

PRACTICAL LESSON PLAN

Subject: Web Applications Development
Topic: Data brokering I
Type: ☒ Programming ☐ Setup

Degree: Informatics Engineering
Professor: Noélia Correia (ncorreia@ualg.pt)
Institution: University of Algarve

GOALS

In web development you will often access and transform data from available APIs. The first step in an Extract, Transform, and Load (ETL) pipeline is to extract data from a data source. Our data source will be the {JSON} Placeholder, a free fake API for testing (see [1]). The goal is to extract data from the subdirectory `/users`, transform part of the data into a new structure, and then load the data into a new JSON file. Besides using TypeScript and the built-in fetch API, you must organize your code into modules. That is, you will have `main.ts`, `myTransformations.ts` and `myLoadDataDestination.ts`, so that in the future you have different transformation and load steps in those modules.

REQUIRED RESOURCES

You need:

- ◊ Your laptop.
- ◊ Node and npm installed.

TASKS

Extracting Data from Source

In this step the goal is to extract the data from the mentioned source, which will be transformed later. More specifically, you must:

- ◊ Extract the data from the source;
- ◊ Orchestrate the steps: extracting data, transforming it, and loading it into the destination.

The fetch returns a Promise, and you have method `json()` available to resolve the return of the Promise to a JavaScript object. So, use chaining together with `async/await` to await for the data before passing it to the Transform step. Catch also any errors thrown when using `async/await`.

Transform Data

The goal of this step is to take data and make it fit into a new format (take a look into the data at <https://jsonplaceholder.typicode.com/users> using your browser. Iterate over every record to:

- ◊ Rename one of the keys;
- ◊ Remove the last data point (of no interest to us);
- ◊ Change every email to @ualg.pt domain.

JSON-Map conversion is usefull. See <https://codingbeautydev.com/blog/javascript-convert-json-to-map/>. For the last step use regex. See <https://www.freecodecamp.org/news/regular-expressions-for-beginners/>.

Load Data to Destination

Now that you have transformed data in the ETL (Extract, Transform, Load) pipeline, it is necessary to write the data to a JSON flat file on the file system, which is the final stage of this ETL pipeline. You should:

- ◊ Write all the data at once;
- ◊ Use Promises, as this is an I/O operation.

You can use the `fs` API together with a `promisify` function in the `util` API, or use the `FS Promises` API.

MAIN REFERENCES

- [1] <https://jsonplaceholder.typicode.com/>