	<p>Master Pro TAA/ 2013-2014</p>	
-----------------------------------------------------------------------------------	--------------------------------------	--

TP 3/TAA: Construire une application n-tiers à base du socle technologique JPA, Jersey, AngularJS

Objectif :

L'objectif de ce TP est de prendre en main *un socle technologique* composé de JPA, JERSEY, et AngularJS afin de construire une application web permettant le tracking de performance sportive.

Installation des outils.

Je souhaite volontairement que vous partiez des outils disponibles sur le Web afin de bien comprendre les différentes étapes de la mise en place d'un serveur Tomcat ou Jetty et d'un serveur EJB. Les différents outils que nous allons utiliser dans ces Tps vont nécessiter 60Mo de d'espace disponible sur vos comptes. Si vous ne disposez pas de cet espace, n'hésitez pas à travailler sur /tmp. Dans ce dernier cas, n'oubliez pas de sauvegarder votre travail en fin de séance car ces données seront sauvegardées localement sur le poste que vous utilisez. Vous risquez donc de perdre des données entre deux séances.

Tomcat.

Téléchargez Tomcat version 7.0.42 sur le site de Tomcat (dernière version stable de la branche 7.0.x)
<http://tomcat.apache.org/download-70.cgi>



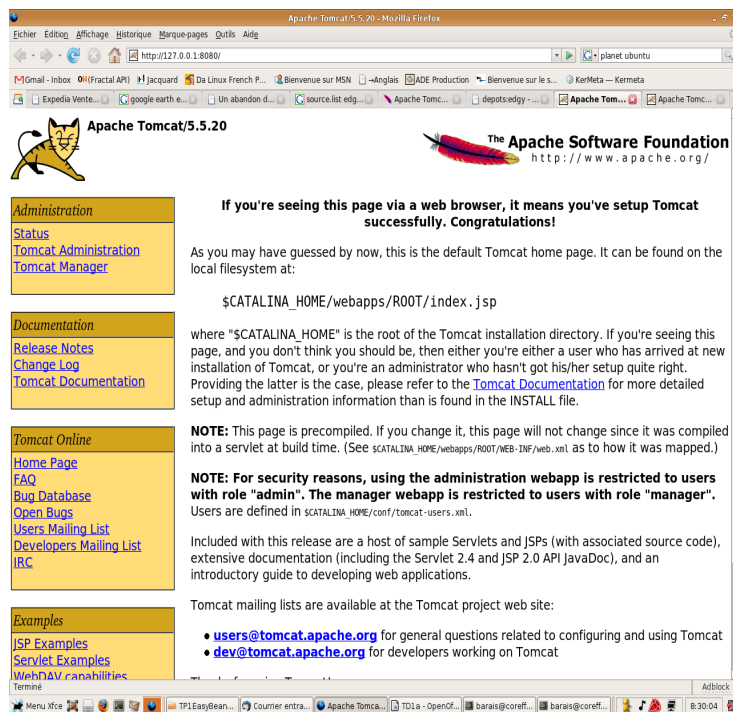
Déployez Tomcat dans votre répertoire de travail.

```
> mkdir 'mon_rep_de_travail_TAA'
> cd 'mon_rep_de_travail_TAA'
> tar -xzf 'tomcat_download_rep'/apache-tomcat-*.tar.gz
```

Test de votre installation

```
> cd apache-tomcat-7.0.42/
> bin/startup.sh
```

Lancez un navigateur, si vous voyez la page ci dessous, c'est bon.



Endomondo like

L'objectif de ce projet est de construire une application type réseau social permettant de comparer ses performances sportives avec ses amis, ses voisins, ... dans la lignée de endomondo.

Ces applications sont idéales pour la course à pied, le cyclisme, la marche ou n'importe quelle activité basée sur la distance.

