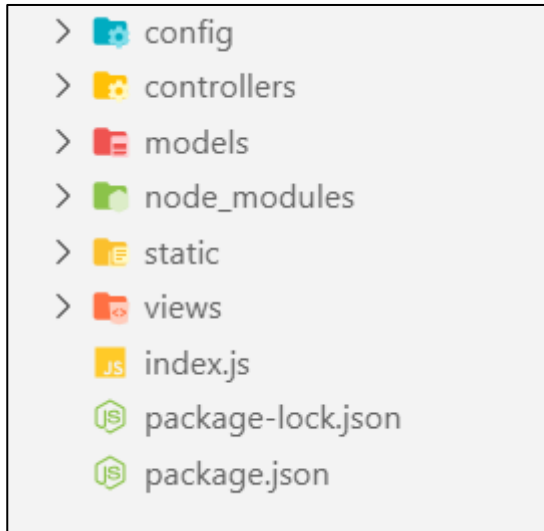


# Workshop: Cubicle - Part 1

"Cubicle" is a place, where you can browse some of the most popular Rubik cubes in the world and add some new cubes that you have discovered.

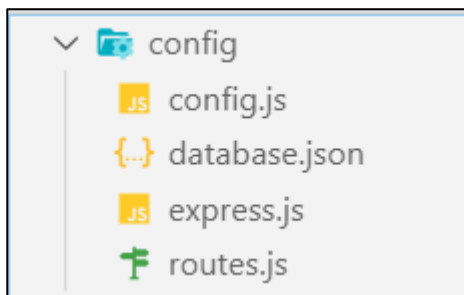
## Folder Structure View

You're provided with a project structure skeleton like this:



This structure includes the following configurations:

## Config Folder View



## Config.js



## Database.json

```
config > {} database.json
```

```
1  []
2
```

## Express.js

```
config > .js express.js > ...
```

```
1  const express = require('express');
2  const handlebars = require('express-handlebars');
3  const bodyParser = require('body-parser');
4
5  module.exports = (app) => {
6
7      // TODO: Setup the view engine
8
9      // TODO: Setup the body parser
10
11     // TODO: Setup the static files
12
13 };|
```

## Routes.js

```
config > .js routes.js > <unknown> > module.exports
```

```
1  // Require Controllers
2
3  module.exports = (app) => {
4      // TODO ...
5  };|
```

## Index.js

```
.js index.js > ...
```

```
1  const env = process.env.NODE_ENV || 'development';
2
3  const config = require('./config/config')[env];
4  const app = require('express')();
5
6  require('./config/express')(app);
7  require('./config/routes')(app);
8
9  app.listen(config.port, console.log(`Listening on port ${config.port}! Now its up to you...`));|
```

# Install Dependencies

Run `'npm install'` to install all the dependencies.

## 1. Create Model

### 1.1 Cube Model

Each cube should have the following properties (for now it could be an ES6 class):

- **Id** - number
- **Name** – string
- **Description** – string
- **Image URL** – string
- **Difficulty Level**– number

