Simon Gomez - Microservices

TD 2 Node JS

Project setup

install nodejs init project

```
sudo apt-get install nodejs
node -v
mkdir motus
cd motus
npm init
```

Run server

Vanilla JS : no framework

```
const http = require('http');
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});
server.listen(port, () => {
  console.log(`Server running at http://localhost:${port}/`);
});
```

Express

Install express

```
npm install express
```

Run express

```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

Home Page

First API

```
mkdir data
cd data
wget https://www.freelang.com/download/misc/liste_francais.zip
unzip liste_francais.zip
rm liste_francais.zip
iconv -f ISO-8859-15 -t UTF-8 liste_francais.txt -o liste_francais_utf8.txt
```

- read the word list and store it in an array variable
- create an API /word which will return the 127e word of the list
- use your browser or `curl localhost:3000/word` to check that your API is working

Improved Algorithm

- generate a random number which will change everyday and be the same all day long
- use this number get the same word for a day
- return the word on the /word endpoint
- your algorithm does not need to be perfect, try to explain the liimitation

TIPS :

- generate a random based on time
- use modulo to get the word
- test in browser

First Page

1. setup a **static folder** using express static middleware https://expressjs.com/fr/starter/static-files.html
2. create a HTML page served by the static server
3. create a form with an input https://www.w3schools.com/html/html_forms.asp
4. on form submit, check if word match and return the result and implement a simple motus algorythm :
    - if letter is part of the word and at the right place : background color : green
    - if letter is part of the word but not a the correct place : bacground color : orange
    - else if letter is not part of the word, no specific background color

TIPS :

- word.split() method can split a word into an array of characters
    - $("#id") get an html element
    - $("#id").val() get an input value
    - $("#id").append(…) append something to a span
    - array.include(somevalue) allow to test if somevalue is part of the array
    -
    - check event prevent default for the form submission https://www.w3schools.com/jsref/event_preventdefault.asp
    - $.get('/word', (data)=> {console.log(data)}) is an easy way to do some request from the browser

Some consideration

word is simply available on the page (cheating is possible) it's easy to test every letter **let's do it server side !**

Some improvements

- add CSS
- split JS and CSS in dedicated page
- store all CSS and JS localy
- add an API to change the seed value
- improve UI
    - color
    - form control
    - button

TIPS :

- with a GET query, you can access the request param with the req.query object in JS
- with a POST query, easiest way to get the parameters will be the middleware express-json

Published with GitHub Pages