API Specification for "The Republic"

This document provides a detailed specification for the API endpoints and functionality to be implemented for "The Republic," a Progressive Web App aimed at enabling users to report and discuss government service delivery issues.

Base URL

To run the server locally:

PROF

```
cd backend
npm i
npm run dev
```

The server will be accessible at:

```
http://localhost:8080
```

Once the complete backend is available, it will be deployed and a link to that will be added to this document.

Authentication

Some endpoints require authentication using a bearer token. Include the token in the `Authorization` header:

```
Authorization: Bearer <token>
Content-type: application/json
```

If the token is missing for an endpoint where it is required the API will respond with `401 Unauthorized`. If it is invalid or expired the response will be `403 Forbidden`.

Endpoints

> Note:`?` after a parameter name means it is optional

1. Create a New Issue

- **Method:** `POST`
- **Endpoint:** `/api/issues/create`
- **Description:** Create a new issue.
- **Authentication:** Required
- **Request Body:**

  ```
  {
    "location_id": "number",
    "category_id": "number",
    "content": "string",
    "image_url?": "string",
    "is_anonymous": "boolean",
    "sentiment": "string"
  }
  ```

PROF

- **Response:**
  - **201 Created** (`data` will contain the created issue)

    ```
    {
      "success": true,
      "code": 200,
      "data": {}
    }
    ```

  - **400 Bad Request** (Invalid input data)

    ```
    {
      "success": false,
      "code": 400,
    ```

```
  "error": "Missing required fields for creating an issue"
}
```

- **401 Unauthorized**

```
{
  "success": false,
  "code": 401,
  "error": "You need to be signed in to create an issue."
}
```

- **500 Internal Server Error**

```
{
  "success": false,
  "code": 500,
  "error": "An unexpected error occurred. Please try again
later."
}
```

2. Get Issues

- **Method:** `POST`
- **Endpoint:** `/api/issues`
- **Description:** Get a range of issues.
- **Request Body:**

```
{
  "from": "number",
  "amount": "number",
  "order_by?": "string",
  "ascending?": "boolean",
  "category?": "string",
  "mood?": "string",
  "user_id?": "string"
}
```

> Issues are sorted by latest first by default

- **Response:**
  - **200 OK**

PROF

```
{
  "success": true,
  "code": 200,
  "data": [
    {
      "issue_id": "number",
      "user_id": "string",
      "location_id": "number",
      "category_id": "number",
      "content": "string",
      "image_url": "string",
      "is_anonymous": "boolean",
      "created_at": "string",
      "resolved_at": "string",
      "sentiment": "string",
      "user": {
        "user_id": "string",
        "email_address": "string",
        "username": "string",
        "fullname": "string",
        "image_url": "string"
      },
      "category": {
        "name": "string"
      },
      "reactions": [
        {
          "emoji": "string",
          "count": "number"
        }
      ],
      "user_reaction": "string",
      "comment_count": "number",
      "is_owner": "boolean"
    }
  ]
}
```

PROF

3. Get Issue by ID

- **Method:** `POST`
- **Endpoint:** `/api/issues/single`
- **Description:** Retrieve an issue by its ID.
- **Request Body:**

```
{
  "issue_id": "number"
}
```

- **Response:**
    - **200 OK**

```
{
  "success": true,
  "code": 200,
  "data": {
    "issue_id": "number",
    "user_id": "string",
    "location_id": "number",
    "category_id": "number",
    "content": "string",
    "image_url": "string",
    "is_anonymous": "boolean",
    "created_at": "string",
    "resolved_at": "string",
    "sentiment": "string",
    "user": {
      "user_id": "string",
      "email_address": "string",
      "username": "string",
      "fullname": "string",
      "image_url": "string"
    },
    "category": {
      "name": "string"
    },
    "reactions": [
      {
        "emoji": "string",
        "count": "number",
        "user_reacted": "boolean"
      }
    ],
    "user_reaction": "string",
    "comment_count": "number",
    "is_owner": "boolean",
    "is_subscribed": "boolean"
  }
}
```

PROF

    - **404 Not Found** (Issue not found)

```
{
  "success": false,
  "code": 404,
  "error": "Issue does not exist."
}
```

- **500 Internal Server Error**

  ```
  {
    "success": false,
    "code": 500,
    "error": "An unexpected error occurred. Please try again
  later."
  }
  ```

4. Resolve an Issue

- **Method:** `PUT`
- **Endpoint:** `/api/issues/resolve`
- **Description:** Mark an issue as resolved.
- **Authentication:** Required
- **Request Body:**

  ```
  {
    "issue_id": "number"
  }
  ```

- **Response:**
  - **200 OK** (`data` will contain the resolved issue)

    ```
    {
      "success": true,
      "code": 200,
      "data": {}
    }
    ```

