

BOOKING SYSTEM TIL VIDENSCENTER FOR ROBOT OG AUTOMATIONSTEKNOLOGI

Dokumentation



Info

The whole codebase, webside, api, and mobile apps are located under the repository <https://github.com/JesperLeoMoberg/sdetobotdb>
It's owned by Jesper Leo Moberg, so only he can give access

The website itself will be on <http://udstyr.sde.dk/> and api on <http://udstyr.sde.dk/api>
The android application for android will be in the repository, it can be compiled to apk inside intellij

If you have an experience with android development or eventually ios development, nodejs, and php, then it should be easy.

If you don't then you shouldn't probably rewrite the project.

Indholdsfortegnelse

Info	1
Website	3
Api	4
Libraries.....	4
Routes	5
Routes 1.1	5
Routes 1.2	6
Routes 1.3	7
Routes 1.4	8
Routes 1.5	9
Database	10
Database diagram	11

Website

The site is pure php, might use lavarel later to make it easier. The admin dashboard lies in /admin, it can only be accessed by an admin user with rank 1.

The user page lies in /user and is composed based on the current logged in user.

The website will NOT work without the api running.

Api

The api is the interface to the backend and database. It makes so that all clients are connected properly and secure to the system.

Libraries

When got access and cloned the project, you should always run `npm i` in cmd in the api folder, ensuring you have nodejs installed. This will install all the libraries mentioned in the `package.json`. (An visual of the file is shown below)

```
{
  "name": "api",
  "version": "1.0.0",
  "dependencies": {
    "boom": "latest",
    "hapi": "^18.1.0",
    "joi": "latest",
    "jsonwebtoken": "^8.5.1",
    "uuid": "^3.3.2"
  },
  "devDependencies": {
    "hapi-auth-jwt2": "^8.3.0",
    "hapi-mysql2": "^2.2.5"
  }
}
```

Routes

Routes 1.1

Product routes 1.1

`/booking/products/location/create` **post**

Authorization required

Creates a room location in the database and responds with json data that has insertId

Requires room name as payload. Id is auto incremented

`/booking/products/svf/create` **post**

Authorization required

Creates a location type svf in the database and responds with json data that has insertId

Requires number and type in the payload

`/booking/products/thp/create` **post**

Authorization required

Creates a location type thp in the database and responds with json data that has insertId

Requires number and type in the payload

`/booking/products/category/create` **post**

Authorization required

Creates a category in the database and responds with json data that has insertId

Requires name in the payload, Id is auto incremented

`/booking/products/address/create` **post**

Authorization required

Creates an address and address short in the database and responds with json data that has insertId

Payload example:

```
$data = array(  
  'schoolname' => $explodedarray[0],  
  'city' => $explodedarray[2],  
  'zipcode' => $explodedarray[3],  
  'address' => $explodedarray[4],  
  'short' => $explodedarray[1],  
);
```

Routes 1.2

Product routes 1.2

`/booking/products/suppliers/create` *post*

Authorization required

Creates a supplier in the database

Payload example:

```
$data = array(  
    'name' => $splittedValueArray[0],  
    'address' => $splittedValueArray[1],  
    'call_number' => $splittedValueArray[2],  
);
```

`/booking/products/create` *post*

Authorization required

Creates a product in the database and responds with json data that has insertId

The payload will require school name short id, supplier company id, and category id as shown below. The client will ask the api if the id exist in the tables, if not it will register new ones and respond with id for the product creation function to use for payload.

You can move this to the api/backend.

Payload example:

```
$data = array(  
    'product_name' => $product,  
    'movable' => $flytbar,  
    'description' => $description,  
    'school_name_short_id' => $virksomhed,  
    'category_id' => $category,  
    'supplier_company_id' => $leverandoer,  
);
```

Routes 1.3

Product routes 1.3

`/booking/products/units/create` **post**

Authorization required

Creates a unit from a product in the database and responds unit created on success and error on fail.

The payload will require school name product id, location room id, product location type svf id, and product location type thp id as shown below.

If one of these ids is undefined, it will fail, even though it might have created the product itself beforehand, which is not good and will be changed

You can move this to the api/backend.

Payload example:

```
$data = array(  
    'unit_number' => $convertEnhedNumberToString,  
    'products_id' => $product_id,  
    'location_room_id' => $lokale,  
    'product_location_type_svf_id' => $svf,  
    'product_location_type_thp_id' => $thp,  
);
```

The reason why these are post is because to not have a long get query.

`/booking/products/location/get` **post**

Authorization required

Returns json data with id based on payload

Payload example:

```
$data = array(  
    'lokale' => $array[3]  
);
```

`/booking/products/svf/get` **post**

Authorization required

Returns json data with id based on payload

Payload example:

```
$data = array(  
    'id' => $array[4],  
    'type' => $firstletter,  
    'nr' => $number  
);
```


Routes 1.4

Product routes 1.4

`/booking/products/thp/get` **post**

Authorization required

Returns json data with id based on payload

Payload example:

```
$data = array(  
  'id' => $array[5],  
  'type' => $firstletter,  
  'nr' => $number  
);
```

`/booking/products/address/get` **post**

Authorization required

Returns json data with id based on payload

Payload example:

```
$data = array(  
  'id' => $array[2],  
);
```

`/booking/products/category/get` **post**

Authorization required

Returns json data with id based on payload

Payload example:

```
$data = array(  
  'id' => $array[0],  
);
```

Routes 1.5

Product routes 1.5

`/booking/products/suppliers/get` **post**

Authorization required

Returns json data with id based on payload

Payload example:

```
$data = array(  
  'id' => $array[6],  
);
```

`/booking/products/products/get` **get**

Authorization required

Returns json data with product information

No payload since it's a get method

`/booking/products/units/get` **get**

Authorization required

Returns json data with unit information

No payload since it's a get method

User routes 1.1

`/users/login` **post**

The login route returns user information on success and password incorrect on password incorrect match and email doesn't exist on nonexistent email.

Authorization not required

Requires email and password in payload.

`/users/register` **post**

Registers a user into the database

Authorization not required

Requires email and password in payload.

```
$data = array(  
  'name' => $_POST['name'],  
  'email' => $_POST['email'],  
  'password' => $_POST['password_1']  
);
```

Database

The database will be on the same server as the api/backend server.

Database diagram