Fonctionnalités et caractéristiques de ce genre d'applications :

- Enregistrer tout sport en extérieur en incluant la durée, la distance, la vitesse et les calories
- Saisir une séance manuellement, ex : course sur tapis, spinning ou poids
- Recevoir des messages d'encouragement de ses amis - ils écrivent un message court sur notre site internet qui vous est lu à voix haute quelques secondes plus tard
- Voir son parcours sur une carte
- Enregistrer son rythme cardiaque
- Voir l'historique de ses séances et étudier les temps intermédiaires à chaque km ou mile
- Définir son objectif de distance et obtenir les informations du *coach* audio sur l'avancée de l'objectif
- Synchronisation des séances à travers de multiples plateformes (saisie web - identification requise)
- Publier ses séances sur la Timeline de Facebook
- Voir les dernières séances de ses amis en temps réel
- Courir contre le chrono d'un ami avec l'aide du coach audio pour améliorer sa performance
- Courir sur un parcours à proximité et rivaliser avec le champion du

parcours

- Suivre les dernières activités de ses amis grâce au widget social sur l'écran d'accueil de son téléphone
- Garder un registre des listes de chansons de chaque séance
- Étudier les parcours à proximité et utiliser la carte pour suivre sa route
- Mettre automatiquement l'application en pause quand l'utilisateur n'est pas en mouvement
- Activer un compte à rebours
- Personnaliser l'écran principal pour afficher l'information qui intéresse le plus l'utilisateur
- Information audio par simple pression sur le bouton media de son casque audio
- Pause et reprise de sa session par pression longue sur le bouton media du casque audio
- Intégration des capteurs de vitesse/cadence et cardio-fréquencemètre par ANT+
- Accéder à tous ses paramètres en ligne : Profil, partage, confidentialité, séance, audio & accessoires
- Informations météo dans le résumé des séances
- Si vous vous identifiez, toutes les données s'enverront automatiquement à votre journal d'entraînement personnel et le réseau social sportif

Le meilleur endroit pour analyser vos entraînements, rivaliser avec vos amis, suivre en direct et communiquer à travers le monde avec les autres sportifs, quel que soit le système GPS qu'ils utilisent.

L'application doit demander l'autorisation pour accéder à vos données personnelles et lire les contacts.

L'objectif des 2 Tps (3-4) est de réaliser en JEE, la partie serveur de cette application.

1/ Faites les modèles métier de votre application. Vous pouvez utiliser des applications en ligne comme genymodel

2/ Identifiez les différents services.

Regroupez ces services aux sein de ressources au sens (REST), faites les choix de mapping des services sur les opérations de base (GET, PUT, DELETE, POST)

Vous pouvez utiliser

<https://chrome.google.com/webstore/detail/dev-http-client/aejoelaoggembcahagimdiliamlcdmfm>

pour tester vos services.

3/ Réalisez l'application, un client lourd en mode texte est suffisant pour tester l'application mais n'hésitez pas à faire mieux.

Voici la liste des dépendances maven et des plugins utiles à votre application.

N'oubliez pas d'inclure un persistence.xml dans un répertoire META-INF dans vos

ressources

N'oubliez pas de lancer votre base de données afin de pouvoir vous y connecter.

```
<dependency>
  <groupId>com.sun.jersey</groupId>
  <artifactId>jersey-server</artifactId>
  <version>${jersey-version}</version>
</dependency>
<dependency>
  <groupId>com.sun.jersey</groupId>
  <artifactId>jersey-json</artifactId>
  <version>${jersey-version}</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-entitymanager</artifactId>
  <version>4.1.7.Final</version>
</dependency>
<dependency>
  <groupId>org.hsqldb</groupId>
  <artifactId>hsqldb</artifactId>
  <version>2.2.8</version>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.21</version>
</dependency>
```

pour les plugins

```
<plugin>
  <groupId>org.apache.tomcat.maven</groupId>
  <artifactId>tomcat7-maven-plugin</artifactId>
  <version>2.1</version>
  <configuration>
```

```
<path>/</path>
</configuration>
</plugin>
```

Si certains sont un peu perdu, je vous ai mis sur share un projet vide configuré convenablement.

N'oubliez pas d'utiliser git ;)

TP 5. Client Web

Créer un répertoire NODE et mettez le dans votre PATH

Télécharger nodejs exe (uniquement).

<http://nodejs.org/download/>

récupérez npm et desintaller le dans votre répertoire node

<http://nodejs.org/dist/npm/npm-1.3.9.zip>

Utilisation avec yeoman.