PROF

  - **401 Unauthorized**

    ```
    {
      "success": false,
      "code": 401,
      "error": "You need be to signed in to resolve an issue."
    }
    ```

  - **404 Not Found**

    ```
    {
      "success": false,
    ```

```
  "code": 404,
  "error": "Issue does not exist."
}
```

- **500 Internal Server Error**

```
{
  "success": false,
  "code": 500,
  "error": "An unexpected error occurred. Please try again later."
}
```

5. React to an Issue

- **Method:** `POST`
- **Endpoint:** `/api/reactions`
- **Description:** React to an issue with an emoji.
- **Authentication:** Required
- **Request Body:**

```
{
  "issue_id": "number",
  "emoji": "string"
}
```

- **Response:**
  - **200 OK**

```
{
  "success": true,
  "code": 200,
  "data": {
    "added?": "string",
    "removed?": "string"
  }
}
```

> The given emoji is added as a reaction and returned. If the user already had a reaction it is removed and also returned. If the new reaction is the same as the old one, the reaction is just removed

  - **400 Bad Request** (Invalid input data)

PROF

```
{
  "success": false,
  "code": 400,
  "error": "Missing required fields for reacting"
}
```

- **401 Unauthorized**

```
{
  "success": false,
  "code": 401,
  "error": "You need to be signed in to react."
}
```

- **500 Internal Server Error**

```
{
  "success": false,
  "code": 500,
  "error": "An unexpected error occurred. Please try again
later."
}
```

6. Comment on an Issue

- **Method:** `POST`
- **Endpoint:** `/api/comments/add`
- **Description:** Comment on an issue.
- **Authentication:** Required
- **Request Body:**

PROF

```
{
  "issue_id": "number",
  "content": "string",
  "is_anonymous": "boolean",
  "parent_id?": "number"
}
```

- **Response:**
  - **201 Created** (`data` will contain the created comment)

```
{
  "success": true,
  "code": 200,
  "data": {}
}
```

- **400 Bad Request** (Invalid input data)

```
{
  "success": false,
  "code": 400,
  "error": "Missing required fields for creating a comment"
}
```

- **401 Unauthorized**

```
{
  "success": false,
  "code": 401,
  "error": "You need to be signed in to comment."
}
```

- **500 Internal Server Error**

```
{
  "success": false,
  "code": 500,
  "error": "An unexpected error occurred. Please try again
later."
}
```

PROF

7. Get Comments for an Issue

- **Method:** `POST`
- **Endpoint:** `/api/comments`
- **Description:** Retrieve a range comments for an issue.
- **Request Body:**

```
{
  "issue_id": "number",
  "from": "number",
  "amount": "number",
  "parent_id?": "number",
  "user_id?": "string"
}
```

- **Response:**
  - **200 OK**

    ```
    {
      "success": true,
      "data": [
        {
          "comment_id": "number",
          "issue_id": "number",
          "user_id": "string",
          "content": "string",
          "created_at": "string",
          "parent_comment_id": "number",
          "user": {
            "user_id": "string",
            "email_address": "string",
            "username": "string",
            "fullname": "string",
            "image_url": "string"
          },
          "is_owner": "boolean"
        }
      ]
    }
    ```

  - **404 Not Found**

    ```
    {
      "success": false,
      "code": 404,
      "error": "Comment does not exist."
    }
    ```

  - **500 Internal Server Error**

    ```
    {
      "success": false,
      "code": 500,
      "error": "An unexpected error occurred. Please try again
    later."
    }
    ```

PROF

8. Get User Profile

- **Method:** `GET`
- **Endpoint:** `/api/users/{user_id}`
- **Description:** Retrieve a user's profile.
- **Path Parameters:** `user_id` (string) - ID of the user
- **Response:**
  - **200 OK**

```
{
  "success": true,
  "code": 200,
  "data": {
    "user_id": "string",
    "email_address": "string",
    "username": "string",
    "fullname": "string",
    "image_url": "string",
    "posts": [
      {
        "issue_id": "number",
        "content": "string",
        "created_at": "string"
      }
    ],
    "is_owner": "boolean"
  }
}
```

  - **404 Not Found** (User not found)

```
{
  "success": false,
  "code": 404,
  "error": "User not found."
}
```

  - **500 Internal Server Error**

```
{
  "success": false,
  "code": 500,
  "error": "An unexpected error occurred. Please try again
later."
}
```

PROF

9. Update User Profile

- **Method:** PUT
- **Endpoint:** `/api/users/{user_id}`
- **Description:** Update a user's profile.
- **Path Parameters:** `user_id` (string) - ID of the user
- **Request Body:**

```
{
  "email_address": "string", // optional
  "username": "string", // optional
  "fullname": "string", // optional
  "image_url": "string" // optional
}
```

