

## TP 1 (Travail individuel)

Distribué en classe le jeudi 27 octobre 2020  
**Date de remise : le jeudi 12 novembre 2020**

### API REST pour la gestion de films

#### Objectif pédagogique

Ce travail pratique porte sur la programmation orientée objet en PHP. L'étudiant doit démontrer sa capacité à appliquer des concepts de programmation orienté objet vus en classe dans le cadre du développement d'une interface de programmation accédant à une base de données via une librairie de type CRUD.

#### Ce qu'il faut accomplir

Vous développerez un API de gestion d'un inventaire de films qui fonctionnera avec une base de données MySQL. Cet API devra répondre exactement aux spécifications fournies : la correction sera entre autres effectuée en vérifiant qu'une application client conforme aux spécifications pourra consommer le JSON généré par l'API adéquatement.

#### Besoins en données :

- Une table film, sachant qu'un film doit contenir un ID, un titre, une durée, une année, un genre et un id de réalisateur qui est une clé étrangère vers la table Réalisateurs.
- Une table réalisateurs, sachant qu'un réalisateur a un ID, un prénom et un nom.

#### Développement :

##### a) **CRUD : Films et réalisateurs (10 points)**

Vous devez concevoir les deux tables nécessaires (Films et Réalisateurs) et développer un CRUD pour ces deux tables à partir de la classe abstraite TableBase vue en classe. Vous devez aussi fournir des tests unitaires pour vous assurer du bon fonctionnement de ces classes avant de passer à l'autre étape.

##### b) **API REST (14 points)**

Selon les spécifications de l'API REST fournies dans les notes de cours, vous devez développer un API REST pour chacune des tables. Vous devez utiliser POSTMAN (<https://www.postman.com/>) pour tester cet API et vous assurer que pour chaque type de requête, vous recevez le **JSON correctement formé**, ou, dans le cas d'une erreur, le **code d'erreur HTTP approprié**. Attention de respecter les standards présentés dans les notes de cours, sinon mon application de test ne fonctionnera pas!

#### Spécifications supplémentaires pour l'API (très important):

- Le répertoire de votre projet devrait s'appeler **APIFilms**, avec ces lettres en majuscules.
- Le nom du fichier contenant votre API devrait être **rest\_film.php** pour l'API des films et **rest\_realisateur.php** pour l'API des réalisateurs. Le fichier .htaccess doit aussi être modifié et mis dans votre répertoire pour que le tout fonctionne avec des URLs propres.
- Le JSON généré doit **absolument** suivre le format suivant.

Pour les **films** :

```
{
  "id" :    "1",
  "titre": "Jaws",
  "duree": "2h04",
  "annee": 1975,
  "idRealisateur" : 1,
  "genre": "Suspense"
}
```

Pour les **réalisateurs** :

```
{
  "id" :    "1",
  "prenom": "Steven",
  "nom": "Spielberg"
}
```

Il y aura aussi 6 points (sur 30) d'accordés pour les pratiques de programmation et le respect des normes de l'API REST vues en classe et explicitées plus haut. Produisez donc un code clair, commenté, et modulaire qui gère les exceptions appropriées.

**Le total compte pour 30% de votre résultat final.**

### **Ce qu'il faut remettre**

Compressez le projet final dans un fichier ZIP identifié par vos noms et déposez ce dernier dans le LÉA avant minuit à la date d'échéance. Vous n'avez pas à installer le projet sur Webdev : si vous respectez ce que j'ai demandé à la lettre, ce sera facile pour moi de le tester de mon côté.