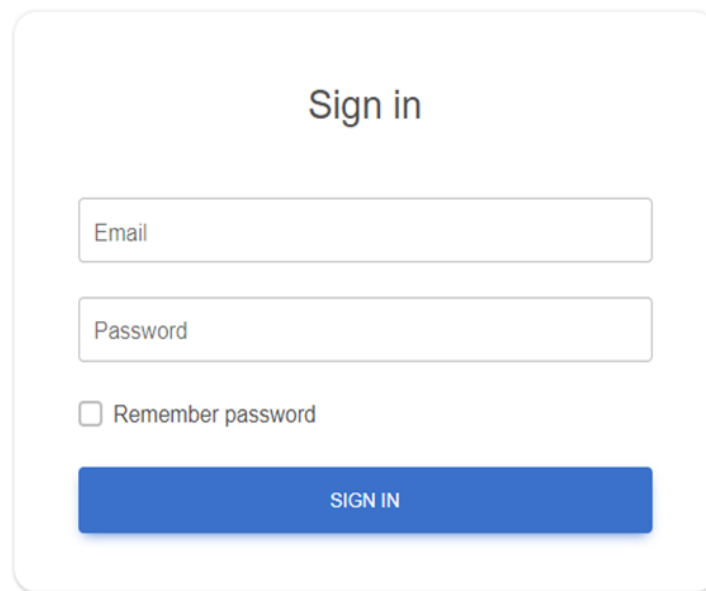
Password: Any characters are acceptable, with a length of 6-30 characters

Repeat Password: Must match password

There is an **alternative sign up options** via **Google or Facebook**.

2.2.3. Sign In Page

The **Sign In** page can be reached from the Home page. The page provides fields for **Email and Password**, as well as a **Remember password** option, and a **Sign In** button.

A sign-in form with a light gray border. At the top, the text "Sign in" is centered. Below it are two input fields: "Email" and "Password". Under the "Password" field is a checkbox labeled "Remember password". At the bottom is a blue button with the text "SIGN IN" in white capital letters.

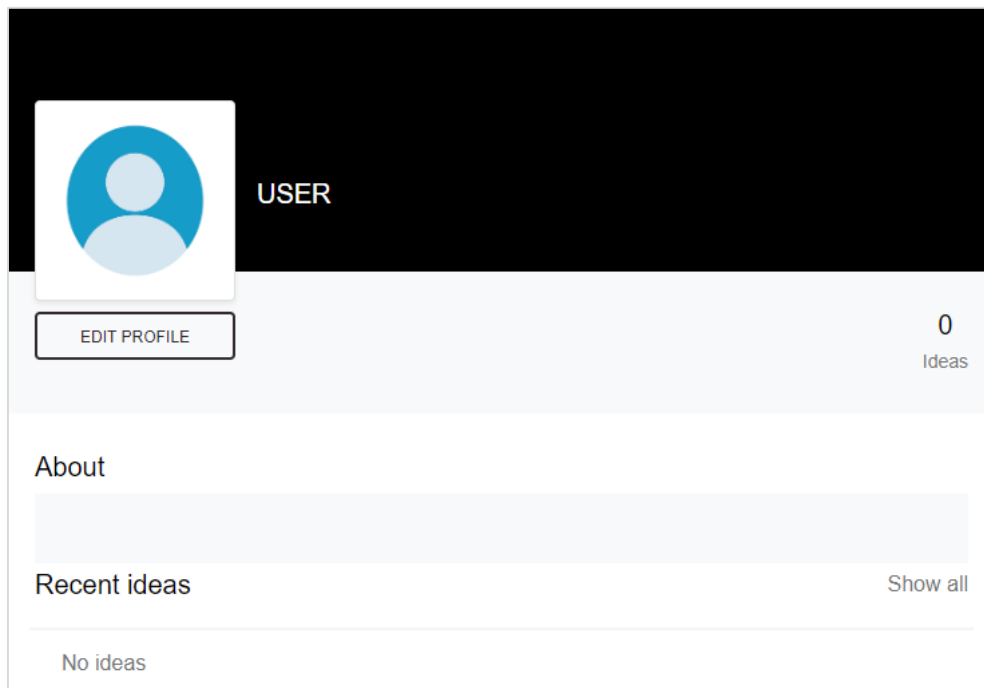
2.2.4. Home Page for logged users

Once a user is **logged in**, they are directed to a different version of **the Home page**. The page includes a **Navbar** with links to **the Home page, My Profile, My Ideas, Create Idea, and a Logout button**. The carousel here includes slides welcoming the user, and offering quick navigation to their profile (**See your profile**) or ideas (**See your ideas**).



2.2.5. My Profile Page

This page is **accessible to logged users** and includes a **default empty profile picture**, an **Edit profile button**, an **empty About section by default**, and a **counter of the user's ideas**, which is initially set to 0. If no ideas have been created yet, the **Recent ideas section shows a No ideas message**. There is also a **Show all link** which leads to **My Ideas page**.

