API documentation oblig 3

Table of Contents

API Specification introduction

Title

SUPER Assessor Card Game

Authors

- Sebastian Skrøvseth Haugen
- Hannah Sofie Eriksen
- Jon Helge Skjærstein

Problem:

Administrators need a secure and intuitive interface to efficiently manage card and user collections for the SUPER Assessor platform.

Solution:

To address this problem, we are implementing a website using react for the frontend and node for the backend. We will be using JWT and other authentication methods to make sure that only authorized personnel can make changes to the cards and users.

Implementation:

Authentication and Authorization: Implement a secure login system with role-based access (user and admins) so that only admins can change card and users' data.

Functionalities: Develop features for card and user management, including CRUD operations (Create, Read, Update, Delete) for card collections, user account management functionalities, and a dashboard providing statistics.

Testing: Conduct testing of the administrative interface to ensure functionality, usability, and security. Perform unit tests, integration tests, and end-to-end tests to validate the behavior of the interface under various scenarios.

Documentation: Provide comprehensive documentation for administrators, including API documentation and troubleshooting instructions. This helps administrators understand how to use the interface effectively and troubleshoot any issues that they may have.

Authentication

Developers gain access via API keys obtained through registration and authentication process.

Spec

You can structure this section in different ways. Organised by resources (if rest), by events, etc. You can add use the following tables.

Errors

HTTP status code	Error code	Verbose error	Description
200	200_OK	OK	The request has succeeded. The requested data is in the response body.
404	404_NOT_FOUND	Not Found	The requested data could not be found in the database.
500	500_INTERNAL_SERVER_ERROR	Internal server error	An error occurred on the server while attempting to retrieve the data.

Endpoints/web services (for Rest or RPC)

If many “resources”, it can be structured by “resource”. If not, all web services in one table

Collection 1 (e.g. users)

USERS

URI	Inputs	Outputs
POST http://localhost:8001/api/auth/register	User object (body parameter)	Creates a new user
POST http://localhost:8001/api/auth/login	Credentials (body parameter)	Logs in to a user profile
POST http://localhost:8001/api/auth/logout	None	Logs out of a user profile
GET http://localhost:8001/api/users	None	Retrieves all users
GET http://localhost:8001/api/users/:id	id (path parameter)	Retrieves a user with the specified ID

PUT http://localhost:8001/api/users/:id	id (path parameter), User object	Updates a user with the specified ID
DELETE http://localhost:8001/api/users/:id	Id (path parameter)	Deletes a user with the specified ID
GET http://localhost:8001/api/auth/:role	Role (path parameter)	Retrieves information about the specified role
GET http://localhost:8001/api/users/count	None	Retrieves the total number of users

MISSIONS

URI	Inputs	Outputs
GET /api/cards/mission	None	Returns a list of all mission cards
GET /api/cards/mission/:id	id (path parameter)	Returns the mission card with the specified id
POST /api/cards/mission	Mission object (body parameter)	Creates a new mission card with specified details
PUT /api/cards/mission/:id	id (path parameter), Mission object (body parameter)	Updates the mission card with specified id
DELETE /api/cards/mission/:id	id (path parameter)	Deletes the mission card with the specified id

ASSESSMENT

URI	Inputs	Outputs
GET /api/cards/assessment	None	Returns a list of all assessment cards
GET /api/cards/assessment/:id	id (path parameter)	Returns the assessment card with the specified id
POST /api/cards/assessment	Assessment object (body parameter)	Creates a new assessment card with specified details
PUT /api/cards/assessment/:id	id (path parameter), Assessment object (body parameter)	Updates the assessment card with specified id
DELETE /api/cards/assessment/:id	id (path parameter)	Deletes the assessment card

		with the specified id
GET /api/cards/assessment/icon/:category	Category (path parameter)	Gets the icon for the specified category
POST /api/cards/upload/upload-icon/:category	Category (path parameter)	Updates the icon for the category
POST /api/cards/upload/upload-json	JSON file	JSON file with data on cards to be saved for the DB

Events and payloads (for Event-driven APIs)

USERS

Events	Payload
User created	User object (json)
User updated	User object (json)
User deleted	User ID (string)

MISSION

Events	Payload
PUT /api/cards/mission/:id	Mission object (body parameter)
POST /api/cards/mission	Assessment object (request body)

ASSESSMENT

Events	Payload
POST /api/cards/assessment	Assessment object (request body)
PUT /api/cards/assessment/:id	Assessment object (body parameter)
POST /api/cards/upload/upload/upload-json	JSON file (request body)
POST /api/cards/upload/upload-icon/:category	Image file (request body)