

Travail pratique 1 : Un Service Web orienté objet en PHP (pondération 20%)

Le travail consiste à créer un Service Web qui permettra aux utilisateurs de manipuler des données provenant de la Ville de Montréal. Le jeu de données est constitué d'une liste des arrondissements et des piscines de la ville. Les données ainsi que la structure des tables sont fournis.

1- Spécifications générales :

Accès au service web

Il y aura deux niveaux d'accès au Service Web. Le niveau de base (requête GET) permet seulement de faire des requêtes pour obtenir des informations provenant du jeu de données. Ce niveau ne requiert aucune identification.

Le niveau admin (requêtes POST et DELETE) permet d'effectuer des modifications sur les données. L'utilisateur admin pourra ajouter, effacer et modifier des données. Ce niveau requiert que l'utilisateur s'identifie au service lorsqu'il effectue ses requêtes de modifications:

- Usager : 582P41
- Mot de passe : ABC123

Gestion des erreurs de requêtes :

- Lorsque la requête est invalide (contient des données erronées ou non valides), le serveur envoie une réponse 400 (*HTTP/1.1 400 Bad Request*).
- En cas d'erreur d'authentification, le serveur envoie une réponse 401 (*HTTP/1.1 401 Unauthorized*).
- Si la requête est faite avec la mauvaise méthode (exemple, POST au lieu de GET), le serveur envoie une réponse 405 (*HTTP/1.1 405 Method Not Allowed*).

Déploiement :

Votre code doit être déployé sur webdev dans le dossier /www/wspiscine/
Le fichier principal doit être index.php

Critères d'évaluation

- Respect des consignes (15%)
- Utilisation pertinente de la programmation orientée objet (20 %)
- Fonctionnement adéquat du service Web (30 %)
- Qualité des tests unitaires (20 %)
- Qualité du code et des commentaires (PHP) (15 %)

Précision :

Votre code doit être indenté et commenté correctement.

Remise :

Remettez votre dossier compressé sur Léa.

2- Documentation de l'API REST du Service Web

Ressource : arrondissement

GET arrondissements

@return Format JSON

Retourne la liste des arrondissements

Exemple de résultat:

```
[{
  "code3l":"ANJ",
  "nom":"Anjou",
  "acronyme":"ANJ"
}, {
  "code3l":"LAC",
  "nom":"Lachine",
  "acronyme":"LAC"
}, {
  etc.
}]
```

GET arrondissements/:id

@return Format JSON

Retourne un arrondissement identifié par son identifiant unique "id" qui correspond à la clé primaire "code3l" (code de 3 lettres) de la table arrondissement.

Si aucun résultat, elle retourne un élément vide.

Exemple de résultat (arrondissements/ANJ)

```
[{
  "code3l":"ANJ",
  "nom":"Anjou",
  "acronyme":"ANJ"
}]
```

Ressource : piscine

GET piscines

@return Format JSON

Retourne la liste des piscines

Exemple de résultat

```
[{
  "id_piscine": "1", "type": "Piscine ext\u00e9rieure",
  "nom": "Parc Lucie-Bruneau",
  "arrondissement_code3l": "ANJ",
  "adresse": "7 051, Avenue de l\u2019Alsace",
  "propriete": "Municipale",
  "gestion": "Priv\u00e9"
}, {
  "id_piscine": "2",
  "type": "Piscine ext\u00e9rieure",
  "nom": "Parc Ch\u00e9nier",
  "arrondissement_code3l": "ANJ",
  "adresse": "5 555, Avenue de l\u2019Ar\u00e9na",
  "propriete": "Municipale",
  "gestion": "Priv\u00e9"
}, {
  etc.
}]
```

GET piscines?code3l=<code3l> ou GET arrondissements/:id/piscines

@param code3l de l'arrondissement

@return Format JSON

Retourne la liste des piscines d'un arrondissement à partir de son code de 3 lettres.

Si aucun résultat, elle retourne un élément vide.

Exemple de résultat (piscines?code3l=OUT)

```
[{
  "id_piscine": "25",
  "type": "Piscine ext\u00e9rieure",
  "nom": "Parc John-F-Kennedy",
  "arrondissement_code3l": "OUT",
  "adresse": "860, Avenue Outremont",
  "propriete": "Municipale",
  "gestion": "Priv\u00e9"
}, {
  "id_piscine": "26",
  "type": "Pataugeoire",
  "nom": "Parc John-F.-Kennedy",
  "arrondissement_code3l": "OUT",
  "adresse": "860, Avenue Outremont",
  "propriete": "Municipale",
  "gestion": "Priv\u00e9"
}]
```

GET piscines/:id

@return Format JSON

Retourne l'information sur une piscine identifiée par son identifiant unique "id" qui correspond à la clé primaire "id_piscine" de la table piscine.

Si aucun résultat, elle retourne un élément vide.

Exemple de résultat (piscines/10)

```
[{
  "id_piscine": "10",
  "type": "Pataugeoire",
  "nom": "Parc Ch\u00e9nier",
  "arrondissement_code3l": "ANJ",
  "adresse": "5 555, Avenue de l'\u2019Ar\u00e9na",
  "propriete": "Municipale",
  "gestion": "Priv\u00e9"
}]
```

GET piscines/:id/horaires

@return Format JSON

Retourne les horaires d'une piscine identifiée par son identifiant unique "id" qui correspond à la clé primaire "id_piscine" de la table piscine.

Si aucun résultat, elle retourne un élément vide.

Il n'y a qu'une seule plage horaire par jour.

Exemple de résultat (piscines/1/horaires)

```
[{
  "id_piscine": "1",
  "jour": "0",
  "debut": "08:00:00",
  "fin": "16:00:00"
}, {
  "id_piscine": "1",
  "jour": "4",
  "debut": "09:00:00",
  "fin": "11:00:00"
}]
```

GET piscines/:id/horaires?date=<date>

@param date en format AAAAMMMJJ

@return Format JSON

Retourne l'heure d'une piscine (id = id_piscine de la table piscine) selon la date passée en paramètre.

Si aucun résultat, elle retourne un élément vide.

Si la date est invalide, elle retourne les horaires complets de la piscine.

Exemple de résultat (piscines/1/horaires?date=20180503)

```
[{
  "id_piscine": "1",
  "jour": "4",
  "debut": "09:00:00",
  "fin": "11:00:00"
}]
```

POST piscines/:id/horaires

@param horaires de la piscine (Format JSON)

@return Array of booleans

Permet de modifier les horaires d'une piscine (id = id_piscine de la table piscine) déjà créée.

Si l'horaire existe, il est modifié, sinon il est créé.

L'information est passée en POST.

L'utilisateur doit s'identifier pour avoir accès à cette commande.

Retourne un tableau avec un booléen pour chaque horaire créé ou modifié, égal à 'true' si la mise à jour a été effectuée, à 'false' sinon.

Exemple de requête (POST : piscines/1/horaires)

Variable PHP, avant le JSON encode:

```
$horaires = array(
    array("id_piscine" => 1,
        "jour" => 2,
        "debut" => "10:30",
        "fin" => "19:10"),
    array("id_piscine" => 1,
        "jour" => 4,
        "debut" => "12:30",
        "fin" => "22:10")
);
```

DELETE piscines/:id/horaires

@return Boolean

Permet d'effacer les horaires pour une piscine spécifique (id = id_piscine de la table piscine).

L'utilisateur doit s'identifier pour avoir accès à cette commande.

Retourne un booléen qui indique si l'opération a été faite ou non.

Si la piscine n'existe pas, retourne false.