# Lab 11: Create a basic Web API

# using Express & MongoDB

**IDE: You need to work with a local IDE like VScode for this lab.**

**If mongodb is installed in your machine skip part A and go to part B**

1. **First part: prepare the project**
2. Connect to <https://www.mongodb.com/> and create an account (Try free).
3. Create a new database “company”, choose the free option.
4. Choose the “Username and Password” authentication method and set your database username and password. Add the 0.0.0.0 ip address (any ip will be able to connect to your database) and click on “finish”.
5. Click on “Database” then click on the “connect” button. Select “Connect your application”, you will obtain a connection string that we will copy into our .env file in our project.
6. **Second part: prepare your web server (node/express)**
7. Create a new folder on your computer for your new project. Open this folder with VSCode.
8. Initialize your folder as a node application (npm init -y). Then, install express, dotenv and mongoose.
9. Create a new file (.env) in the root of your project and create a key that will hold the PORT and a second one for MONGO\_URI.

The connection string:

mongodb+srv://yourUsername:yourPassword@se371.p9faam0.mongodb.net/CompaniesDB

If you have a local mongoDB instance, you can use:

Mongodb://127.0.0.1:12017/CompaniedDB

1. Create your main application file app.js, then import express, mongoose and call the config function from dotenv to load your environment variables. Add a console.log(process.env.PORT) and execute the program to test that dotenv is working properly.
2. Create an express application using const app = express();
3. **Create the mongoose model**
4. Create a new folder “model” then create a new file in this folder and call it “company.js”. In this file, import ‘mongoose’.
5. Create a mongoose schema for the company entity:

A company has:

* code: String, required.
* name: String, required.
* address: String.
* description: String.
* capital: Number.
* owner: String.

1. In your app.js, import the company model then create an instance and try to log it to the console to test that the model works properly.
2. Connect to your database after the creation of the express app:

// Connect to database then start server

mongoose.connect(process.env.MONGO\_URI)

.then((result) => {

console.log('Connected to database...');

app.listen(process.env.PORT, "localhost", () => {

console.log(`Listening on port ${process.env.PORT}...`)

});

})

.catch((err) => {console.log(err); });

1. Create an API endpoint that will allow users of your Web API to add company objects into the database and return the \_id in the response object:

Route: (POST) '/v1/companies/code/:code/name/:name/address/:address/description/:description/capital/:capital/owner/:owner'

1. Create an API endpoint that will allow users of your Web API to retrieve all the companies from the database and return the result in the response object.

Route: (GET)

'/v1/companies/'

1. (OPTIONAL) Create a simple front-end to display all companies and a form to add a new company.

END.