**Node.js / Express.js**

**DEADLINE:** 05/05/2019

## FOLDER STRUCTURE

flx\_homework\_nodejs-expressjs/\*

   homework/\*

       src/\*

           handlers/\*

           car.js\*

            middlewares/\*

           delete-authorization.js\*

            routes.js\*

            server.js\*

       db/\*

            data.json\*

      tests/\*

           api.test.js\*

package.json\*

**PREREQUISITES**

* Open the homework folder
* Navigate to the folder which contains the package.json file
* Open command line
* Execute the **npm install** command to install all required dependencies

## TASK

In this task you need to build a RESTful CRUD API server using Node.js and Express.js framework. Follow the next steps:

* Create a new server, which will listen port **3000** using [Express.js](http://expressjs.com/).
* As the main storage, you have the file **data.json** in the **db** folder.
* Create delete-authorization middleware which will check only **DELETE** requests and if Authorization http header value equals to “X-Password qwerty” - will call next middleware, in any other case - will return **401** Unauthorized.
* Create following [handlers](http://expressjs.com/en/guide/routing.html), which will be run according to the different requests:
* POST /car
* GET /car
* GET /car/{id}
* PUT /car/{id}
* DELETE /car/{id}

**POST /car**

Creates a new car (add into storage).

Response:

* Return the 201 status after successfully handling and return created car in body
* Return the 409 status with message, if resource already exists (e.g. {"message": "Car already exists."} )

**GET /car**

Returns list of cars (whole data from storage).

Response:

* Return the 200 status after successfully handling

**GET /car/{id}**

Returns a car data by id.

Response:

* Return the 200 status with the body after successfully handling
* Return the 404 status, if car with such id has not been found

**PUT /car/{id}**

Update car data by id.

Response:

* Return the 200 with the new car info in body
* Return the 404, if car with such id has not been found

**DELETE /car/{id}**

Removes car by id.

Response:

* Return the 200 status, if car has been removed. Body must contain the message about successfully handling. (e.g.{"message": "The car has been successfully removed"})
* Return the 404 status, if car with such id has not been found
* Add additional middleware to process POST body incoming values. Use body-parser.
* You could use the [fs](https://nodejs.org/api/fs.html#fs_fs_readsync_fd_buffer_offset_length_position) module for working with files.

## RESTRICTIONS

* tests/\* files shouldn't be touched at all
* node\_modules directory should be added into .gitignore

## BEFORE SUBMIT

* Test API endpoints with Postman.
* Run **npm test** command to check API handlers (it requires server to be up).

## SUBMIT

* The folder should be uploaded to GitHub repository 'fl-10' into master branch

## USEFUL LINKS

* [http://expressjs.com](http://expressjs.com/)
* <https://www.getpostman.com/>
* <https://nodejs.org/api/>
* <https://mochajs.org/>