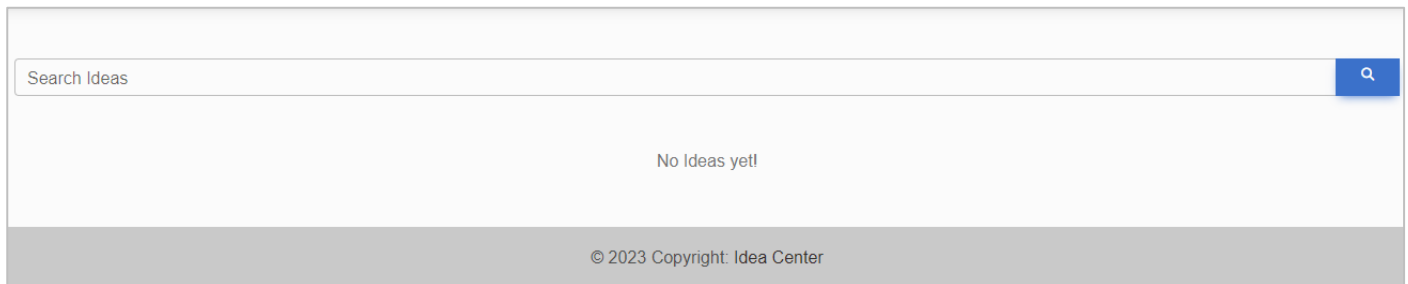


2.2.6. Edit Profile Info Page

Accessible from the **My Profile page**, upon clicking on **EDIT PROFILE button** this page allows **users to edit their profile information**, including **Profile picture** (picture is not uploaded, but must be a **valid URL of a picture**), **First name** (max 60 characters length), **Last name** (max. 60 characters length), **City** (max. 120 characters length), and a **Describe yourself** section (max. 256 characters length).

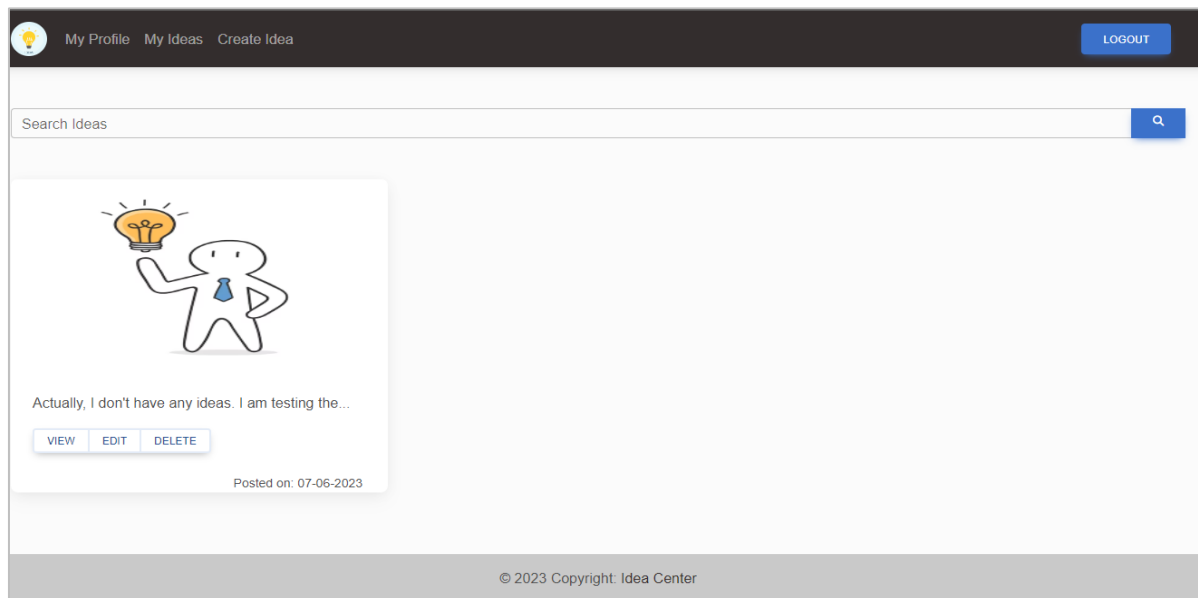
2.2.7. My Ideas Page

This page is accessible **from the Navbar** and contains a **Search ideas field** and a **Search button**. If no ideas have been created yet, a **No ideas yet!** message is displayed. Users can **enter keywords and click the Search button** to initiate the search. The search functionality then filters the list of ideas based on the provided search terms.



The screenshot shows the 'My Ideas' page. At the top, there is a search bar with the placeholder text 'Search Ideas' and a magnifying glass icon. Below the search bar, the text 'No Ideas yet!' is displayed. At the bottom of the page, there is a copyright notice: '© 2023 Copyright: Idea Center'.

