Prénom: Nom: Groupe:

R3.01 - dev. web - côté serveur (75 min)

Joseph AZAR

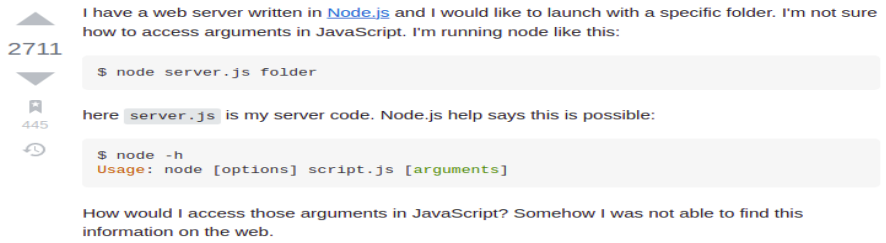
Dec 2022

Questions à choix multiple (14 points) (entourer la bonne lettre)

+0.5 pour une bonne réponse, -0.25 pour une réponse incorrecte

1. Qu'est-ce qu'un «Callback»?
   a) Le “Calback” est un équivalent asynchrone pour une fonction.
   b) Une fonction de rappel (Callback) est appelée à la fin d'une tâche donnée.
   c) Le rappel (Callback) est généralement le dernier paramètre d'une fonction
   d) Toutes ces réponses.
   e) Aucune de ces réponses.
2. En utilisant **module.exports**, vous pouvez exporter des fonctions et des objets mais pas des variables
   a) Vrai b) Faux
3. Les API de NodeJS sont par défaut :
   a) Asynchrone b) Synchrone c) Toutes ces réponses. d) Aucune de ces réponses.
4. Pour installer le module Express:
   a) $ npm install express
   b) $ node install express
   c) $ install express
   d) Aucune de ces réponses.
5. Qu'est-ce que REPL dans Node.js?
   a) Loop b) Print c) Eval d) Toutes ces réponses.
6. _____ : un mécanisme utilisé pour trouver des ressources sur Internet
   a) URL b) HTTP c) serveur Web d) REST d) Toutes ces réponses.
7. Laquelle des affirmations suivantes est vraie à propos de l'objet global **__filename**?
   a) Le __filename représente le nom de fichier du code en cours d'exécution.
   b) Le __filename représente le chemin absolu résolu du fichier de code.
   c) Toutes ces réponses.
   d) Aucune de ces réponses.
8. Il existe trois types de fonctions d'API dans Node.js: asynchrone, synchrone et RESTful.
   a) Vrai b) Faux

9. Si nous voulons obtenir la valeur de “tagId” dans cette URL: *localhost:3000/?tagId=5*, nous utilisons:
    a) req.query.tagId
    b) req.params.tagId
    c) req.queryString.tagId
    d) Aucune de ces réponses.
10. La ressource Web renvoie un contenu dynamique et nous devons envoyer un ou plusieurs paramètres au serveur, nous pouvons donc utiliser :
    a) Query Strings b) parameters c) Toutes ces réponses. d) Aucune de ces réponses.
11. Lesquels des éléments suivants ne sont pas des attributs de **package.json**:
    a) dependencies
    b) devDependencies
    c) author
    d) Tous les éléments ci-dessus sont des attributs de **package.json.**
12. Que signifie REST?
    a) Resource Efficient State Transfer
    b) Real Elegant State Transfer
    c) Resource Elegant State Transfer
    d) Aucune de ces réponses.
13. Lequel des moteurs de templates **(template engine)** suivants peut être utilisé avec Node.js?
    a) Express b) Pino c) Handlebars d) Toutes ces réponses. d) Aucune de ces réponses.
14. Ce pauvre codeur pose cette question sur StackOverflow:



    Pouvez-vous lui dire comment accéder aux arguments dans Node.js?
    a) utiliser process.argc
    b) utiliser process.argv
    c) vous ne pouvez pas passer d'arguments à Node.JS, vous pouvez le faire en C++ et Java pas en JavaScript.
    d) Aucune de ces réponses.
15. NodeJS est un :
    a) langage de programmation b) environnement d'exécution c) Toutes ces réponses.
    d) Aucune de ces réponses.

