|  |  |
| --- | --- |
| WEBA – Test 3 | |
| Nom et prénom des deux étudiant-e-s | Filipe Dias Morais  Luca Bohlen |

|  |  |
| --- | --- |
| Modalités | |
| Type de travail | Projet en groupe de deux |
| Fichiers sources | Documentation d’exemple |
| À rendre | * Application fonctionnelle selon le présent cahier des charges * Un script SQL permettant de créer votre base de données * Votre documentation * Collection Postman permettant de tester au moins 3 URLs de votre API |
| Moyens d’aide | Tout, sauf les autres groupes |
| Nombre total de points | 93 |

|  |  |
| --- | --- |
| Résultat | |
| Points obtenus |  |
| Note |  |

# Consigne

On vous demande de réaliser une API en PHP qui représentera des notes (à des tests) dans des branches.

Une branche (subject en anglais) est représentée par un nom (name). Une note (grade) par une description (description), une valeur (la note) (value), une date (et heure) (date) et est associée à une branche.

Votre API devra stocker ses données dans une base de données locale (MySQL, MariaDB). Le script permettant de créer votre base de données (ainsi que quelques données de tests) fait partie du rendu.

Le code source doit être structuré avec MVC. Les réponses de l’API doivent être au format JSON.

Les requêtes suivantes sont à gérer :

Requête 1 – Obtention des branches

* L’API doit lister toutes les branches avec leurs champs.
* Un paramètre doit permettre d’inclure dans chaque branche ses notes. Dans ce cas, tous les champs de chaque note sont transmis.
* Un paramètre doit permettre de ne retourner qu’une seule branche en passant son id. Dans ce cas, le retour ne sera pas un tableau.

Requête 2 – Obtention des notes

* L’API doit lister toutes les notes avec leurs champs. La branche de chaque note est retournée sous forme d’un id uniquement.
* Un paramètre doit permettre d’obtenir uniquement les notes d’une branche en spécifiant son id.
* Un paramètre doit permettre de ne retourner qu’une seule note en passant son id. Dans ce cas, le retour ne sera pas un tableau.

Requête 3 – Ajout d’une note

* Il doit être possible d’ajouter une nouvelle note, en spécifiant l’id de sa branche. L’API doit renvoyer l’id de la nouvelle note créée (au format JSON).

Requête 4 – Suppression d’une note

* Il doit être possible de supprimer une note.

Requête 5 – Moyenne

* Il doit être possible d’obtenir la moyenne d’une branche.

Pour chaque requête, supportez les différents types de retours vus en cours, selon leur utilité (200, 201, 204, 400, 404, etc.).

En plus du code source de l’API, il est demandé de réaliser une documentation technique permettant à un développeur externe d’utiliser votre API. Chaque requête de votre API devra être détaillée avec les éléments suivants :

* Description de la requête (ce qu’elle permet de faire) ;
* Son URL ;
* Ses éventuels paramètres et leur format attendu ;
* Ses codes HTTP de retours ;
* Le format des données retournées.

Dans les fichiers sources, une documentation détaillant uniquement la première requête est fournie. Vous pouvez l’utiliser comme référence pour votre documentation.

Utilisez le logiciel Postman pour interagir avec votre API et la tester. Une « collection » Postman est à rendre.