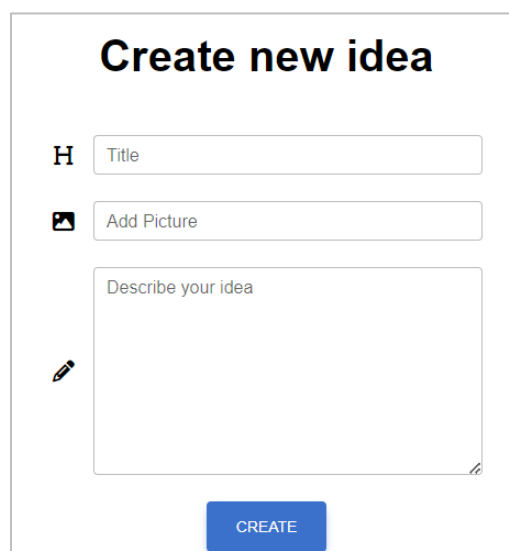
Upon **Creating an idea**, it appears on the **My Ideas page** where the user can **VIEW, EDIT, or DELETE** it.



The screenshot shows the 'My Ideas' page with a single idea card. The card features a lightbulb icon and the text 'Actually, I don't have any ideas. I am testing the...'. Below the text are three buttons: 'VIEW', 'EDIT', and 'DELETE'. The card also shows the date 'Posted on: 07-06-2023'. At the bottom of the page, there is a copyright notice: '© 2023 Copyright: Idea Center'.

2.2.8. Create Idea Page

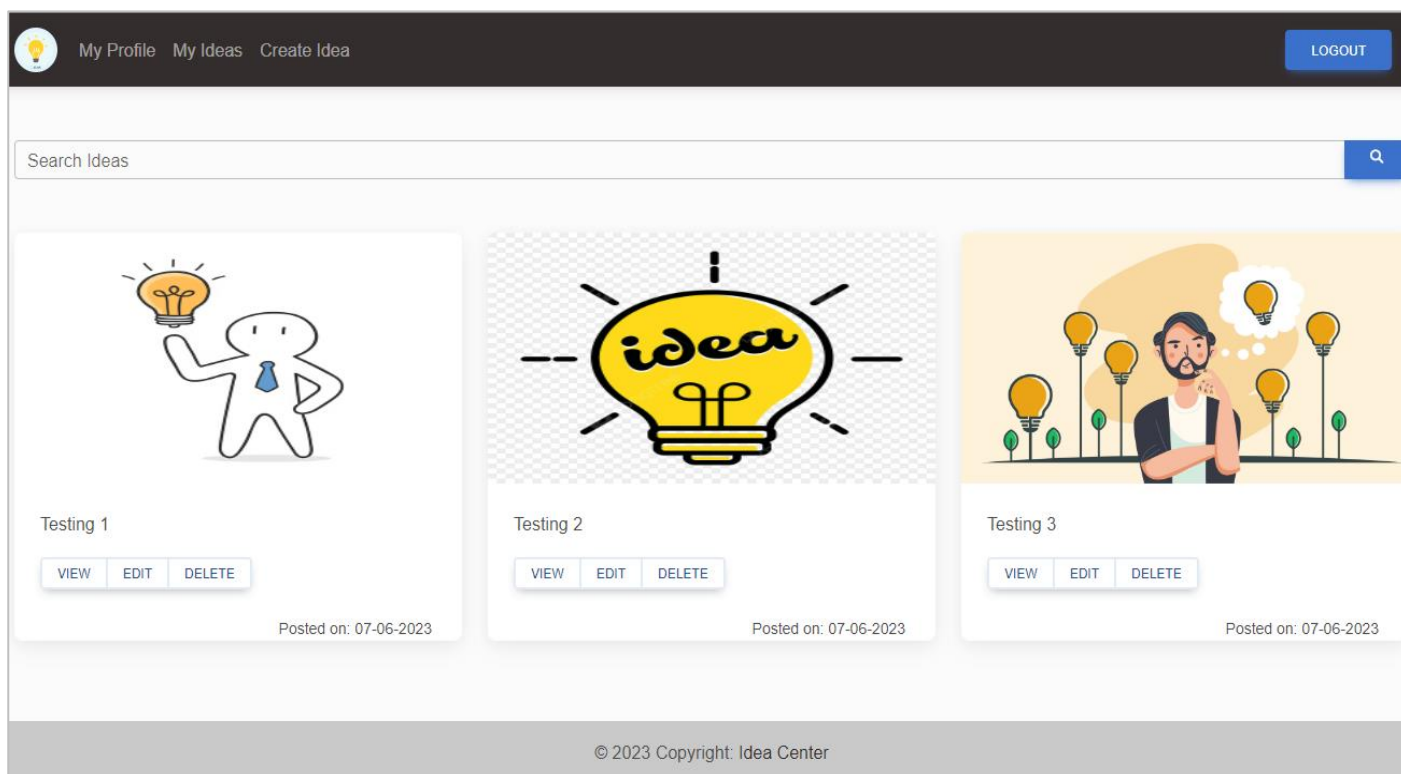
This page can be **accessed from the Navbar** and contains fields for **creating a new idea**, including a **Title** (3-70 characters length), **Add picture field** (picture is not uploaded, but must be a valid URL of a picture), and a **Describe your idea field** (3-400 characters length). Title and Description are mandatory. After entering these details, the user can click on the **CREATE button** to save the idea.



