+1/1/60+

# INF3054L – LIFWEB — Printemps 2023–2024

Ce contrôle est un QCM corrigé **automatiquement** en tout ou partie. Répondre au **stylo bille foncé** en **noircissant complètement** la case correspondante sur **la feuille de réponse fournie**. Les questions **avec** le symbole ♣ peuvent présenter *zéro, une ou plusieurs bonnes réponses*. Ces questions sont notées sur 2 points avec -1 point par réponse fausse (0 minimum par question). Les questions **sans** le symbole ♣ ont une *unique bonne réponse*. Ces questions sont notées sur 2 points avec 2 points si la réponse est correcte, 0 s'il n'y a pas de réponse et -0.5 si la réponse est fausse. Le barème des questions ouvertes est précisé sur chacune.

---

## 1 Programmation JavaScript générale

On considère l'extrait du code du TP4-b *GitHub Verifier* ci-après.

```
1  const $fileSelector = document.querySelector("#file-selector");
2  let controller;
3
4  async function checkLinkAlive(link, signal) {
5    /* code omis - (A) */
6    return { ...link, ok: true };
7  }
8
9  function displayLink(link) {  /* code omis */ }
10
11 async function downloadAndCheck() {
12   const url = new URL(`data/${$fileSelector.value}`);
13   try {
14     const response = await fetch(url);
15     if (!response.ok) {
16       throw new Error(`${response.status} (${response.statusText})`);
17     }
18     const links = await response.json();
19     /* code omis - (B) */
20     controller = new AbortController();
21     const results = await Promise.all(links.map((link) => checkLinkAlive(
           link, controller.signal)));
22     /* code omis - (C) */
23   } catch {
24     /* code omis - (D) */
25   }
26 }
```



**Question 1** **(/4)** Expliquer ce que fait *l'opérateur virgule* `(,)` (*comma operator*) et donner un exemple d'utilisation. *Rappel* : il n'a rien avoir avec la notion de paire ou de tuple.

**Question 2** **(/2)** Quel est le rôle de l'instruction `await` dans une fonction asynchrone ?

- A Déclencher un événement interrompant l'exécution de la fonction.
- B Exécuter une fonction de rappel (*callback*) après un certain délai.
- C Attendre la résolution d'une promesse avant de continuer l'exécution.
- D Renvoyer une promesse rejetée en cas d'erreur.

**Question 3** **(/4)** Expliquer ce que calcule la fonction `const f = (x, y)=> x.reduceRight((q, p) => q.then(p), Promise.resolve(y));` et donner un exemple d'utilisation.

**Question 4** **(/2)** Quelle est la raison de l'utilisation de la variable `controller` dans la fonction `downloadAndCheck` ?

- A Pour stocker les liens.
- B Pour annuler les requêtes HTTP.
- C Pour gérer la vérification des liens.
- D Pour contrôler l'accès aux ressources.

**Question 5** **(/2)** À quel endroit de la fonction `downloadAndCheck` la fonction `displayLink` devrait-elle être utilisée ? Les références `(A)`, `(B)`, `(C)` et `(D)` sont identifiés dans les commentaires.

- A `(D)`
- B `(A)`
- C Ailleurs
- D `(C)`

**Question 6** **(/2)** Que faut-il faire à la référence `(B)` identifiée dans les commentaires du code ?

- A Rien
- B Mettre-à-jour l'interface en cas d'exception.
- C Mettre-à-jour l'interface en activant/désactivant des boutons.
- D Mettre-à-jour l'interface avec les résultats de la vérification des liens.

**Question 7** **(/4)** La fonction `downloadAndCheck` emploi une stratégie appellée *fail-first*. Expliquer brièvement en quoi consiste cette stratégie en illustrant sur cet exemple (donner les lignes du code).

**Question 8** **(/2)** On souhaite calculer et afficher le nombre total d'étoiles (*stars*) des dépôts dans l'extrait de code. Quel est l'endroit le plus approprié pour le faire ?

- A Dans la fonction `downloadAndCheck`.
- B À la racine du fichier.
- C Dans la fonction `displayLink`.
- D Dans la fonction `checkLinkAlive`.

**Question 9** **(/4)** Pourquoi teste-t-on `response.ok` dans la fonction `downloadAndCheck` ? Préciserla différence entre ce test et rattraper une exception levée par `fetch (url)` dans un `catch`.

**Question 10** **(/4)** Si on remplace `await Promise.all(links.map((link)=> checkLinkAlive (link, controller.signal)));` par `const results = []; for (const p of links.map((link)=> checkLinkAlive(link, controller.signal))){ results.push(await p)},` le traitement devient-il séquentiel ? Justifier.

**Question 11** **(/6)** Réécrire la fonction `downloadAndCheck` en utilisant explicitement les promesses, *sans utiliser* `async/await`. Pour simplifier, on ne demande pas de gérer le `AbortController()`.