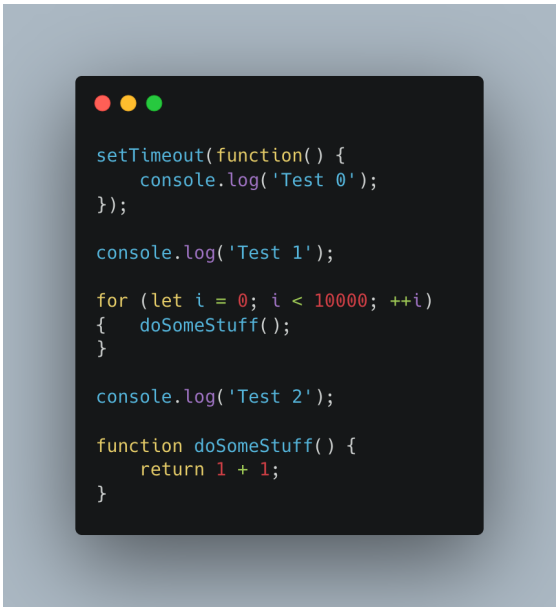
16. __dirname, setTimeout(cb, ms) et Process sont tous des objets globaux:
    a) Vrai b) Faux
17. La version LTS actuelle de NodeJS est:
    a) 14 b) 16 c) 18 d) Aucune de ces réponses.
18. Les variables peuvent être transmises au middleware suivant en utilisant:
    a) req (request) b) res (response) c) req et res d) Aucune de ces réponses.
19. Comment NodeJS empêche le code de blocage (blocking code)?
    a) Async/Await b) Promises c) Callbacks d) Toutes ces réponses.
20. Est-ce une route valide:
    http://localhost:3000/users?username=joseph&username=alex&username=trump ?
    a) Oui b) Non
21. Le code ci-dessous est:

```
var callback = function (err, data) {
  if (err) return console.error(err);
  console.log(data);
  sleep(10000);
};
fs.readFile('test.txt', callback);
console.log("done!");
```

    a) Code bloquant b) Code non bloquant c) Code erroné
22. Dans lequel des cas suivants, il **n'**est **pas** conseillé d'utiliser Node.js?
    a) Applications d'une seule page.
    b) Applications basées sur des API JSON.
    c) Applications en temps réel à forte intensité de données (Data Intensive Realtime Applications).
    d) Aucune de ces réponses.
23. Node utilise le moteur:
    a) Chrome V8 b) Libuv c) Event Loop d) Toutes ces réponses.
24. L'application Node.js s'exécute sur
    a) Plusieurs threads b) Thread unique c) Plusieurs processus d) Processus unique
25. Comment puis-je vérifier de manière synchrone, à l'aide de node.js, si un fichier ou un répertoire existe?
    a) Node est par défaut asynchrone, il ne prend pas en charge les fonctions synchrones
    b) await fs.readFile(“nom de fichier”, cb)
    c) Aucune de ces réponses.
26. Par défaut, npm installe toute dépendance avec _____
    a) le mode local b) le mode global
27. Un Calback qui prend trop de temps à s'exécuter peut bloquer la boucle d'événement (Event Loop).
    a) Vrai b) Faux
28. Node.js est écrit en _____
    a) C++ b) JavaScript c) Toutes ces réponses. d) Aucune de ces réponses.

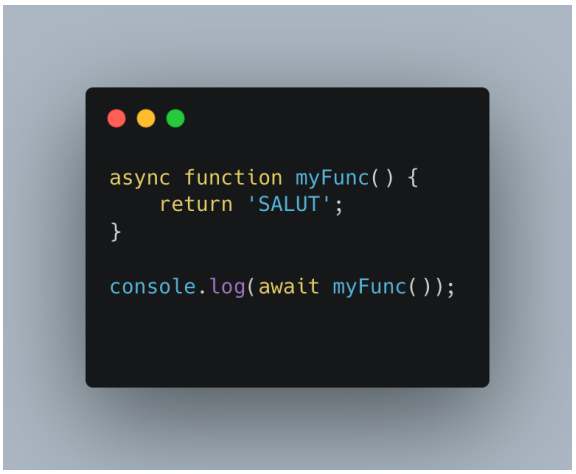
Javascript asynchrone et NodeJS (2.5 points)