The screenshot shows the 'Create new idea' form. It has three main input fields: 'Title', 'Add Picture', and 'Describe your idea'. The 'Describe your idea' field is a larger text area. Below the text area is a 'CREATE' button. The form is titled 'Create new idea'.

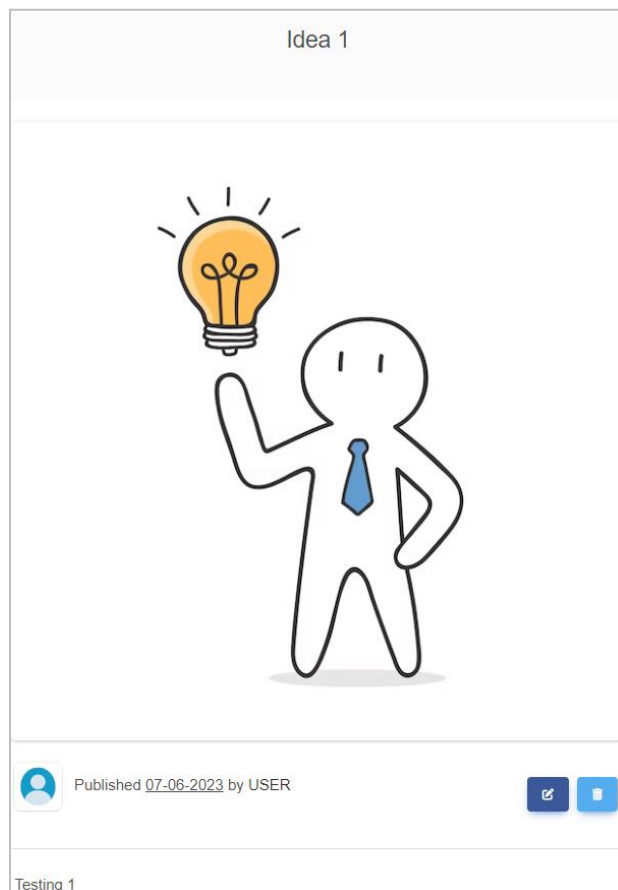
2.2.9. Idea Management

Managing Ideas is **available** for all created ideas from **My Ideas Page**. Each Idea has three options – **VIEW**, **EDIT**, and **DELETE**.



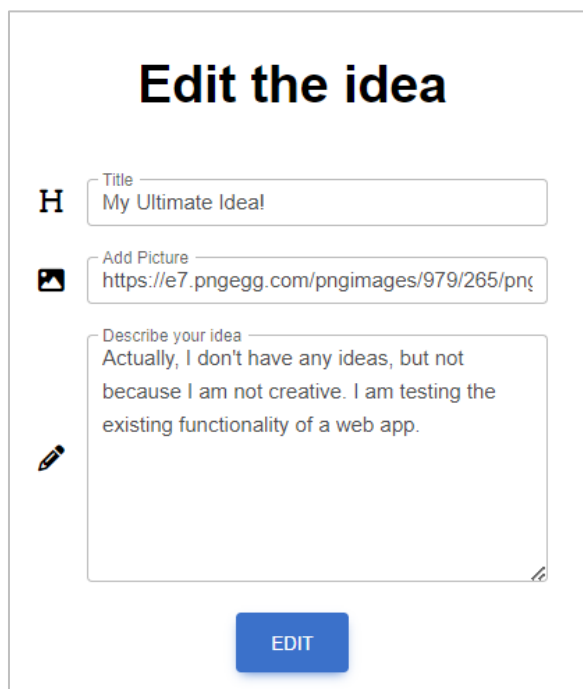
a. View

Available after clicking on **VIEW** button.




b. Edit


Available after clicking on **EDIT** button.



Edit the idea

H Title

 Add Picture

 Describe your idea

EDIT

c. Delete

After clicking on **DELETE** button, the Idea is deleted and it **disappears** from **My Ideas Page**.

Functional Requirements

1. Use Case 1 (Home Page)

Users can access The Idea Center **from its designated URL via the internet**, which will load the Home page, Carousel, and Navigation Pane, appropriate to the user's logged-in status (unregistered/non-logged or registered/logged).

2. Use Case 2 (User Registration)

Unregistered users should be able to successfully go through **the Sign Up process**. This involves the utilization of fields such as Username, Email, and Password, all subject to their respective constraints and character type acceptance. User should have the **option to Sign Up via Facebook or Google** if he prefers.

3. Use Case 3 (User Sign In)

Registered users should be able to successfully go through the **Sign In process**. They should be able to sign in using Email and Password.