- **Response:**
  - **200 OK**

    ```
    {
      "success": true,
      "code": 200,
      "data": {}
    }
    ```

  - **400 Bad Request** (Invalid input data)

    ```
    {
      "success": false,
      "code": 400,
      "error": "Invalid input data. Please provide valid fields."
    }
    ```

  - **404 Not Found** (User not found)

    ```
    {
      "success": false,
      "code": 404,
      "error": "User not found."
    }
    ```

  - **500 Internal Server Error**

PROF

```
{
  "success": false,
  "code": 500,
  "error": "An unexpected error occurred. Please try again
later."
}
```

10. Subscribe to an Issue

- **Method:** `POST`
- **Endpoint:** `/api/subscriptions/issues`
- **Description:** Subscribe to an issue.
- **Request Body:**

```
{
  "user_id": "string", // required
  "issue_id": "number" // required
}
```

- **Response:**
  - **201 Created**

```
{
  "success": true,
  "code": 201,
  "data": {}
}
```

  - **400 Bad Request** (Invalid input data)

```
{
  "success": false,
  "code": 400,
  "error": "Invalid input data. Please provide all required
fields."
}
```

  - **409 Conflict** (Already subscribed)

```
{
  "success": false,
  "code": 409,
```

PROF

```
  "error": "User is already subscribed to the issue."
}
```

- **500 Internal Server Error**

```
{
  "success": false,
  "code": 500,
  "error": "An unexpected error occurred. Please try again
later."
}
```

11. Subscribe to a Category in a Location

- **Method:** `POST`
- **Endpoint:** `/api/subscriptions/categories`
- **Description:** Subscribe to a category in a location.
- **Request Body:**

```
{
  "user_id": "string", // required
  "category_id": "number", // required
  "location_id": "number" // required
}
```

- **Response:**
  - **201 Created**

```
{
  "success": true,
  "code": 200,
  "data": {}
}
```

  - **400 Bad Request** (Invalid input data)

```
{
  "success": false,
  "code": 400,
  "error": "Invalid input data. Please provide all required
fields."
}
```

PROF

   - **409 Conflict** (Already subscribed)

     ```
     {
       "success": false,
       "code": 409,
       "error": "User is already subscribed to the category in the
     location."
     }
     ```

   - **500 Internal Server Error**

     ```
     {
       "success": false,
       "code": 500,
       "error": "An unexpected error occurred. Please try again
     later."
     }
     ```

12. Subscribe to a Location

- **Method:** `POST`
- **Endpoint:** `/api/subscriptions/locations`
- **Description:** Subscribe to a location.
- **Request Body:**

  ```
  {
    "user_id": "string", // required
    "location_id": "number" // required
  }
  ```

PROF

- **Response:**
  - **201 Created**

    ```
    {
      "success": true,
      "code": 200,
      "data": {}
    }
    ```

  - **400 Bad Request** (Invalid input data)

```
{
  "success": false,
  "code": 400,
  "error": "Invalid input data. Please provide all required
fields."
}
```

- **409 Conflict** (Already subscribed)

```
{
  "success": false,
  "code": 409,
  "error": "User is already subscribed to the location."
}
```

- **500 Internal Server Error**

```
{
  "success": false,
  "code": 500,
  "error": "An unexpected error occurred. Please try again
later."
}
```

13. Unsubscribe from an Issue

- **Method:** `DELETE`
- **Endpoint:** `/api/subscriptions/issues`
- **Description:** Unsubscribe from an issue.
- **Request Body:**

```
{
  "user_id": "string", // required
  "issue_id": "number" // required
}
```

- **Response:**
  - **200 OK**

```
{
  "success": true,
  "code": 200,
  "data": {}
}
```

- **400 Bad Request** (Invalid input data)

  ```
  {
    "success": false,
    "code": 400,
    "error": "Invalid input data. Please provide all required fields."
  }
  ```

- **404 Not Found** (Subscription not found)

  ```
  {
    "success": false,
    "code": 404,
    "error": "Subscription not found."
  }
  ```

- **500 Internal Server Error**

  ```
  {
    "success": false,
    "code": 500,
    "error": "An unexpected error occurred. Please try again later."
  }
  ```

PROF

14. Unsubscribe from a Category in a Location

- **Method:** `DELETE`
- **Endpoint:** `/api/subscriptions/categories`
- **Description:** Unsubscribe from a category in a location.
- **Request Body:**

  ```
  {
    "user_id": "string", // required
    "category_id": "number", // required
    "location_id": "number" // required
  }
  ```

- **Response:**
  - **200 OK**

    ```
    {
      "success": true,
      "code": 200,
      "data": {}
    }
    ```

  - **400 Bad Request** (Invalid input data)

    ```
    {
      "success": false,
      "code": 400,
      "error": "Invalid input data. Please provide all required
    fields."
    }
    ```

