

## Bonjour, Demo User #14

**CONFIDENTIEL**  
**POUR ADMINISTRATION EEMI**  
**NE PAS DIFFUSER**

Ceci est un examen **individuel** noté sur **20 points**. Durée: **2 heures**.

### Consignes et informations:

- Vos réponses seront évaluées de manière automatique => respecter **à la lettre** les consignes et **valeurs fournies** dans l'énoncé, lors de la rédaction de votre code.
- Tous documents autorisés. L'usage de tout moyen de communication (ex: email, chat, SMS, discussion via forums) et toute fraude détectée engendrera une **note de 0/20**.
- Pensez à rendre votre copie **avant** l'heure de fin d'examen. Si vous sortez de la salle avant l'heure de la fin d'examen, montrez l'écran final de rendu au surveillant puis laissez votre ordinateur dans la salle, écran fermé.

### Quelques conseils:

- Lisez tous les énoncés en début d'examen.
- Pensez à tester votre code, puis à corriger l'indentation avant de le saisir ici.
- Les exercices ne contiennent aucun piège. Ils peuvent chacun être implémentés en moins de 10 lignes de code.

Gardez votre calme, tout va bien se passer ! 🙌

En cas de **dysfonctionnement**: utilisez le mode "navigation privée" de Google Chrome pour accéder à la page, assurez-vous que les cookies et cookies tiers sont activés dans les paramètres du navigateur, rafraîchissez la page à l'aide de `Cmd-R` ou `Ctrl-R`, puis reconnectez-vous à l'aide de votre compte Google de l'EEMI. Prévenez votre surveillant seulement si vous ne parvenez toujours pas à accéder à l'examen après avoir effectué ces manipulations.



## QCM (1 point par bonne réponse)

### Question 1

J'ai créé une API avec Node.js et l'ai mise en production sur Heroku. Où s'exécute le code permettant de répondre aux requêtes reçues par cette API ?

- ☐ Dans le navigateur
- ☐ Dans la commande curl
- ☐ Sur un des serveurs de Heroku
- ☐ Sur GitLab

**CONFIDENTIEL**  
**POUR ADMINISTRATION EEMI**  
**NE PAS DIFFUSER**

## Question 2

Je viens de "cloner" un projet Node.js depuis GitHub. Comment installer les dépendances de ce projet ?

- ☐ npm start
- ☐ npm install
- ☐ npm init
- ☐ npm install express --save

## Question 3

Laquelle de ces phrases est CORRECTE ?

- ☐ MongoDB est une base de données SQL constituée de collections et de documents
- ☐ MongoDB est une base de données SQL constituée de tables et de lignes
- ☐ MongoDB est une base de données NoSQL constituée de collections et de documents
- ☐ MongoDB est une base de données NoSQL constituée de tables et de lignes

## Question 4

À quoi sert Express.js ?

- ☐ à faciliter la création d'un serveur HTTP
- ☐ à accélérer la vitesse de traitement des requêtes HTTP
- ☐ à gérer des requêtes asynchrones comme si elles étaient synchrones
- ☐ à mettre une API en production

### Question 5

Laquelle de ces phrases est INCORRECTE ?

- ☐ await permet d'attendre qu'une Promise soit résolue
- ☐ await ne peut être utilisé que depuis une fonction async
- ☐ await permet d'exécuter une fonction synchrone comme si elle était asynchrone
- ☐ sans await, deux fonctions asynchrones peuvent être exécutées en parallèle

## Exercices de code (5 points par exercice)

### Question 6

Ré-écrire ce code de manière à ce qu'il utilise `async` et `await`, au lieu de `then()` et `catch()`. Les erreurs doivent être correctement interceptées.

```
const MongoClient = require('mongodb').MongoClient;
MongoClient.connect('mongodb://localhost:27017/test')
  .then(function (db) {
    return db.collection('dates').find().toArray();
  })
  .then(function (dates) {
    console.log('dates:', dates);
  })
  .catch(function (err) {
    console.error('erreur:', err)
  });
```

Saisissez votre code Javascript ici

---

**CONFIDENTIEL**  
**POUR ADMINISTRATION EEMI**  
**NE PAS DIFFUSER**

## Question 7

Le fichier `index.js` contient le code suivant:

```
const express = require('express');
const app = express();

app.listen(process.env.PORT || 3000);
```

Quelles lignes de code faut-il ajouter à ce fichier pour que `curl http://localhost:3000/bonjour` affiche "Hello", une fois qu'on aura exécuté ce programme avec `node index.js` ?

Saisissez votre code Javascript ici

---

## Question 8

L'objectif est d'écrire un programme Node.js permettant d'afficher dans la sortie standard (c.a.d. en utilisant `console.log()`) l'adresse email de trois personnes dont les données seront à récupérer en JSON depuis les URLs suivantes:

- <https://js-jsonplaceholder.herokuapp.com/users/1>
- <https://js-jsonplaceholder.herokuapp.com/users/2>
- <https://js-jsonplaceholder.herokuapp.com/users/3>

Seules les adresses email doivent être affichées. Et l'affichage de ces adresses doit respecter l'ordre des URLs fournies ci-dessus. C'est à dire que l'adresse email de la personne `1` doit être affichée en premier, et ainsi de suite.

Votre programme devra utiliser le module `node-fetch` pour effectuer les requêtes. Voici le contenu actuel du programme:

```
const fetch = require('node-fetch');
const urlsToFetch = [
  'https://js-jsonplaceholder.herokuapp.com/users/1',
  'https://js-jsonplaceholder.herokuapp.com/users/2',
  'https://js-jsonplaceholder.herokuapp.com/users/3'
];
```

Quelles lignes de code faut-il ajouter pour que ce programme affiche l'adresse email de ces utilisateurs, quand on l'exécutera avec `node` .

Saisissez votre code Javascript ici

---

**CONFIDENTIEL**  
**POUR ADMINISTRATION EEMI**  
**NE PAS DIFFUSER**

## Rendu de copie

✓ RENDRE LA COPIE