## 2. Storage

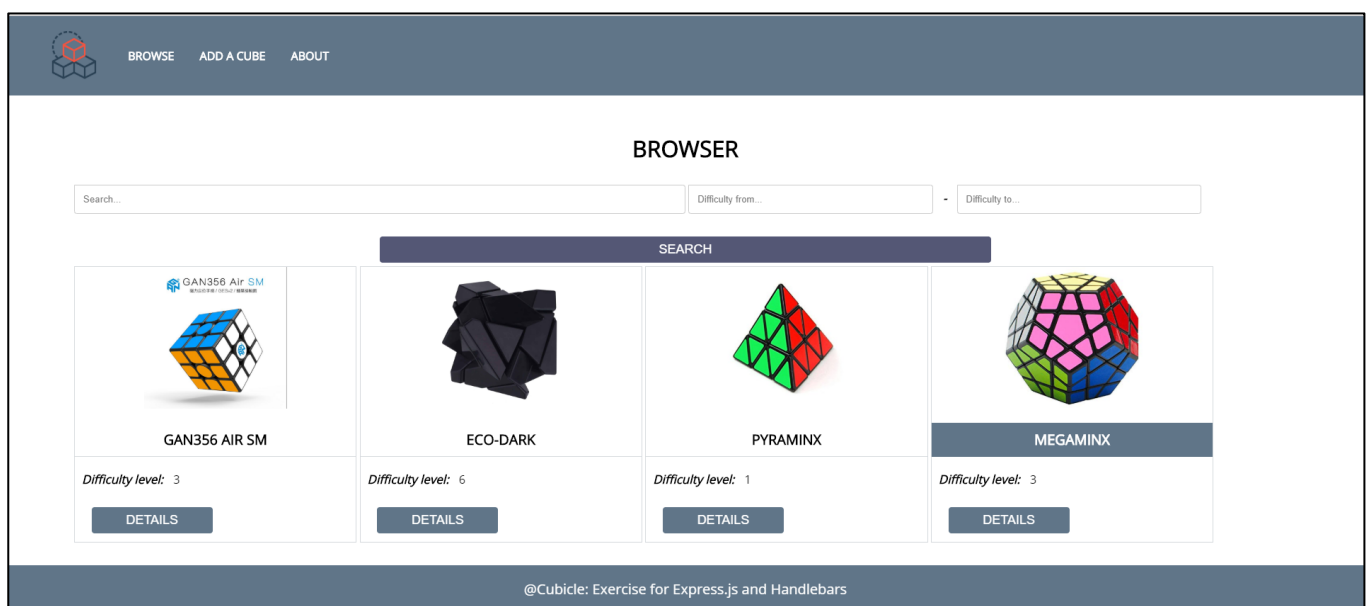
Store the cubes inside a `/config/database.json`


## 3. Create Routes

You should implement the following routes:

- `/` - the main page (should visualize all the cubes in the database and a search field)
- `/about` – should render the about page
- `/create` – should render the create cube form
- `/details/:id` – should render the details page about selected cube
- **Any other** - should render the 404 not found page

## Main Page



BROWSEADD A CUBEABOUT


BROWSER

SEARCH

NO CUBES WERE FOUND

@Cubicle: Exercise for Express.js and Handlebars

## About Page


BROWSEADD A CUBEABOUT

ABOUT CUBICLE

"Cubicle" is a place, where you can browse some of the most popular rubik cubes in the world and add some new cubes that you have discovered. This application was created as an exercise for the **JS BACK-END COURSE** at the **SOFTWARE UNIVERSITY**.

@Cubicle: Exercise for Express.js and Handlebars

## Create Page

BROWSEADD A CUBEABOUT

CREATE

Name

Description

ImageUrl


Difficulty

1 - Very Easy


CREATE

@Cubicle: Exercise for Express.js and Handlebars

## Details Page

BROWSEADD A CUBEABOUT

### GAN356 AIR SM



**Description:** Magnets in AirSM will not drop, and their positions will be more precise with the Magnets-Snap-On design. With the use of 3mm\*2mm magnets, the handfeel will be more stable and more comfortable. P.S. This design is brand new for the AirSM.

**Difficulty level:** 3

BACK

@Cubicle: Exercise for Express.js and Handlebars

## Not Found Page

BROWSEADD A CUBEABOUT



404  
Page not found

@Cubicle: Exercise for Express.js and Handlebars

## 4. Create Templates

Use the provided HTML to create templates using Handlebars. Identify the dynamic parts and use appropriate syntax for interpolating and rendering the application context.

## 5. \*Search

Implement searching logic. Use the following validation:

- If the user searches **only** a string and **NO difficulty**, render **all difficulties**

If the search does **NOT** meet the requirements, just **redirect** to the home page ('/').

# Good Luck! 🤗