4. Use Case 4 (Profile Management)

Upon successful Sign In, **logged users** should be able to navigate to **My Profile page**, edit their Profile details, and **view their Ideas count**. This involves the **changing of profile picture** inserted via URL, and **editing of user data** fields including **First Name, Last Name, City, and Describe Yourself**.

5. Use Case 5 (Idea Creation)

Logged users should be able to navigate to the **Create Idea page**, where they can create a **New idea with a Title, Picture (URL), and Description**. After the idea creation process, users should be redirected to the My Ideas page, where the newly created idea should be present.

6. Use Case 6 (Idea Management)

On the **My Ideas page**, logged users should see **all their ideas listed**. Each idea should have options for **VIEW, EDIT, and DELETE**. If no ideas have been created yet, a **No ideas yet! message** should be displayed. Users also should have the option to search for ideas, based on keywords.

Idea Center Web App Tasks

1. Test Cases

You need to write **test cases for each of the 6 Use cases**.

Each Use case should have at **least 2 Test Cases**.

You should write **at least 15 test cases**.

2. Bug Reports

You should find and describe at least **5 bugs**.

3. Template

Use the provided **Test-Management-and-Bug-Tracking-Template.xlsx** to document your work.

Part II - Idea Center API

This section provides information about the **Idea Center API endpoints** and their functionalities. It serves as a guide to understand how to interact with the Idea Center API.

Methods supported

You can see all the supported methods on the following URL:

<http://softuni-qa-loadbalancer-2137572849.eu-north-1.elb.amazonaws.com:84/api/Info/Methods>

Endpoints

The following endpoints are supported:

1. User

- **POST /api/User/Create** – create a new user: post a JSON object in the request body, e.g.
{ "userName": "string", "email": "user@example.com", "password": "string", "rePassword": "string", "acceptedAgreement": true }


- POST /api/User/Authentication:** log in an existing user (post a JSON object in the request body, e.g. `{ "email": "user@example.com", "password": "string"}`)

Access Token

- When a user logs in, the response format is JSON object, e.g.
- ```
{"email": "test@gmail.com", "password": "1234567", "accessToken":
"eyJhbGciOiI..."}

```

**NB! Access token is needed for all idea requests. It should be placed under the Authorization tab, Bearer Token option.**



The screenshot shows the Postman interface for a POST request to `https://localhost:7114/api/Idea/Create`. The 'Authorization' tab is selected, and the 'Type' is set to 'Bearer Token'. A warning message is displayed: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)'. The 'Token' field contains the value `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`.

## 2. Idea

All of the **following** requests require Authotization.

- GET /api/Idea/All** – list all ideas (empty request body)
- POST /api/Idea/Create** – create a new idea: post a JSON object in the request body, e.g.
- ```
{"title": "string", "description": "string"}
```
- PUT /api/Idea/Edit** – replace the existing idea with the new one: post a JSON object in the request body, e.g. `{"title": "string", "url": " ", "description": "string"}`;
- requires a **Query Parameter: ideald**
- DELETE /api/Idea/Delete** – delete existing idea; requires a **Query Parameter: ideald**

Idea Center API Tasks: Postman Requests

Your task is to write **API requests** with Postman for certain RESTful API endpoints.

You should organize your requests in a collection.

1. Log in to the API

Send a POST request with your username and password, that you previously created from the Idea Center Web App. If properly executed, you will receive your access token as a part of the response body. Use it as a Bearer Token in the Authorization section for the next requests.

Example Request:

The screenshot shows the IntelliJ IDEA REST Client interface. At the top, there's a header bar with the IntelliJ logo, 'Idea Center / Log In and get Token', and buttons for 'Save' and 'Send'. Below the header, a dropdown menu is set to 'POST' and the URL 'https://localhost:7114/api/User/Authentication' is entered. The 'Body' tab is selected, showing a JSON payload:

```
{
  "email": "testuser@example.com",
  "password": "string1"
}
```

 The 'Headers' tab shows 8 headers. The 'Authorization' tab is also visible. The 'Send' button is highlighted in blue.

Example Response:

[illegible]

2. Create a new idea

Send a POST request with the attributes needed for creating a new idea

Keep in mind that the full attributes in the JSON body are as follows:


`{"title": "string", "url": "http://...jpg ", "description": "string"}],` but as you can see from the example requests, the url attribute is not mandatory, meaning that you can create an idea only with title and description. If you decide to use the url attribute, notice, that it has to point to a picture format or it can be left blank.

Example Requests:

Whithout url

A screenshot of the Postman REST client interface. At the top, the breadcrumb navigation shows 'Idea Center / Create New Idea'. On the right, there are 'Save' and 'Send' buttons. The main area shows a POST request to 'https://localhost:7114/api/Idea/Create'. Below the URL bar, tabs for 'Params', 'Authorization', 'Headers (9)', 'Body', 'Pre-request Script', 'Tests', and 'Settings' are visible, with 'Body' being the active tab. Under the 'Body' tab, there are radio buttons for different formats: 'none', 'form-data', 'x-www-form-urlencoded', 'raw' (which is selected and highlighted with a red dot), 'binary', and 'GraphQL'. To the right of these tabs are links for 'Cookies' and 'Beautifuly'. The request body is displayed in a text area with line numbers 1 through 4, containing the JSON: {"title": "New Idea 1", "description": "New Idea Description"}. A small 'JSON' icon is visible in the top left of the text area.

