SUPER Assessor Card API

Oblig 2 Martin Heggholmen

Table of Contents

[API Specification introduction 3](#_Toc63373636)

[Title 3](#_Toc63373637)

[Authors 3](#_Toc63373638)

[Problem 3](#_Toc63373639)

[Solution 3](#_Toc63373640)

[Implementation 3](#_Toc63373641)

[Authentication 4](#_Toc63373642)

[Other things we considered 4](#_Toc63373643)

[Spec 5](#_Toc63373644)

[Errors 5](#_Toc63373645)

[Endpoints/web services (for Rest or RPC) 5](#_Toc63373646)

[Collection 1 (e.g. users) 5](#_Toc63373647)

[Events and payloads (for Event-driven APIs) 7](#_Toc63373648)

# API Specification introduction

## Title

SUPER Assessor Card API

## Authors

* Martin Heggholmen

## Problem

The problem I want to solve with this API is that regular learning environments are not engaging enough, the students is so used to the same old linear lecture with PowerPoint and the teacher singing the same old songs, this is what we want to solve with the SUPER Assessor Card API

## Solution

The solution I have come up with is to develop a fantastic tool to spice up every day lectures and make them special, to do this I have created an API where you can fetch cards in different categories, one category is assessment cards, this is the strategy you are implementing to do cards in the second category which is mission cards, the mission cards contains your mission, the thing you are trying to accomplish.

## Implementation

Discussion about technology: paradigm picked and list of reasons

For the Super Assessor project I used MongoDB and RESTful APIs mostly because of the high speed, scalability, simplicity, and the fact that it is well documented, because MongoDB and RESTful APIs are widely used.

RESTful APIs are straightforward to implement and interact with, using standard HTTP methods and status codes, which enhances developer productivity and reduces the learning curve.

The database was easily created in MongoDB atlas, and did requests to this database through POSTMAN, both services was pretty out-of-the-box, and had a step by step setup, with good documentations and tutorials on how to set it up.

To host the index.js I used NODE and ran queries through POSTMAN.

## Authentication

To give API access to users you must first send in a request, afterwards you can get access to an API key, and thereafter make API requests, currently its just configured to my personal account and password through .env file stored locally.

## Other things we considered

|  |  |  |  |
| --- | --- | --- | --- |
| Pattern, paradigm or protocol considered | Pros | Cons | Selected? |
| MongoDB | * High speed * Simple * Quick set-up * Flexible * Scaleable | * Limited data size * Nesting * Duplicates * High memory usage | Yes |
| GraphQL | * Gets exactly the data you need * Flexible efficient data retrieval * Reduces the number of request needed | * higher complexity on the server side * can be overkill for simple APIs | No |
| RESTful API | * widely adopted and supported * easy to understand * statelessness * better performance than most alternatives | * requires multiple requests | yes |

# 

# 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 |
| 400 | BadRequest | Indalid Card type | The card type is invalid, so the request could not be processed |
| 404 | NotFound | Card Not Found | The card was not found in the database |
| 404 | NotFound | Document to update was not found | No document available to add to the new card |
| 500 | InternalServerError | Error fetching cards | An unexpected error occurred while attempting to retrieve cards |
| 500 | InternalServerError | Error adding new card | An unexpected error occurred while attempting to add a new card to the database |
| 500 | InternalServerError | Error updating card | An unexpected error occurred while attempting to update a card in the database |
| 500 | InternalServerError | Error deleting card | An unexpected error occurred while attempting to delete a card from the database |

## Endpoints/web services (for Rest or RPC)

### Collection 1 (e.g. users)

Get all cards

{{host}}/cards

Get all missions

{{host}}/cards?type=Missions

Get all assessment cards

{{host}}/cards?type=Assessment cards

Get single card

{{host}}/cards/15

Post card example:

{{host}}/cards/

{

"card-id":"15",

"card-type":"Mission",

"card-name":"hehe",

"card-category":"Category of the mission",

"card-description":"Description of the mission",

"card-details":"Details about the mission"

}

update card example:

{{host}}/cards/15

{

"card-id":"15",

"card-type":"Mission",

"card-name":"Updated Mission Name",

"card-description":"Updated description",

"card-details":"Updated details",

"card-category":"Updated category"

}

Delete card example:

{{host}}/cards/11

## Events and payloads (for Event-driven APIs)

|  |  |
| --- | --- |
| Events | Payload |
| CardCreated | New Card |
| CardUpdated | Updated Card |
| CardDeleted | Deleted Card |
|  |  |