



# **Titre professionnel DWWM ECF 3**

## **Back-end « Gestion d'un parc locatif »**

### **Compétences évaluées**

#### Activité-type 2

5. Créer une base de données
6. Développer les composants d'accès aux données
7. Développer la partie back-end d'une application web ou web mobile

### **Contexte du projet**

Réaliser le back-end d'une application web, sous une approche « web services » destinés être appelés ultérieurement par un front-end pour le compte d'un organisme de gestion d'un parc locatif de locaux professionnels.

### **Charte graphique**

Néant. Aucun composant front-end n'est à développer.

### **Processus métier**

L'application web est destinée à gérer les **entités métier** du processus de gestion de la location de locaux professionnels. Les règles métier de l'organisme gestionnaire sont les suivantes :

- L'organisme est propriétaire de biens immobiliers qui sont caractérisés par :
  - **une surface** (en mètres carrés)
  - **un nombre de pièces**
  - **un prix de loyer mensuel en euros**
  - **une adresse** (dans un souci de simplification, on ne retiendra que le nom de la ville pour la localisation du bien).
- Pour identifier ses biens, l'organisme attribue à chacun d'entre eux un **code de gestion unique** sous la forme « LOC#### » où #### correspond à une série de 4 chiffres maximum.
- **Chaque bien immobilier** peut être loué simultanément à une seule et unique entreprise cliente.

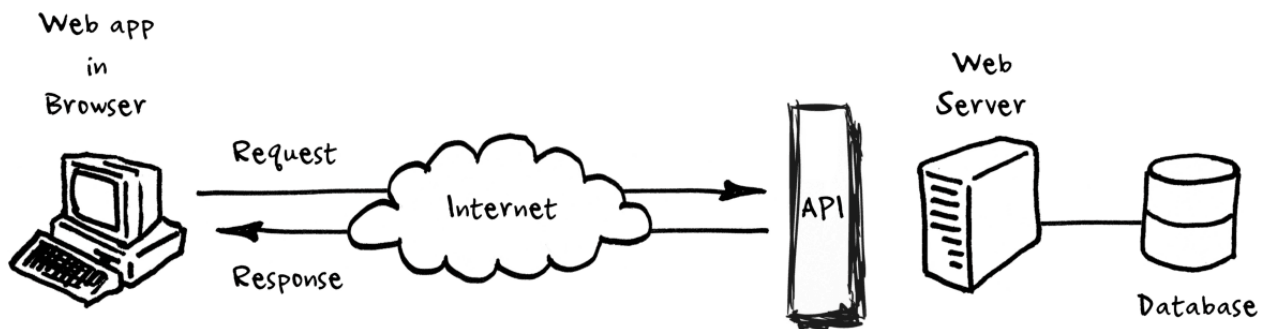
- Chaque entreprise cliente peut louer un ou plusieurs biens simultanément.
- Chaque entreprise cliente est caractérisée par :
  - son nom
  - son numéro unique d'immatriculation officiel appelé SIREN (dans un souci de simplification, on considérera que le SIREN est un simple code à 9 caractères).
- Pour identifier ses clients, l'organisme attribue à chacun d'entre eux un code client unique sous la forme « CLI#### » où #### correspond à une série de 4 chiffres maximum.

L'organisme ne conserve pas d'historique sur les locations et souhaite seulement avoir un état des lieux à un instant donné du parc locatif : bien loué ou bien vacant.

### Spécifications fonctionnelles

Le back-end de l'application se présente sous la forme d'une API « web services » permettant de manipuler les entités du processus métier selon le modèle CRUD. L'API a donc pour but de lister, créer, modifier, supprimer et rechercher les entités métier.

Techniquement, l'API est un jeu d'URL HTTP en méthode GET renvoyant des données au format JSON. Le format des URL à développer n'est pas standardisé mais correspond à une implémentation « propriétaire » interne à l'organisme. Le schéma suivant illustre le contexte technique.



Dans un souci de simplification, seule l'URL permettant de lister toutes les entités métier de type « client » sera implémentée. La spécification de l'URL est la suivante :

## ● Lister toutes les entités

- Rôle : extraire tous les codes métier (et uniquement les codes) des entités d'un type donné présentes dans la base de données
- Format de l'URL d'appel : <http://localhost/type-entite.php?action=list>
- Format du JSON renvoyé :

```
{
    "action": "list",
    "entity": "type-entite",
    "data": [
        "code-entite-1",
        "code-entite-2",
        ...
        "code-entite-n"
    ]
}
```

- Exemple : <http://localhost/client.php?action=list> liste tous les codes client en renvoyant le JSON suivant :

```
{
    "action": "list",
    "entity": "client",
    "data": [
        "CLI0001",
        "CLI0002",
        ...
    ]
}
```

Pour des questions de sécurité, le back-end se connectera à la base de données à l'aide du compte utilisateur MySQL « app\_loc » avec le mot de passe « 8y-toP# ».

## Réalisation attendue

La réalisation portera sur les seuls éléments suivants :

- Création de la base de données MySQL permettant de stocker les entités métier
- Peuplement de la base de données « à la main » avec un jeu de test
- Création des pages PHP auto-documentées précisées dans la partie « Spécifications fonctionnelles »

## **Jeu de test**

Afin de tester l'application, il est demandé de saisir un jeu de données de 3 entités minimum pour chaque entité métier.

## **Contraintes techniques**

La réalisation devra impérativement respecter les contraintes suivantes :

- Réalisation des pages PHP sans framework
- Utilisation de PHP version 7.4
- Utilisation de MySQL version 5.6 ou 5.7
- Sécurisation des paramètres des requêtes HTTP GET
- Utilisation du connecteur PHP PDO pour les requêtes SQL
- Sécurisation des requêtes SQL
- Auto-documentation du code
- Respect des normes, règles et conventions de programmation back-end
- Modularité et réutilisation du code PHP
- Encapsulation de la connexion PDO au sein d'une classe utilitaire
- Conformité de syntaxe du format JSON.

La réalisation pourra éventuellement respecter les contraintes optionnelles suivantes :

- Ecriture du code PHP en POO
- Organisation du code PHP selon une architecture MVC.

## **Livrable**

Le livrable sera constitué des éléments suivants :

- Un fichier zip comprenant l'arborescence du back-end PHP
- Un deuxième fichier zip contenant le script de restauration de la base de données MySQL (structure, données de test et compte utilisateur) ainsi qu'une image correspondant au schéma relationnel physique de la base de données.

Aucune documentation technique additionnelle n'est demandée.

## **Date de livraison**

Le livrable sera remis le 6 décembre sur le dépôt Métis dans la mission « ECF ».