With url



The screenshot shows a REST client interface. The top bar indicates the URL is `https://localhost:7114/api/Idea/Create` and the method is `POST`. The request body is in `JSON` format. The body content is:

```
{
  "title": "GREAT IDEA122",
  "url": "http://.gif",
  "description": "New Idea Description"
}
```

A red box highlights the `"url": "http://.gif"` line, and a red arrow points to it with the following text:

starts with `http://...`
ends with picture format, e.g. `.jpg`, `.svg`, `.bmp`, `.gif`, `.png`.

Empty url

[HTTP](#) Idea Center / Create New Idea

Save

POST ▼ https://localhost:7114/api/Idea/Create Send ▼

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON ▼

Cookies

Beautify

```
1 {
2   "title": "GREAT IDEA122",
3   "url": "",
4   "description": "New Idea Description"
5 }
```

Example Response:

Without url

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 66 ms Size: 257 B Save as Example

Pretty Raw Preview Visualize JSON ▼

```
1 {
2   "msg": "Successfully created!",
3   "idea": {
4     "title": "New Idea 1",
5     "url": null,
6     "description": "New Idea Description"
7   }
8 }
```

With url

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 60 ms Size: 269 B Save as Example

Pretty Raw Preview Visualize JSON ▼

```
1 {
2   "msg": "Successfully created!",
3   "idea": {
4     "title": "GREAT IDEA14",
5     "url": "http://.gif",
6     "description": "New Idea Description"
7   }
8 }
```

Empty url

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 20 ms Size: 257 B Save as Example

Pretty Raw Preview Visualize JSON ▼

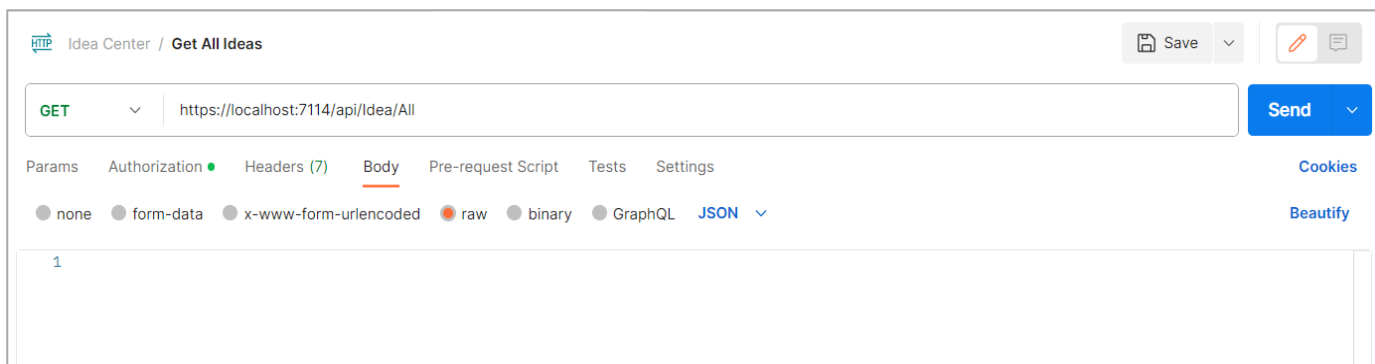
```
1 {
2   "msg": "Successfully created!",
3   "idea": {
4     "title": "GREAT IDEA17",
5     "url": "",
6     "description": "New Idea Description"
7   }
8 }
```

3. List all ideas

Send a GET request to receive a list of all created ideas.

This request will return all ideas created through your profile. So, if you created any ideas via the Web App, they also will be listed. In the list of ideas, find the idea that you just created. You will **need its id** for the next request.

Example Request:



