Correction examen HMIN103 - Données du web - 17/01/2020
Partie Pierre Pompidor

Quel est le problème de ce fragment de code (hormis l'éventuelle homonymie) ? (1.5 points)

```
app.get("/panier/:nom/:prenom", (req, res) => {
  db.collection("membres").find({"nom":req.params.nom, "prenom":req.params.prenom})
    .toArray((err, documents) => { let panier = [];
       db.collection("paniers").find({"email":documents[0].email}).toArray((err, documents2) => {
         panier = documents2[0].produits; });
       res.setHeader("Content-Type", "application/json");
       res.end(JSON.stringify(panier));
  });
});
```

Le problème est que, dû à l'asynchronisme de JavaScript, la liste *panier* a de fortes probabilités d'être renvoyée avant que le second appel à la base de données ait fourni un résultat.

Le fait que la variable panier soit initialisée en tant que liste puis qu'elle prenne directement une valeur n'est pas un problème en soit (cette intialisation étant un moyen d'indiquer dans le code son type).

Que résoud ce code et que se passe-t-il si l'instruction next() est oubliée ? (1.5 points)

```
app.use(function (req, res, next) {
    res.setHeader('Access-Control-Allow-Origin', '*');
    res.setHeader('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE');
    next();
});
```

Ce code résoud les problèmes de CORS (Cross-Origin Resource Sharing) qui surviennent quand un code chargé sur le client accède à des données provenant d'un serveur qui n'est pas celui qui a fourni ce code au client. Si l'instruction *next()* est oubliée, l'entête HTTP du résultat est bien modifiée, mais la main n'est pas redonnée à la méthode qui renvoie le résultat.

Décrivez (en quelques mots à chaque fois) les trois manières par lesquelles les objets sont indirectement instanciés dans une application Angular ? (3 points)

- via l'utilisation du sélecteur d'un composant
- via l'invocation d'un composant par une route
- via une injection de dépendance (ce qui revient au même que déclarer un service comme provider)

Communication composant père vers composant fils (2 points)

Soit cette balise intégrant un composant "fils" dans le template d'un composant "père" :

```
<compteur secondes="{{nombresSecondes}}" />
```

Ecrivez le fragment de code qui dans la classe du composant fils permet de récupérer la valeur de l'attribut *secondes*.

```
@Input() secondes: number;
```

Voici le code de la classe AuthentificationService vue en cours ? (2 points)

```
private user:Subject<string> = new BehaviorSubject<string>(undefined);
constructor(private http: HttpClient) { }
getUser() { return this.user; }
connect(data: string) { this.user.next(data); }
disconnect() { this.user.next(null); }
verificationConnexion(identifiants): Observable<any> {
  return this.http.post('http://localhost:8888/membre/connexion',JSON.stringify(identifiants),options);
}
```

- *undefined* pourrait-il être remplacé par autre chose ?
  oui mais seulement par "" (cette valeur correspondant à l'adresse mail de l'utilisateur connecté, et au lancement de l'application, personne n'est connectée par défaut)
- A quels endroits de l'application, l'observable créé dans cette classe va-t-il être utilisé ?
  par tous les composants ayant besoin de savoir si l'internaute est identifié (et notamment dans leurs templates)
  (et bien sûr la valeur de cet observable est modifiée lors de la connexion - et donc de la soumission du formulaire de connexion - ou de la déconnexion de l'internaute).