

## Mini projet : Services web REST

Nous continuons le mini projet sur la gestion de prêt de véhicules du TP précédent en ajoutant des de fonctionnalités. De plus, nous allons utiliser une base de données pour le service. À ce effet un **script sql** pour créer des tables et éventuellement les données dans la BDD est à déposer en même temps que votre projet, ainsi que un **README** pour expliquer votre code et son utilisation.

Notez que il y aura **quatre projets distincts** à rendre également, dont un projet pour le serveur de type **Java Entreprise / Rest service** sous IntelliJ ou **Web Application** sous Netbeans, trois projets pour le client : deux projets **Java Application** avec **Swing GUI / Java FX** pour la gestion des véhicules et la gestion des membres, et une projet web **HTML5/Javascript** pour la recherche, la réservation des véhicules et la consultation de compte d'utilisateur.

### Sujet \_\_\_\_\_ Gestion de prêt de véhicules

Nous allons réaliser un web service permettant d'aider à la gestion de prêt des véhicules de services (voitures et motos) exclusivement à destination d'une agence immobilière. Plus précisément, le service aide à la gestion des prêts, réservations et retours des véhicules de services de l'agence. Seuls les employés de l'agence immobilière peuvent emprunter des véhicules disponibles pour réaliser des trajets de travail (des visites des appartements par exemple). Le service de gestion de prêt de véhicules est géré par un administrateur/responsable de service. La recherche et la consultation de véhicules disponibles peut être faite par l'employé via une page web.

L'objectif de cet exercice est de développer un service web REST qui permette d'interroger une base de données pour connaître la liste de tous les véhicules ainsi que les véhicules disponibles pour le prêt, d'obtenir les informations d'un véhicule en particulier et confirmer la réservation d'un véhicule par employé de l'entreprise, . . . . Les réponses de ce service web seront toutes **transmises en deux formats XML et JSON** suivant le choix de l'utilisateur lors de l'appel de service.

Chaque employé de l'agence immobilière, ayant un permis de conduit, doit s'inscrire au service de prêt s'il veut emprunter des véhicules. Pour cela, il doit fournir son nom, son prénom, son numéro de sécurité sociale (il est formé du numéro d'inscription (NIR) à 13 chiffres et d'une clé de contrôle à 2 chiffres<sup>1</sup>), son numéro de permis de conduire et son adresse domicile. Une fois inscription est validée, l'administrateur de service va donner à la personne un numéro d'identification (numéro de membre) pour le service de prêt. Un employé inscrit, avec son numéro de membre, peut ensuite emprunter des véhicules disponibles de l'agence. Pour chaque emprunt par un employé de l'agence, le numéro d'immatriculation en format AA-001-AA<sup>2</sup> du véhicule prêté ainsi que la durée de prêt sont notés dans le système. Chaque emprunt ne peut **pas durer plus qu'une**

---

1. [https://fr.wikipedia.org/wiki/Numéro\\_de\\_sécurité\\_sociale\\_en\\_France](https://fr.wikipedia.org/wiki/Numéro_de_sécurité_sociale_en_France)

2. [https://fr.wikipedia.org/wiki/Plaque\\_d%27immatriculation](https://fr.wikipedia.org/wiki/Plaque_d%27immatriculation)

**journée** (de 8h00 à 18h00 de même jour) et l'employé ne peut pas **faire plus que 3 emprunts par semaine**.

Pour la gestion des employés pour le service de prêt, l'agence immobilière propose de fournir l'accès au service web REST via une **application java** avec une interface graphique comme ci-après. Cette application permet de contrôler les appels aux différents fonctionnalités suivantes :

- Inscrire un employé
- Afficher les informations d'un employé et la liste de tous les employés
- Mettre à jour (= modifier) les informations d'un employé comme nom, prénom, ... et **pas le numéro d'identifiant ni le numéro de sécurité sociale**. Pour cela, l'application doit permettre de chercher les informations d'un employé par numéro d'identifiant ou numéro de sécurité sociale
- Supprimer un membre induit à **l'annulation des prêts en cours** du membre concerné
- **Emprunter et retourner** : le prêt d'un véhicule par un employé (après donner le numéro de membre et le numéro d'immatriculation du véhicule à emprunter et à retourner ainsi que la date de l'emprunt). Indication : Vérifier que l'emprunt est valide, c-à-d le véhicule est disponible ce jour là et l'employé n'est pas encore dépassé le nombre d'emprunts maximal, la validité de la date, ...

Notez que lors de l'inscription de membre, le numéro de membre est attribué automatiquement par l'application et pré-rempli dans le formulaire d'inscription.

