**1. Explore the basics of APIs**

Create an API that responds to each request with:

* status: 200
* text: the method of request (e.g., GET, POST, PUT, DELETE)

Hints:

* handle all requests:

app.use((req, res) => {

// here you put the code

});

* respond with a status and a message: res.status(200).send('Hello World');
* access the method of request: req.method

**2. Create a sum API**

Create an API that exposes the route **/sum** on the verb (method) **GET** and receive two numbers (a and b) through the query string. It returns to the client:

If a and b are both valid numbers:

* status: 200
* text: value of sum

Otherwise:

* status: 400
* text: invalid operands

Hints:

* handle get on route /sum:

app.get('/sum',(req, res) => {

// handle the request

});

* send data through query string: http://localhost:8080/sum?a=1&b=2
* take data from query string: req.query

**3. Upgrade the heroes API**

* Run the 5-api-server-crd example and understand the roles of server and client:
  + copy the 5-api-server-crd into the /dev/projects/
  + with conemu, go to the /dev/projects/5-api-server-crd/ and run >npm install
  + run the server with: nodemon server.js
* Modify the app so that each hero to be represented by name (string) and level (string).
* Modify the app so that heroes to be represented in a table.

**4. Modularization of the API**

* Use the example form lab-06 to split the server.js into 3 files: server.js, model.js, and repository.js.

**5.** **From in-memory to file repository**

* Store the heroes into repository.json file. Each time a read is performed, get heroes from the file and store them into the heroes array.
* Each time an operation that changes the data (create, delete) is performed, overwrite the repository.json file by saving the updated array into the file.

**6. Add the Update option with the help of PUT method**