1. Quelle est la sortie de ce code ? Que peut-on conclure de cette sortie ?



```
setTimeout(function() {
    console.log('Test 0');
});

console.log('Test 1');

for (let i = 0; i < 10000; ++i)
{   doSomeStuff();
}

console.log('Test 2');

function doSomeStuff() {
    return 1 + 1;
}
```

2. Quel est le nom du problème que vous pouvez rencontrer lors de l'utilisation des rappels ? Comment pouvez-vous résoudre ce problème ?
3. Quelle est la sortie de ce code ? pourquoi pensez-vous que nous avons obtenu cette sortie ?



```
async function myFunc() {
    return 'SALUT';
}

console.log(await myFunc());
```

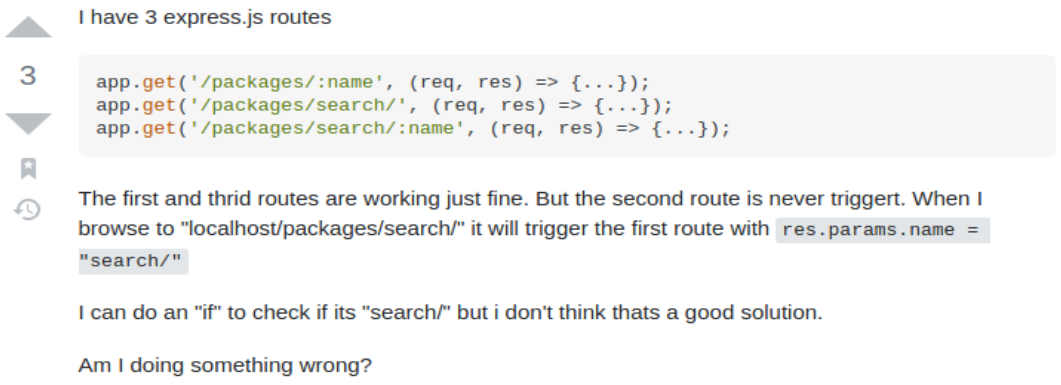
4. Citer un avantage de l'architecture Single Threaded Event Loop Model.

Questions sur NodeJS et le backend (6 points)

1. Donnez un cas d'utilisation dans lequel NodeJS n'est pas le framework côté serveur optimal à utiliser et dites pourquoi.
2. Mentionnez quelles sont les méthodes HTTP prises en charge par REST?
3. Qu'est-ce qui n'est pas correct dans le code suivant?

```
const customersController = require("../controllers/customers.controller");
const express = require("express");
let router = express.Router();
router.get( path: "/add", handlers: (req :Request<P, ResBody, ReqBody, ReqQuery, Locals> ,res :Response<ResBody, Locals> )=>{
    let firstname = req.query.name;
    let lastname= req.query.lastName;
    try{
        customersController.addCustomers(firstname,lastname);
        res.json( body: {message:"Success"});
    }catch (e) {
        res.json( body: {message:"Error"});
    }
});
```

4. Ce pauvre codeur se pose la question suivante:



I have 3 express.js routes

3

```
app.get('/packages/:name', (req, res) => {...});
app.get('/packages/search/', (req, res) => {...});
app.get('/packages/search/:name', (req, res) => {...});
```

The first and thrid routes are working just fine. But the second route is never triggert. When I browse to "localhost/packages/search/" it will trigger the first route with `res.params.name = "search/"`

I can do an "if" to check if its "search/" but i don't think thats a good solution.

Am I doing something wrong?

Son deuxième route ne fonctionne pas. Répondez-lui quel est le problème et comment peut-il le résoudre?