The image shows three screenshots of a web application interface for member management. The first screenshot, titled 'Gestion des membres', shows a menu with four buttons: 'Inscrire membre', 'Modifier membre', 'Supprimer membre', and 'Emprunter véhicule'. It also has radio buttons for 'XML' (selected) and 'Json'. The second screenshot, titled 'Inscription membre', shows a form with fields for 'N° membre' (pre-filled with 123), 'Nom & prénom', 'N° permis', 'N° sécu sociale', and 'Adresse domicile', with a 'Valider Inscription' button. The third screenshot, titled 'Modifier membre', shows a form with fields for 'N° member' (pre-filled with 123), 'Nom & Prénom' (pre-filled with Dupont Jean), 'N° permis' (pre-filled with 051227308989), 'N° Sécu sociale' (pre-filled with 123456789101213), and 'Adresse domicile' (pre-filled with 12 rue jardin, 54000 Nancy), with a 'Valider modification' button.

Le système doit gérer également des véhicules (voitures et motos) de l'agence. Dans ce contexte, chaque véhicule est caractérisé par un numéro d'immatriculation (identifiant du véhicule), le type de véhicule (voiture ou motos), la marque, le modèle et la catégorie du véhicule (électrique, essence, hybride ou autres), la boîte de vitesse (manuelle ou automatique), le nombre de places et une description pour les informations supplémentaires : l'état du véhicule, les options, ... L'administrateur du système peut gérer les véhicules via une **application java** pour

- Ajouter un véhicule
- Rechercher un véhicule par le numéro d'immatriculation
- Mettre à jour (= modifier) les informations d'un véhicule
- Supprimer un véhicule induit à **l'annulation des prêts en cours** du véhicule concerné.

Ci-dessous d'un exemple de l'interface de l'application pour la gestion de véhicule :

Gestion des véhicules
☒ XML
☐ Json

Ajouter véhicule
Recherche véhicule

Supprimer véhicule
Modifier véhicule

Ajouter un véhicule

N° immatriculation

Type de véhicule
Voiture

Marque

Modèle

Catégorie
Electrique

Boîte de vitesse
☐ Manuelle
☐ Automatique

Nombre de places
4

Description
GPS, clim

Valider

Le service dispose également un système CRUD via une **page web** pour l'interrogation des véhicules. Les ressources manipulées par le service sont donc les véhicules. Le service web REST doit fournir les fonctionnalités suivantes aux utilisateurs pour rechercher les véhicules :

- Lister l'ensemble de tous les véhicules dans le système
- Vérifier la disponibilité d'un véhicule pour une date donnée
- Rechercher un véhicule en fonction de différents critères (type, marque, modèle, nombre de places, ...)

Ci-dessous d'un exemple de la page web de recherche d'un véhicule :

## Véhicules de l'agence

Recherche un véhicule

Type
--Choisir voiture ou moto--

Marque

Modèle

Nombre de places
--Choisir le nombre de places--

Disponibilité
dd/mm/yyyy

Rechercher

Les représentations fournissent par le service sont en format **JSON** et **XML**. Le client de service peut demander les ressources soit en JSON, soit en XML suivant le choix de l'utilisateur lors de l'appel de service.

Ci-après un exemple de la structure XML d'un véhicule et d'un membre :

```

<vehicule>
  <registrationNumber>
    ...
  </registrationNumber>
  <type> ... </type>
  <brand> ... </brand>
  <model> ... </model>
  <category> ... </category>
  <gearbox> ... </gearbox>
  <seats> ... </seats>
  <comments> ... </comments>
</vehicule>

```

```

<member>
  <membershipNumber>
    ...
  </membershipNumber>
  <name> ... </name>
  <drivingLicenseNumber>
    ...
  </drivingLicenseNumber>
  <socialSecurityNumber>
    ...
  </socialSecurityNumber>
  <address> ... </address>
</member>

```

Ci-après un exemple de la structure XML la liste des véhicules et des membres :

```

<vehicles>
  <vehicle> ... </vehicle>
  <vehicle> ... </vehicle>
</vehicles>

```

```

<members>
  <member> ... </member>
  <member> ... </member>
</members>

```

Pour améliorer le service de prêt des véhicules dans l'entreprise, nous allons ajouter **le service de réservation de véhicule à partir d'un page web**. Pour cela, nous allons créer un **système d'utilisateur** dans lequel chaque employé procède un compte (= un login avec le numéro de membre) et un mot de passe. Une fois employé est login à son compte sur le site du service, il peut consulter la liste des prêts de véhicules (passés et en cours, en réservation ou validés), modifier le mot de passe, ... Il peut également faire une recherche des véhicules disponibles et puis **réserver** le véhicule (au moins deux jours avant la date de prêt). L'administrateur du service, via l'application java, peut consulter la liste de toutes les réservations des employés et **valider une réservation**. Une fois la réservation est validée, l'employé peut emprunter le véhicule pour la date demandée. Indication : Lors que l'administrateur valide la réservation, vérifiez bien que le prêt est valide, si ce n'est pas le cas, un message d'erreur sera affiché sur l'interface java en indiquant la raison avec un message explicite, par exemple, le nombre de prêts maximaux par semaine est atteint, le véhicule n'est plus disponible, ...