  - **404 Not Found** (Subscription not found)

    ```
    {
      "success": false,
      "code": 404,
      "error": "Subscription not found."
    }
    ```

  - **500 Internal Server Error**

    ```
    {
      "success": false,
      "code": 500,
      "error": "An unexpected error occurred. Please try again
    later."
    }
    ```

PROF

15. Unsubscribe from a Location

- **Method:** `DELETE`
- **Endpoint:** `/api/subscriptions/locations`
- **Description:** Unsubscribe from a location.
- **Request Body:**

```
{
  "user_id": "string", // required
  "location_id": "number" // required
}
```

- **Response:**
  - **200 OK**

    ```
    {
      "success": true,
      "code": 200,
      "data": {}
    }
    ```

  - **400 Bad Request** (Invalid input data)

    ```
    {
      "success": false,
      "code": 400,
      "error": "Invalid input data. Please provide all required
    fields."
    }
    ```

  - **404 Not Found** (Subscription not found)

    ```
    {
      "success": false,
      "code": 404,
      "error": "Subscription not found."
    }
    ```

  - **500 Internal Server Error**

    ```
    {
      "success": false,
      "code": 500,
      "error": "An unexpected error occurred. Please try again
    later."
    }
    ```

PROF

16. Get User's Subscriptions

- **Method:** `GET`
- **Endpoint:** `/api/subscriptions/{user_id}`
- **Description:** Retrieve a user's subscriptions.
- **Path Parameters:** `user_id` (string) - ID of the user
- **Response:**
  - **200 OK**

    ```
    {
      "success": true,
      "code": 200,
      "data": {
        "issues": [
          {
            "issue_id": "number",
            "content": "string",
            "created_at": "string"
          }
        ],
        "categories": [
          {
            "category_id": "number",
            "name": "string",
            "location_id": "number"
          }
        ],
        "locations": [
          {
            "location_id": "number",
            "name": "string"
          }
        ]
      }
    }
    ```

PROF

  - **404 Not Found** (User not found)

    ```
    {
      "success": false,
      "code": 404,
      "error": "User not found."
    }
    ```

  - **500 Internal Server Error**

    ```
    {
      "success": false,
    ```

```
  "code": 500,
  "error": "An unexpected error occurred. Please try again
later."
}
```

17. Get User's Issues

- **Method:** `GET`
- **Endpoint:** `/api/users/{user_id}/issues`
- **Description:** Retrieve a user's issues.
- **Path Parameters:** `user_id` (string) - ID of the user
- **Query Parameters:**
  - `status` (string) - Filter issues by status (e.g., "resolved" or "unresolved")
    - Example: `/api/users/{user_id}/issues?status=resolved`
- **Response:**
  - **200 OK**

```
{
  "success": true,
  "code": 200,
  "data": {
    "issues": [
      {
        "issue_id": "number",
        "user_id": "string",
        "location_id": "number",
        "category_id": "number",
        "content": "string",
        "image_url": "string",
        "is_anonymous": "boolean",
        "created_at": "string",
        "resolved_at": "string",
        "sentiment": "string",
        "category": {
          "name": "string"
        },
        "reactions": [
          {
            "emoji": "string",
            "count": "number"
          }
        ],
        "comment_count": "number"
      }
    ],
    "is_owner": "boolean"
  }
}
```

PROF

- **404 Not Found** (User not found)

```
{
  "success": false,
  "code": 404,
  "error": "User not found."
}
```

- **500 Internal Server Error**

```
{
  "success": false,
  "code": 500,
  "error": "An unexpected error occurred. Please try again
later."
}
```

18. Get User's Notifications

- **Method:** `GET`
- **Endpoint:** `/api/notifications/{user_id}`
- **Description:** Retrieve a user's notifications.
- **Path Parameters:** `user_id` (string) - ID of the user
- **Response:**
  - **200 OK**

```
{
  "success": true,
  "code": 200,
  "data": [
    {
      "notification_id": "number",
      "user_id": "string",
      "content": "string",
      "created_at": "string",
      "is_read": "boolean",
      "issue_id": "number",
      "category_id": "number",
      "location_id": "number"
    }
  ]
}
```

  - **404 Not Found** (User not found)

PROF

```json
{
  "success": false,
  "code": 404,
  "error": "User not found."
}
```

- **500 Internal Server Error**

```json
{
  "success": false,
  "code": 500,
  "error": "An unexpected error occurred. Please try again later."
}
```

Rate Limiting

To prevent abuse, rate limiting should be applied to all endpoints. Users are allowed a maximum number of requests per minute. If the limit is exceeded, the API will respond with a `429 Too Many Requests` status code along with an error message:

```json
{
  "success": false,
  "code": 429,
  "error": "Too many requests. Please try again later."
}
```