# **TP: Gifts Factory**

**Introduction**

You will need these tools to complete this TP:

- Node.js and NPM

- Visual Studio Code

- Docker to install and use your api easily!

**Presentation**

For this TP, you will get an API using Docker and develop a website which will (in part) call the API. You will be tested in your object-oriented programming capabilities.

**Provided files**

In addition to this subject, we will provide you:

- the PowerPoint of this course

- a zip file containing your development environment

- a backup zip containing the API, if you encounter an issue with Docker

**Configuration presentation**

The provided configuration you will use is far better than the one you used during previous TP:

* Your webpage will be opened automatically when you will launch a successful *npm start*
* It will refresh itself for every modification you make, included inside your HTML file
* You can import css inside your JS file
* You can declare properties outside methods in your classes.

We also added the allowed packages: axios, bootstrap and jquery.

**Setup**

You can set up the API using Docker with this command:

docker run -p 8081:8081 flotorel/api-tp3

In case you encounter an issue with Docker, you just need to get the API from the zip file and launch it just like you did for your first TP: use ***npm install*** and ***node ./server.js*** commands

**Project**

This project will be tougher than the two previous ones, you’ve been warned! Due date is **October, Saturday 19**! But we will add new things next week so don’t take your time… 😊

You will have to create a website to help Santa Claus (he exists…) manage is **gifts factory**. These are the features you must implement:

1. Implement different types of gifts

You have **three types of gifts** to manage:

* the **small** gift: weights 1kg, takes 0.5 seconds to be prepared
* the **normal** gift: weights 2kg, takes 1 second to be prepared
* the **big** gift: weights 5kg, takes 2 seconds to be prepared

Every time you want a gift to be prepared, you just need to tell **the** **dwarf** what type of gift he needs to prepare. **He must take the right preparation time to prepare it** (during this time he can’t do another thing…)

You don’t have to manage stock problems: Santa Claus always has enough gifts!

1. Add the gift to the sled

Remember the magic sled pulled by eight reindeers? Well it’s true! When the Dwarf has finished preparing the gift, **he places it in the sled**.

The sled can take gifts **until they weight more than 12kg in total**. If the sled is full, the dwarf will refuse to prepare it!

1. Tell the reindeers to deliver the gifts to children (and the magic happens!)

**When your sled is not empty, you can ask the reindeers to deliver the sled content**. You will not have to manage the delivering: the reindeers will deal with it!

The API will simulate the delivery of the reindeers: for every gift in the sled, they will take 0.5 seconds to deliver it (not depending on the weight). In the end, they will tell you everything was fine.

Well, sometimes it is *not* fine: they are stupid! You have 1 chance out of 5 to have a “*I am hungry!*” response from them. In this case, you must ask them again until they agree! (*stupid reindeers…*)

When the sled is away due to a delivery, the dwarf refuses to prepare new gifts because “*the sled is not here, so why would I bother?!*”

The API will listen on <http://localhost:8081> and expect a POST request with data containing all your gifts inside a *gifts* property.

Provide a User Interface which allows you to simulate the Gifts Factory and Santa Claus will be pleased! Ergonomics will matter!

**What you should NOT do in ANY case!**

**Any of these will give you an unconditional 0:**

- using any library which is not explicitly needed

- impossibility for us to install and build your project using “npm install” and “npm start”

- checking-in/sending us the node\_modules folder

**What you should send to us**

You just need to send your Website content using WeTransfer. Again: do not include your node\_modules folder!

# **Good luck! Until next week! ☺**