

09a – Documentation

Package.json:

```
"swagger-jsdoc": "^6.2.8",  
"swagger-ui-express": "^5.0.0",
```

Config:

```
import swaggerJSDoc from "swagger-jsdoc";  
  
const swaggerDefinition = {  
  openapi: "3.0.0",  
  info: {  
    title: "Express API for Projects",  
    version: "1.0.0",  
    description: "This is a REST API application made with Express.",  
  },  
  servers: [  
    {  
      url: "http://localhost:4000",  
      description: "Development server",  
    },  
  ],  
};  
  
const options = {  
  swaggerDefinition,  
  apis: ["../routes/*.ts", "./apiComponents.ts"],  
};  
export const swaggerSpec = swaggerJSDoc(options);
```

apiComponent:

```
1  /**  
2   * @openapi  
3   * components:  
4   *   schemas:  
5   *     Project:  
6   *       type: object  
7   *       properties:  
8   *         id:  
9   *           type: integer  
0   *         uuid:  
1   *           type: string  
2   *           format: uuid  
3   *         name:  
4   *           type: string  
5   *         description:  
6   *           type: string  
7   *           nullable: true
```

Serve swagger-ui at api/docs for api documentation

```
import swaggerUi from "swagger-ui-express"; 8.2k (gzipped: 3.3k)
import { swaggerSpec } from "../swaggerConfig";

const app: any = express();

async function startServer() {
  app.use(express.json());


  app.use("/api-docs", swaggerUi.serve, swaggerUi.setup(swaggerSpec));
}
```

Bruger OpenAPI-jsdoc biblioteket hvilket gør at jeg kombinerer JSdoc-stil kommentar med OpenAPI for at generere API dokumentation.

```
/**
 * @openapi
 * /projects:
 *   get:
 *     summary: Retrieves all projects
 *     description: Returns a list of all projects, including their snapshots and tombstones.
 *     responses:
 *       200:
 *         description: A list of projects
 *         content:
 *           application/json:
 *             schema:
 *               type: array
 *               items:
 *                 $ref: '#/components/schemas/Project'
 *       500:
 *         description: Internal server error
 */

router.get("/projects", async (req: Request, res: Response) => {
  try {
    const projects = await prisma.project.findMany({
      include: {
        snapshots: true,
        tombstones: true,
      },
    });
    res.send(projects);
  } catch (error) {
    console.error(error);
    res.status(500).send({ error: "Failed to fetch projects" });
  }
});
```

Swagger-UI

 **Swagger**
Support by SMARTBEAR

Express API for Projects 1.0.0 OAS 3.0

This is a REST API application made with Express.

Servers

http://localhost:4000 - Development server

default

GET

/projects

Retrieves all projects

POST

/projects

Creates a new project with an image uploaded to AWS S3.

POST

/projects/update

Updates an existing project and creates a snapshot of the previous state.

POST

/projects/delete

Marks a project as deleted based on UUID provided.

Schemas

Teste /projects

GET /projects

Retrieves all projects

Returns a list of all projects, including their snapshots and tombstones.

Parameters

No parameters

Cancel

Execute

Clear

Responses

Curl

curl -X 'GET' \ 'http://localhost:4000/projects' \ -H 'accept: application/json'

Request URL

http://localhost:4000/projects

Server response

Code

Details

200

Response body

[{"id": 4, "uuid": "4a381e31-18d6-4973-9ab8-8c645a213695", "name": "Testigenigen", "description": "tesigengein", "imageUrl": "https://mybucket-zando-commerce.s3.amazonaws.com/8c64f983-65ce-41d2-b984-f9738cc2174d-bedste-seng-i-test.jpg", "createdAt": "2024-04-29T12:49:28.777Z", "updatedAt": "2024-04-29T12:49:28.777Z", "isDeleted": false, "versionId": "4ndc26de-75ca-48f2-cc6fa-7b51dc930b22"}]

Database scema / hænger sammen med apiComponent.

Schemas

Project

id

uuid

name

description

imageUrl

createdAt

updatedAt

isDeleted

userId

GUID

isSnapshot

snapshotDate

snapshots

tombstones

integer

string(\$uuid)

string

string

string

string(\$date-time)

string(\$date-time)

boolean

string

string

boolean

string(\$date-time)

[ProjectSnapshot > {...}]

[ProjectTombstone > {...}]

nullable: true

default: false

nullable: true

nullable: true

default: false

nullable: true

ProjectSnapshot

>

ProjectTombstone

>