HTTP Idea Center / Get All Ideas

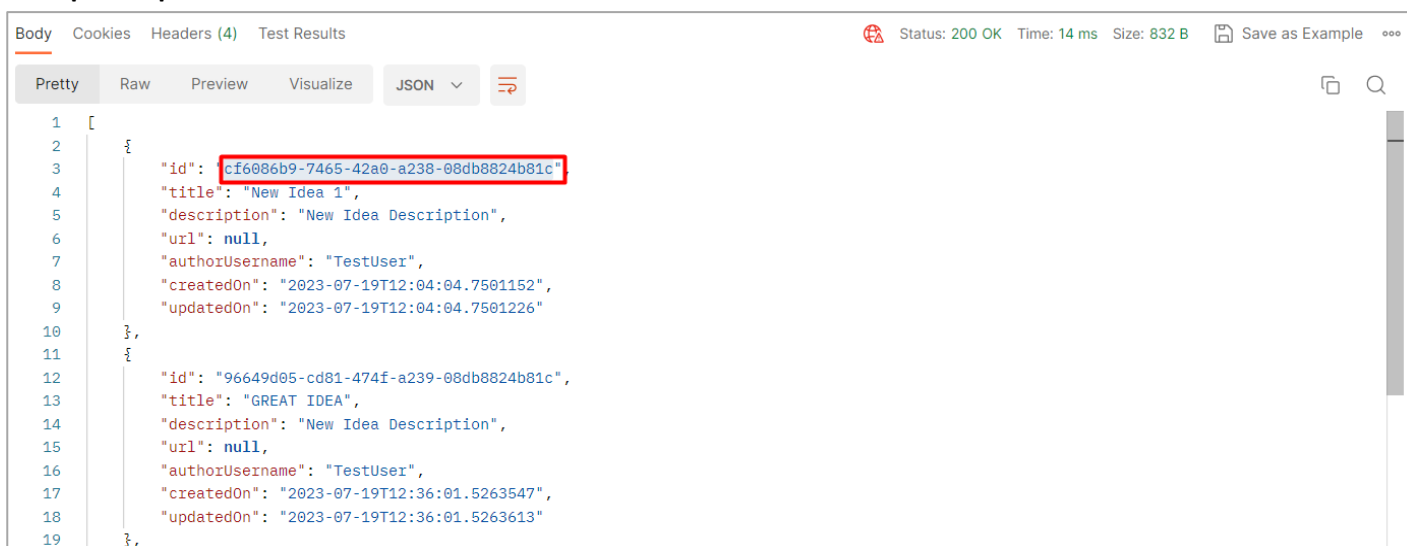
GET https://localhost:7114/api/Idea/All

Params Authorization Headers (7) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1

Example Response:



Body Cookies Headers (4) Test Results

Status: 200 OK Time: 14 ms Size: 832 B Save as Example

Pretty Raw Preview Visualize JSON

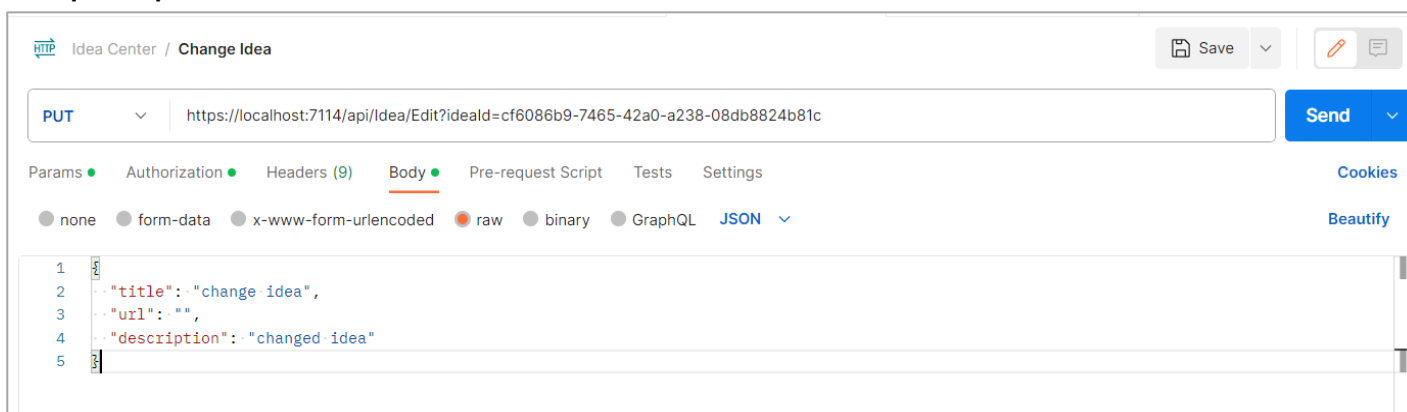
```
1 [
2   {
3     "id": "cf6086b9-7465-42a0-a238-08db8824b81c",
4     "title": "New Idea 1",
5     "description": "New Idea Description",
6     "url": null,
7     "authorUsername": "TestUser",
8     "createdOn": "2023-07-19T12:04:04.7501152",
9     "updatedOn": "2023-07-19T12:04:04.7501226"
10  },
11  {
12    "id": "96649d05-cd81-474f-a239-08db8824b81c",
13    "title": "GREAT IDEA",
14    "description": "New Idea Description",
15    "url": null,
16    "authorUsername": "TestUser",
17    "createdOn": "2023-07-19T12:36:01.5263547",
18    "updatedOn": "2023-07-19T12:36:01.5263613"
19  },
20 ]
```

4. Edit Idea

Send a PUT request with the attributes needed for replacing the content of the idea that you created

You will need to send your request to `/api/Idea/Edit?ideald=cf6086b9-7465-42a0-a238-08db8824b81c`, the number in bold being the idea id. You can get the idea id from the previous request.

