

TP03 : Mise en place d'un échange SOAP entre Cordova et Java

1) Mise en place de Tomcat

Si ce n'est pas déjà fait, vous pourrez télécharger Tomcat à l'adresse suivante :

<http://apache.mindstudios.com/tomcat/tomcat-8/v8.5.24/bin/apache-tomcat-8.5.24-windows-x64.zip>

Décompressez-le quelque part sur le système.

Afin de pouvoir contacter le web service, il faut autoriser le CORS (Cross-origin resource sharing). Pour ce faire, il faut modifier le fichier de configuration 'web.xml', rendez vous dans le répertoire :

« MY_PATH\apache-tomcat-8.5.24\conf\web.xml »

Et ajouter les lignes suivantes

```
<filter>
  <filter-name>CorsFilter</filter-name>
  <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
  <init-param>
    <param-name>cors.allowed.origins</param-name>
    <param-value>*</param-value>
  </init-param>
  <init-param>
    <param-name>cors.allowed.methods</param-name>
    <param-value>GET,POST,HEAD,OPTIONS,PUT</param-value>
  </init-param>
  <init-param>
    <param-name>cors.allowed.headers</param-name>
    <param-value>Content-Type,X-Requested-With,accept,Origin,Access-Control-Request-Method,Access-Control-Request-Headers</param-value>
  </init-param>
  <init-param>
    <param-name>cors.exposed.headers</param-name>
    <param-value>*</param-value>
  </init-param>
  <init-param>
    <param-name>cors.support.credentials</param-name>
    <param-value>>true</param-value>
  </init-param>
</filter>
```

```
</init-param>
<init-param>
  <param-name>cors.preflight.maxage</param-name>
  <param-value>180</param-value>
</init-param>
</filter>
<filter-mapping>
  <filter-name>CorsFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

2) Déployer le web service

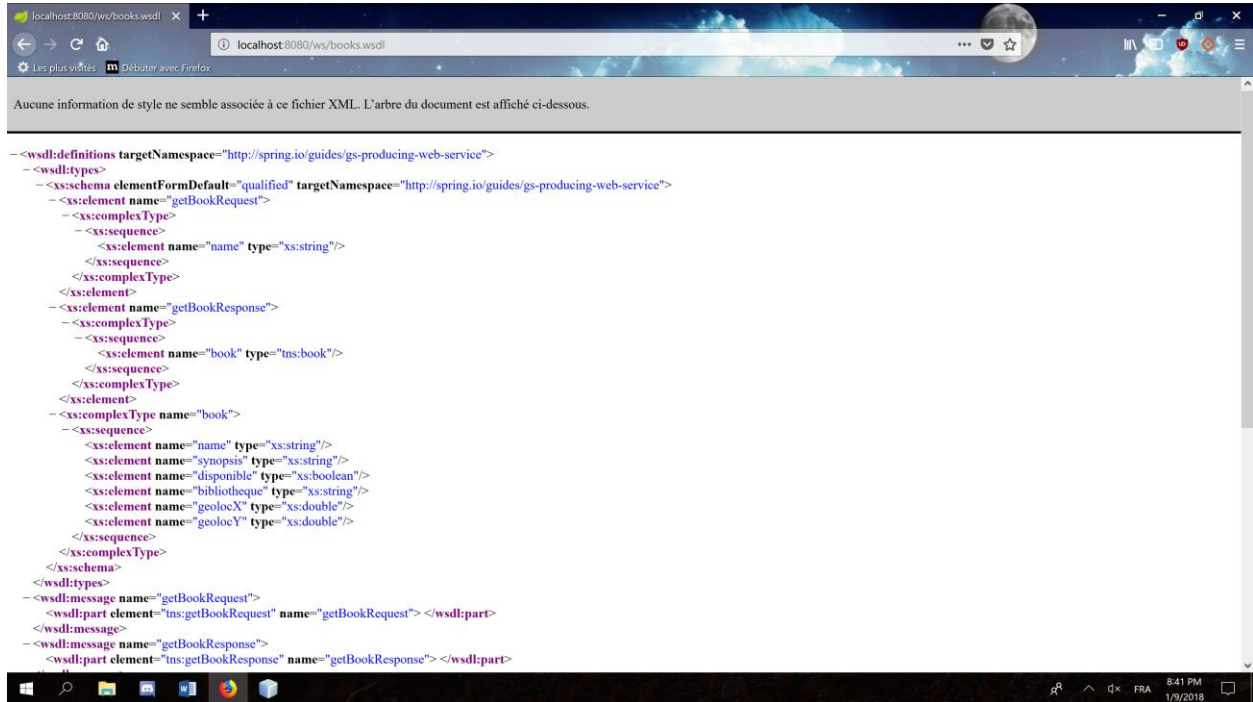
Pour déployer le web service, télécharger l'archive du projet JAVA fourni avec le TP.

Pour le lancer dans Tomcat, vous pouvez procéder de deux manières :

- Ouvrir le projet (MAVEN), avec votre IDE préféré et le déployer dans Tomcat (si votre IDE possède une intégration Tomcat)
- Déployer le WAR, dans le dossier « target » de l'archive copier le war dans le dossier WebApp de Tomcat.

Une fois Tomcat lancé, vous pouvez vérifier le bon fonctionnement du web service :

<http://localhost:8080/ws/books.wsdl>



3) Créer le front

Récupérer l'archive et dans le fichier Index.js compléter ; le script d'appel et d'affichage de SOAP.

Le but de ce TD est d'aller chercher le livre « UML », d'afficher ses informations et de mettre un bouton qui lors du click ouvre une map de localisation de la bibliothèque universitaire.

Phase 1 : à l'aide de vos connaissances, de l'enveloppe affichée au tableau et du projet Java, rédiger l'enveloppe à envoyer au serveur afin de récupérer un livre.

Phase 2 : créer le script ajax (avec jquery ou autre méthode), afin d'envoyer l'enveloppe SOAP au serveur, ensuite rédiger la fonction de callback 'onSuccess'. De récupération des informations et d'affichage dans le HTML.

Phase 3 : rédiger le script JavaScript qui affiche une carte google maps afin de géolocaliser la bu correspondante au livre renvoyé par le serveur.

Phase 4 : faites évoluer le code de la première partie 3 afin d'afficher l'itinéraire correspondant (position actuelle vers la BU)

Bonne chance