

Final Exam: NODEJS

WEP 2023

{2h}

24 MARCH 2023

EXERCICES	POINTS
Theory	20
Practice 1	20
Practice 2	25
Practice 3	35
TOTAL	100

TODO BEFORE START

- ✓ Close all applications (telegram, chrome, Facebook...)
- ✓ For **THEORY**, you are not allowed to search on internet or write code on VSCode
- ✓ For **PRACTICE** : install the NPM packages if needed, using the bellow command:

```
npm install
```

Tips:

- ✓ *to not be confused, you can open each practice folder in VSCode one by one*
- ✓ *For PRATICE, read all instruction and chose first the exercise you are the most comfortable with*

TODO BEFORE SUBMITING

WARNING: if submit is not done properly, 5 points will be withdrawn from your final score

RULE 1	Check the content of your folders
RULE 2	Remove /nodes_modules from all folders
RULE 3	Name you folder : FIRSTNAME-LASTNAME example: HIM-HEY
RULE 4	Zip your folder example: HIM-HEY.ZIP

THEORY (20 points)

Google Form:

<https://forms.gle/CCFZi6LN1T1q4vhn7>

PRACTICE 1 (20 points)

We want to display the ID card of person, with random data provided by a public API

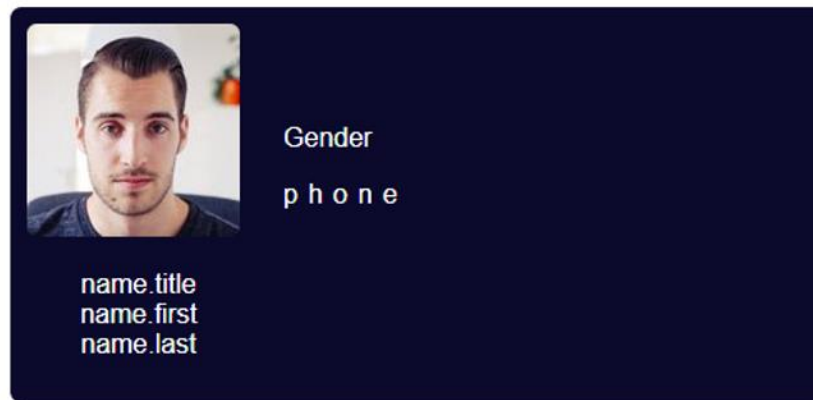
Q1.

In **main.js**, request data from the following API URL:

`https://randomuser.me/api`

Q2.

In **main.js**, display data in card following the below template:



Q3.

Change Card color depending on the person **title**:

Title	Card Color
Mr	teal
Mrs	purple
Miss	pink
<i>otherwise</i>	#0b0a2c



PRACTICE 2 (25 points)

For this practice you need to:

- ✓ Create the API with NodeJS
- ✓ Test the API with Thunder Client

A library needs to develop an application to manage its books:

- The books are stored in an array
- Each book has:
 - an id (a unique integer)
 - a title (a string)
 - the name of its author (a string)

Complete the file *routes/books_route.js* to manage the following routes

REQUEST	HTTP METHOD	PATH	REQUEST BODY	RESPONSE
Get all the books	GET	/books		The list of books <i>Example:</i> <i>[{id: 1, title: AA, author: BB}, {id: 2, title: CC, author: DD}]</i>
Create a new book	POST	/books	{ title: <i>bookTitle</i> , author: <i>authorName</i> }	The new list of books
Update one book by id	PUT	/books/:book_id	{ title: <i>bookTitle</i> , author: <i>authorName</i> }	The new list of books <i>If there is no book with this id: send a response with status 404 and message "book not found"</i>
Delete books by author (remove all the books written by the author given in query parameter)	DELETE	/books Query Parameter: <i>?author=book_author</i>		The new list of books

PRACTICE 3 – (30 points)

A product has the following properties: an **ID**, a **NAME**, and a **BRAND**

Attributes	Type	Description
id	Integer	The ID of the product
name	String	The name of the product
brand	String	The product's brand

Our application can retrieve the product NAME or BRAND by providing its ID:

Get Name

Get Brand

For example, if we have the following 2 products in our data:

id	name	brand
3	computer	Asus
4	screen	HP

Then the GET NAME button will display SCREEN for ID = 4:

Get Name

Get Brand

The product name is: screen

And the GET BRAND button will display ASUS for ID = 3:

Get Name

Get Brand

The product brand is: Asus

But for ID=5, we shall display: “No product found!”:

Get Name

Get Brand

No product found!

[SERVER SIDE]

Q1 – Write a **PRODUCTS.JSON** file with the appropriate data (create your own data).

Q2 – Complete SERVER.JS to read the file **PRODUCTS.JSON**

Q3 – Complete SERVER.JS to manage the following request:

GET <YOUR IP>:3000/product_name?id=<PRODUCT_ID>

This request shall return:

{"name": <the product's name>}

if the product is found

{"name": null}

if the product if not found

Q4 – Complete SERVER.JS to manage the following request:

GET <YOUR IP>:3000/ product_brand?id=<PRODUCT_ID>

This request shall return:

{"brand": <the product's brand>}

if the product is found

{"brand": null}

if the product if not found

[CLIENT SIDE]

Q5 – In PRODUCTS.JS, link the button GET NAME and GET BRAND to the appropriate request and display the result on the p #message.

Note: if cannot find the product, display: “no product found” (in red)

Get Name

Get Brand

The product name is: keyboard

Get Name

Get Brand

The product brand is: Dell

Get Name

Get Brand

No product found!

Get Name

Get Brand

No product found!