5. Nous voulons lire le fichier "monfichier.txt" de manière asynchrone en utilisant le modèle de Callback “erreur en premier” (error-first). Écrivez la fonction "ErrorFirstCallback" ci-dessous:

```
const fs = require("fs");
const file = "monfichier.txt";
// Error first callback
const ErrorFirstCallback = ... // Compléter cette fonction
fs.readFile(file, ErrorFirstCallback);
```

6. Lors de l'exécution du code ci-dessous, nous obtenons cette erreur: “«Express.js Routing error: Can't set headers after they are sent»”. (Erreur de routage Express.js: Impossible de définir les en-têtes après leur envoi».)

   Pourquoi pensez-vous que nous obtenons cette erreur et comment la corrigeons-nous?

```
module.exports = function(router) {
    router.route('/posts')
        .post(function(req, res) {
            let name = req.body.firstName;
            saveClient(name,function(err) {
                if (err)
                    res.send(err);
                res.json({ message: 'post created!' });
            });
        });
}
```

7. Quelle est la différence entre require(x) et import x (ES6) dans NodeJS ?
8. Pouvez-vous citer ce qui constitue les composants de base d'une requête HTTP et d'une réponse HTTP?
9. Quelle réponse obtenons-nous lorsque nous envoyons cette requête:

   GET: **http://localhost:3000/api/albums/artwork?msg=1**

```
app.get('/api/albums', (req,res)=>{res.send("A")});
app.get('/api/albums/:id', (req,res)=>{res.send("B")});
app.post('/api/albums/artwork', (req,res)=>{res.send("C")});
app.get('/api/albums/artwork', (req,res)=>{res.send("D")});
```

10. Pouvez-vous aider ce pauvre codeur à résoudre son problème. Il veut accéder aux champs du formulaire POST. Dites-lui où est l'erreur et comment la résoudre.

854

217

Here is my simple form:

```
<form id="loginformA" action="userlogin" method="post">
    <div>
        <label for="email">Email: </label>
        <input type="text" id="email" name="email"></input>
    </div>
<input type="submit" value="Submit"></input>
</form>
```

Here is my Express.js/Node.js code:

```
app.post('/userlogin', function(sReq, sRes){
    var email = sReq.query.email.;
}
```

I tried `sReq.query.email` or `sReq.query['email']` or `sReq.params['email']`, etc. None of them work. They all return `undefined`.

When I change to a Get call, it works, so .. any idea?

11. Citez une approche qui peut être utilisée pour authentifier les utilisateurs et un module qui peut être utilisé pour hacher les mots de passe.
12. Pouvez-vous donner un module NodeJS pour les tests unitaires ?

API REST (2.5 points)

1. Fournissez un exemple spécifique où il serait plus approprié de renvoyer une erreur 500 au lieu d'une erreur 40X dans une API REST.

2. Il existe quatre principes directeurs suggérés par Roy Fielding qui constituent les contraintes nécessaires pour satisfaire l'interface uniforme. Parmi les quatre principes figure les Messages autodescriptifs (Self-descriptive messages). Que signifient les messages autodescriptifs ? pouvez-vous citer deux autres principes (sans les décrire) ?

3. Nous souhaitons créer une API REST qui traite les ressources suivantes :
   a. **Books**: C'est l'entité principale et elle possède toutes les propriétés requises pour identifier un livre et le localiser dans un magasin spécifique.
   b. **Authors**: Cette ressource est étroitement liée à la ressource d'un livre car elle répertorie l'auteur de chaque livre dans un magasin.
   c. **Stores**: Informations de base sur chaque magasin, y compris l'adresse, les employés, etc.
   d. **Employees:** les informations sur les employés, les données de contact et d'autres propriétés internes qui peuvent être utiles pour un utilisateur de type administrateur.

   Complétez le tableau suivant (la première ligne est un exemple) :

Endpoint/Route	Méthode HTTP	Description
/books	GET	répertorie et recherche tous les livres
		met à jour les informations d'un livre
		renvoie le ou les auteurs d'un livre spécifique
		ajoute un nouvel auteur
		retourne les données sur un magasin spécifique
		renvoie une liste des employés travaillant dans un magasin spécifique
		renvoie une liste de livres écrits par un auteur spécifique