Example Request:



HTTP Idea Center / Change Idea

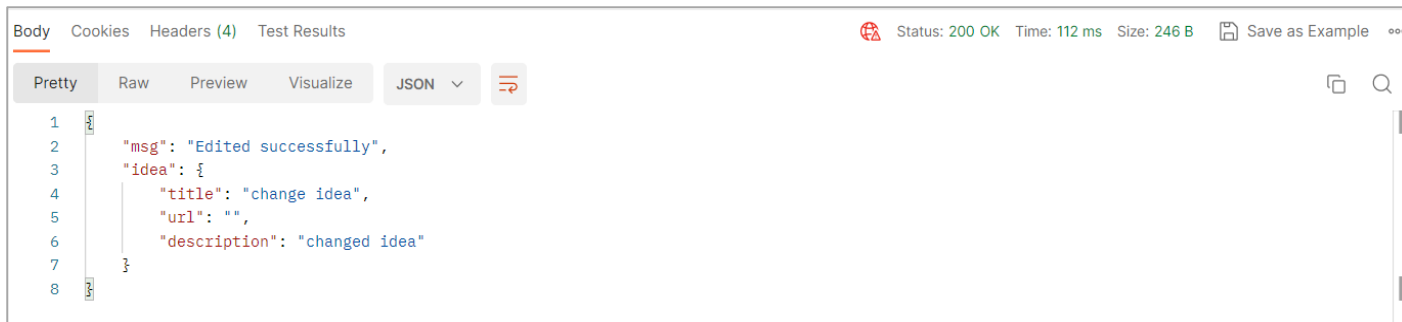
PUT https://localhost:7114/api/Idea/Edit?ideald=cf6086b9-7465-42a0-a238-08db8824b81c

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "title": "change idea",
3   "url": "",
4   "description": "changed idea"
5 }
```

Example Response:



The screenshot shows a REST client interface with the 'Body' tab selected. The response is a JSON object indicating a successful deletion. The status is 200 OK, the time is 112 ms, and the size is 246 B. The JSON body is as follows:

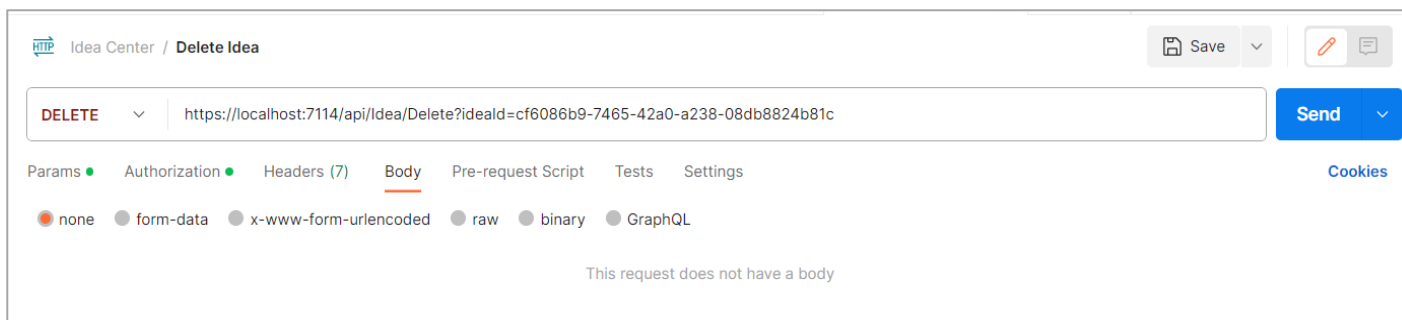
```
1 {
2   "msg": "Edited successfully",
3   "idea": {
4     "title": "change idea",
5     "url": "",
6     "description": "changed idea"
7   }
8 }
```

5. Delete Idea

Send a DELETE request with the attributes needed for deleting the idea you just replaced

You will need to send your request to `/api/Idea/Delete?ideald=cf6086b9-7465-42a0-a238-08db8824b81c`, the number being the idea id. You can get the idea id from the list all ideas request.

Example Request:



The screenshot shows a REST client interface for a DELETE request. The URL is `https://localhost:7114/api/Idea/Delete?ideald=cf6086b9-7465-42a0-a238-08db8824b81c`. The request type is set to DELETE. The 'Body' tab is selected, and it shows 'This request does not have a body'. The status bar indicates a successful response (200 OK) with a time of 120 ms and a size of 162 B.

Example Response:



The screenshot shows a REST client interface with the 'Body' tab selected. The response is a plain text message indicating the idea has been deleted. The status is 200 OK, the time is 120 ms, and the size is 162 B. The text body is as follows:

```
1 The idea is deleted!
```

How to submit your work

1. Create a .zip or .rar archive containing the following:

- Test-Management-and-Bug-Tracker.xlsx file with your work;
- JSON file containing your collection of requests, created via Postman.

2. Upload your archive on the SoftUni website, into Regular Exam section.

Note: It's the same as uploading your homework.