POO : API & Outils
TD et TP

Anass OUSMOI anass.ousmoi@univ-lemans.fr
Florentin PAILLIER florentin.paillier@soprasteria.com
Karim RAKHILA karim.rakhila@soprasteria.com

TP 4 : DM noté 2/2 : Utilisation d'API web avec Spring Boot

Ce TP est noté !

- Vous pouvez le terminer chez vous ;
- A rendre via : UMTICE (cours : POO API et outillages) ;

Ce TP est dans la continuité du TP précèdent :

- Utilisation de Spring Framework pour ajouter les fonctionnalités de base de données, utilisation d'API REST en ligne, génération de pages web ;
- Utilisation de H2 pour créer une base de données in-memory ;
- Utilisation de Thymeleaf pour faire des pages web ;
- Récupération d'une adresse via un formulaire pour ensuite appeler une API donnant les coordonnées GPS;
- Utilisation d'une API web pour récupérer la météo aux coordonnées GPS précises.

Etape 1

Nous allons incorporer une nouvelle page (à l'url \addresse") contenant le formulaire du TP3, permettant de demander une adresse.

- Créez un contrôleur, et ajoutez un formulaire permettant d'insérer l'adresse dans son template.

```
<form action="/meteo" method="post">
	<div>
	<label for="address">Please input your adress :</label>
		<input name="address" id="address">
	</div>
	<div>
		<button>Get the weather at the given address !</button>
	</div>
</form >
```

- Mettez à jour la navbar pour qu'un lien aille sur cette nouvelle page \adresse".

Etape 2

Nous allons maintenant travailler sur les interactions entre pages.

- Créez un nouveau contrôleur, et son template (vide pour l'instant) pour l'url « meteo ».

- Cette page « meteo » doit être la cible du formulaire de la page « adresse » que vous venez de créer, et sur validation du formulaire, doit envoyer le contenu vers le Controller météo.
- Modifier les attributs du formulaire précédent pour rediriger vers « meteo ».
- Dans le contrôleur de la météo, ajoutez une méthode permettant de recevoir des appels POST (indice : annotation PostMapping).
- Récupérez la donnée du formulaire entrée par l'utilisateur dans le template. Dans la méthode du contrôleur de la météo, trouvez le bon paramètre permettant d'indiquer à Spring de valoriser l'adresse insérée dans le formulaire (aidez-vous de internet et de la documentation Spring).
- Une fois que le code compile, verriez qu'il fonctionne en mettant un point d'arrêt et en lançant le programme en mode debug.

Etape 3

A partir de l'adresse récupérée de l'utilisateur, faites un appel HTTP GET vers l'API Adresse de Etalab.

- Lisez la documentation Etalab Adresse : https://geo.api.gouv.fr/adresse
- Faites des tests d'URL via le navigateur (l'API fonctionne en GET, donc testable facilement via le navigateur).
- Regardez la structure de la réponse donnée par Etalab Adresse, et concevez l'objet Java de réponse (cf. documentation Spring : https://spring.io/guides/gs/consuming-rest/).
- N'oubliez pas le proxy à éventuellement configurer si vous êtes sur le réseau de l'université : dans la partie « Astuces » dans ce document, il y a une aide sur la configuration du proxy pour Java.

Etape 4

Pour vérifier que vous obtenez bien les informations dans votre application, essayez de l'afficher dans le template Thymeleaf du contrôleur météo, ou via point d'arrêt en mode debug.

Etape 5

Une fois que vous arrivez à obtenir les coordonnées GPS, vous pouvez maintenant appeler l'API de MeteoConcept (GET /api/forecast/daily). Lisez la documentation pour comprendre comment spécifier la clé API, comment indiquer les coordonnées GPS, et comment est structure la reponse.
N'hésitez pas à faire des tests via votre navigateur.

- Lisez la documentation Meteo-Concept : https://api.meteoconcept.com/documentation
- Créez un compte pour avoir une clé API.
- Faites des tests d'URL via le navigateur (l'API fonctionne en GET, donc testable facilement via le navigateur).
- Regardez la structure de la réponse donnée par Meteo-Concept, et concevez l'objet Java de réponse (cf. documentation Spring)

Etape 6

Mettez la réponse à chacune de ces questions dans le README de votre projet :

- Faut-il une clé API pour appeler MeteoConcept ?
- Quelle URL appeler ?
- Quelle méthode HTTP utiliser ?
- Comment passer les paramètres d'appels ?
- Où est l'information dont j'ai besoin dans la réponse :
- Pour afficher la température du lieu visé par les coordonnées GPS
- Pour afficher la prévision de météo du lieu visé par les coordonnées GPS

Etape 7

Pour rendre le devoir-maison :

- Dans le README.md du projet, ajoutez un lien vers votre projet GitHub.
- Poussez votre code sur GitHub.
- Faites un zip de votre code source (lancez un « mvn clean » pour enlever les fichiers temporaires).
- Uploadez le zip sur UMTICE, dans la partie devoir-maison.