

Web

TP 1 : site web statique

Téléchargez mongoose (<http://code.google.com/p/mongoose/>), un serveur web extrêmement simple, et placez l'exécutable dans un répertoire vide (par exemple : *staticwebsite*), sur votre disque. Lancez ensuite l'exécutable, et un serveur web sera lancé sur le port 8080. Ce serveur web sert les fichiers placés dans le répertoire *staticwebsite*.

Téléchargez Twitter Bootstrap (<http://twitter.github.io/bootstrap/>), et créez un site web statique pas trop moche, listant une série de spectacles à l'URL `/index.html`.

Un clic sur chaque spectacle ira vers une page à l'URL `/spectacle/x.html` (où x est l'ID du spectacle), qui affichera le détail du spectacle. Un lien sur cette page permettra de revenir à la page d'accueil. La page de détail devra afficher une photo du spectacle, dont l'URL sera `/img/spectacle/x.jpg`.

TP2 : application web : packaging et déploiement

Téléchargez Tomcat 7 (version binaire, fichier zip : <http://tomcat.apache.org/download-70.cgi>). Dézippez-le sur votre disque.

Transformez ensuite votre site web statique réalisé à l'étape 1 en une application web Java EE (fichier war), et déployez cette application dans Tomcat (il suffit pour cela de copier le war dans le répertoire webapps de Tomcat).

Sans l'aide d'un IDE, créez ensuite un servlet. Compilez-le et intégrez-le à votre application. L'API servlet, dont vous aurez besoin pour compiler votre servlet, est livrée avec Tomcat (`lib/servlet-api.jar`). Ce servlet devra répondre à l'URL `.../representation/places?id=xxx`, et renvoyer simplement un nombre de places disponibles (nombre arbitraire) pour la représentation xxx.

TP3 : MVC

Faites un clone du repository github suivant : <https://github.com/Ninja-Squad/tpweb.git>. Intégrez les fichiers de votre webapp à la structure du projet Gradle. Vous pouvez à présent générer votre webapp en utilisant la commande `gradlew war`. Vous pouvez aussi générer un projet Eclipse en utilisant la commande `gradlew eclipse`. Une fois le projet ouvert dans l'IDE, vous pouvez y configurer un serveur tomcat, et y déployer votre webapp directement depuis l'IDE.

Transformez votre webapp afin d'utiliser le pattern MVC.

Créez un servlet permettant de lister les spectacles. Ce servlet créera une liste (en dur) de spectacles, et forwardera vers une JSP affichant la liste des spectacles.

Faites de même pour la liste des représentations, et pour le nombre de places disponibles.

TP4 : Session et formulaire

Pour chaque représentation, ajoutez un lien permettant d'afficher un formulaire de réservation. Ce formulaire contiendra :

- Un champ caché contenant l'ID de la représentation
- Un champ de type number, pré-initialisé à 1, et pouvant aller de 1 à 5.

La soumission du formulaire ajoutera une réservation à un panier de réservations stocké en session.

Chaque page affichera une section contenant un résumé du panier (nombre total de places réservées, et un lien permettant d'afficher ce panier.

TP 5 : réutilisation

Créez un tag pour afficher le panier.