## 2 Programmation Web en Node.js

On rappelle que les principales méthodes HTTP, appelées aussi verbes HTTP, utilisées pour effectuer ces opérations sont `POST`, `GET`, `PUT` et `DELETE`. Dans cette partie, on s'intéresse à la programmation Web en Node.js et à l'architecture utilisée dans les TP5 (*DevOps*) et TP6 (*URLs Shortener*). On donne à cet effet un extrait du code du routeur pour l'API de raccourcissement d'URL ci-après.

```
1  import Boom from "@hapi/boom";
2
3  async function validateShortenedLink(request, h) {
4    const { short } = request.params;
5    const link = await request.server.app.db.getLinkByShort(short);
6    if (link === undefined) {
7      throw Boom.notFound(`Shortened link ${short} not found`);
8    }
9    return link;
10 }
11
12 export default {
13   register: async function (server, _options) {
14     const { db } = server.app;
15     const checkLinkObject = {
16       pre: [{ method: validateShortenedLink, assign: "link" }],
17     };
18
19     server.route({
20       method: "GET",
21       path: "/{short}",
22       handler: async (request, h) => {
23         server.log("debug", `Redirecting to ${request.pre.link.long}`);
24         await db.visitLink(request.pre.link.short);
25         return h.redirect(request.pre.link.long);
26       },
27       options: checkLinkObject,
28     });
29
30     /* ensuite, les autres routes sont configurées */
31   },
32 };
```



**Question 12 ♣** **(/2)** Pour quelles raisons a-t-on choisi d'utiliser une base de données dans le projet Node.js *URL Shortener* ?

- A Pour son intégration native avec Hapi.
- B Pour assurer l'intégrité des données.
- C Pour pouvoir utiliser SQL.
- D Pour partager l'état à plusieurs instances.

**Question 13** **(/2)** Quel est le principal avantage de Node.js en termes de montée en charge dans le contexte de la programmation Web côté serveur ?

- A Les opérations I/O non bloquantes.
- B L'équilibrage automatique de la charge.
- C Le moteur JavaScript V8.
- D La prise en charge native de SSH.

**Question 14** **(/2)** Est-il recommandé que le compte utilisé pour le déploiement, `gitlabci` dans les TP5 et TP6, soit `sudoer` ?

- A C'est impossible.
- B Ça dépend de l'application.
- C Non.
- D Oui.

**Question 15** (/4) Expliquer pourquoi on doit utiliser l'option `verify` quand on teste une API Web déployée sur une des VMs comme dans les TPs 5 et 6, par exemple dans la commande `https --verify no https://api.lifweb.os.univ-lyon1.fr`.

**Question 16** (/4) Donner deux avantages de la stratégie de rendu-côté client (*Client-Side Rendering – CSR*) sur celle côté serveur (*Server-Side Rendering – SSR*).

**Question 17** ♣ Quel est l'intérêt des fichiers `.env` dans un projet ?

- [A] N'a pas d'avantage, car toutes les configurations peuvent être en dur dans la source.
- [B] Facilite la gestion des variables d'environnement de chaque instance.
- [C] Permet de séparer la configuration de l'application de son code source.
- [D] Permet de rendre publique la configuration dans le projet Git.

**Question 18** (/4) Quel est l'intérêt d'utiliser un serveur HTTP comme Nginx en amont de l'application Node.js, pourquoi ne pas utiliser directement Node.js sans intermédiaire ? Donner deux arguments.

**Question 19** (/2) Quand une fonctionnalité n'est pas implémentée, on renvoie une réponse HTTP *Not Implemented*. Quel est le code HTTP associé ?

- [A] 500
- [B] 501
- [C] 502
- [D] 400

**Question 20** (/2) Quel est le nom du module Hapi utilisé pour valider les données entrantes dans les requêtes HTTP du TP6 ?

- [A] Inert.
- [B] Il n'y en a pas.
- [C] Joi.
- [D] Boom.

**Question 21** (/4) Dans la route `GET /{short}` de l'extrait de code, on enregistre un objet `checkLinkObject` dans le paramètre optionel `pre`. Justifier ce cas d'usage par deux arguments.

**Question 22** (/4) Dans la route `GET /{short}` de l'extrait de code, identifier les éléments du modèle MVC (*Model-View-Controller*) correspondants à chaque partie du code. Autrement di, préciser ce qui relève de la vue, du contrôleur et du modèle.

**Question 23** (/8) On considère l'API Web d'une application *TODO List* permettant aux utilisateurs de gérer leurs listes de tâches. L'API doit permettre aux utilisateurs de lister, créer, lire, mettre-à-jour et supprimer des tâches de leur liste. Proposer une API Web en utilisant les verbes HTTP appropriés et les chemin associés pour ces cinq routes.

**Question 24** (/4) Dans l'application *TODO List*, on souhaite ajouter une fonctionnalité d'authentification pour les utilisateurs. On souhaite utiliser la même stratégie que dans le challenge CTF. Expliquer brièvement comment on peut implémenter cette fonctionnalité en Hapi, en précisant comment la clef d'API est transmise et à quel moment du cycle de vie de la requête HTTP cette information est utilisée.



# Feuille de réponses à compléter

0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9

← codez votre numéro d'étudiant-e à *8 chiffres* et reportez-le ci-dessous par sécurité.

.....



**Question 1 :** F I PJ J

**Question 2 :** A B C D

**Question 3 :** F I PJ J

**Question 4 :** A B C D

**Question 5 :** A B C D

**Question 6 :** A B C D

**Question 7 :** F I PJ J

**Question 8 :** A B C D

**Question 9 :** F I PJ J

**Question 10 :** F I PJ J

**Question 11 :** F I PJ J

**Question 12 :** A B C D

**Question 13 :** A B C D

**Question 14 :** A B C D

**Question 15 :** F I PJ J

**Question 16 :** F I PJ J

**Question 17 :** A B C D

**Question 18 :** F I PJ J

**Question 19 :** A B C D

**Question 20 :** A B C D

**Question 21 :** F I PJ J

**Question 22 :** F I PJ J

**Question 23 :** F I PJ J

**Question 24 :** F I PJ J