Dans le répertoire de votre web-app (en ligne de commande).

```
npm install -g yo grunt-cli bower
```

```
npm install -g generator-angular # install generator
yo angular                       # scaffold out a AngularJS project
bower install angular-ui         # install a dependency for your project from Bower
grunt test                       # test your app
grunt server                     # preview your app
grunt
```

arretez grunt.

Mettez à jour maven pour utiliser un plugin spécifique à yeoman.

Je vous ai mis en annexe de ce document les modifications à faire sur le pom.xml.

Je vous fournis aussi sur share, un exemple de projet convenablement configuré.

à l'utilisation,

la compilation yeoman est appelé lors du mvn clean install.
vous pouvez maintenant lancez
mvn jetty:run.

Pas besoin de relancez jetty à chaque modification du js ou de l'html, ce dernier est rechargé à la volée.

Bon TP.

L'objectif sur la partie web est de réaliser quelques pages web de création de comptes, recherche d'amis, recherche de parcours, visualisation de parcours ...

Remarque

Il est important de bien comprendre le pourquoi de chaque action que vous effectuez dans ce TP. Si vous ne voyez pas l'intérêt de certaines de ces actions ou si vous trouvez certaines commandes magiques, n'hésitez pas à poser de nombreuses questions.

Documentation

Les transparents de cours évidemment ...

Annexe

`<plugin>`

```
<groupId>com.github.trecloux</groupId>
<artifactId>yeoman-maven-plugin</artifactId>
<version>0.1</version>
<executions>
  <execution>
    <goals>
      <goal>build</goal>
    </goals>
  </execution>
</executions>
```

`</plugin>`

`<plugin>`

```
<artifactId>maven-war-plugin</artifactId>
<version>2.3</version>
<configuration>
  <webResources>
```

```

        <resource>
            <directory>yo/dist</directory>
        </resource>
    </webResources>
</configuration>
</plugin>
<plugin>
    <artifactId>maven-clean-plugin</artifactId>
    <version>2.5</version>
    <configuration>
        <filesets>
            <fileset>
                <directory>yo/dist</directory>
            </fileset>
            <fileset>
                <directory>yo/.tmp</directory>
            </fileset>
            <fileset>
                <directory>yo/app/components</directory>
            </fileset>
            <fileset>
                <directory>yo/node_modules</directory>
            </fileset>
        </filesets>
    </configuration>
</plugin>
<plugin>
    <groupId>org.apache.tomcat.maven</groupId>
    <artifactId>tomcat7-maven-plugin</artifactId>
    <version>2.1</version>
    <configuration>
        <path>/</path>
    </configuration>
</plugin>
<plugin>

```

```

<groupId>org.mortbay.jetty</groupId>
<artifactId>jetty-maven-plugin</artifactId>
<configuration>
    <scanIntervalSeconds>1</scanIntervalSeconds>
    <webAppSourceDirectory>${basedir}/yo/app</webAppSourceDire
    ctory>
    <webAppConfig>
        <contextPath>/</contextPath>
        <descriptor>${basedir}/src/main/webapp/WEB-INF/web.xml
    </descriptor>
        <!--
    <jettyEnvXml>${basedir}/src/over/here/jetty-env.xml</jettyEnvX
    ml> →
    </webAppConfig>
    <classesDirectory>${basedir}/target/classes</classesDirectory>
    <!-- <scanTargets>
    <scanTarget>src/java/**/*.java</scanTarget>
    <scanTarget>src/myfile.txt</scanTarget>
    </scanTargets> -->
    <scanTargetPatterns>
        <scanTargetPattern>
            <directory>src/main/java/</directory>
            <includes>
                <include>**/*.java</include>
                <include>**/*.properties</include>
            </includes>
            <excludes>
                <exclude>**/myspecial.xml</exclude>
                <exclude>**/myspecial.properties</exclude>
            </excludes>
        </scanTargetPattern>
    <scanTargetPattern>
        <directory>target/classes/</directory>
        <includes>
            <include>**/*.class</include>
            <include>**/*.properties</include>
        </includes>
    </scanTargetPattern>

```



```
<scanTargetPattern>
  <directory>yo/app/</directory>
  <includes>
    <include>*/*</include>
  </includes>
</scanTargetPattern>
</scanTargetPatterns>
</configuration>
</plugin>
```