

*Travail en équipe de deux.*

**Dates de remise:**

- **Au plus tard jeudi le 12 décembre 2013 avant 8h00 sur LEA.**
- **Remise électronique uniquement (aucune copie papier).**

---

## **Objectif et présentation général du projet**

---

L'objectif de ce travail est de construire et d'exploiter un service Web pour le projet de réseau social *twitface*. Ce service Web sera fondé sur l'architecture RESTful et utilisera l'API JAX-RS de Java. Les informations proviendront de la base de données *twitface* qui sera accessible seulement à partir du service Web; le client du service Web n'aura pas accès à celle-ci. Une petite application mettant à profit le service Web devra être aussi construite (client du service Web).

---

## **Partie 1 – Construction du service Web REST (60%)**

---

- Vous devez créer un nouveau projet NetBeans (Java EE 6 pour GlassFish) à partir de rien. Le nom du projet et du dossier doit être « **twitface\_ws\_SL\_JPB** » en remplaçant « SL » et « JPB » par vos initiales.
- Pour ce projet, utilisez toujours le nom de *package* suivant sauf pour la classe « RequeteBD.java » :

**VotreDomaineInverse.twitface.ws.rest**

Remplacez "VotreDomaineInverse" par votre nom de domaine créé dans le cadre du cours en ordre inverse; par exemple, "com.dinfogarneau.twitface.ws.rest".

- Dans ce nouveau projet, créez un nouveau **service Web RESTful** (*from Patterns, Simple Root Resource*) ayant « **twitface** » comme *path* pour la classe du service Web; le nom de la classe doit être « **RessourceTwitface** ».
- Créez la classe « **Membre** » (annotée avec JAXB) permettant de conserver les informations suivantes sur un membre : nom, ville d'origine, ville actuelle, courriel et nom d'utilisateur.
- Créez une classe utilitaire appelée « **UtilitaireTwitface** » qui permettra de regrouper ensemble les méthodes statiques permettant l'accès aux données; ceci évitera d'alourdir inutilement la classe pour le service Web.

---

## **Première ressource à créer : Liste des membres en XML**

---

- Ajoutez une ressource dans ce service Web avec la méthode **GET** et le *path* local « **membres** ». Le *path* complet ressemblera à ceci (les paramètres sont optionnels et les valeurs sont fournies à titre indicatif) :

`http://localhost:8080/twitface_ws_SL_JPB/webresources/twitface/membres?debut=0&max=10&nom=car`

- Cette ressource retourne une représentation XML des membres (`ArrayList<Membre>`) correspondant aux critères qui sont spécifiés dans les paramètres de la requête (`@QueryParam`). Les paramètres de la requête sont :
  - « **debut** » : Indice du premier membre à retourner (les membres trouvés sont triés en **ordre croissant de nom**). Ce paramètre est optionnel et sa valeur par défaut est 0.
  - « **max** » : Nombre maximal de membres à retourner. Ce paramètre est optionnel et sa valeur par défaut est 10.
  - « **nom** » : Nom partiel du membre. Ce paramètre est optionnel et s'il n'est pas spécifié tous les membres correspondent aux critères de recherche mais seulement ceux à l'intérieur des limites spécifiées par les paramètres « debut » et « max » sont retournés.
- S'il y a une exception dans le code Java, le service Web doit lancer une « `WebApplicationException` » en ne modifiant pas la cause de l'exception.

---

## Deuxième ressource à créer : Information sur un membre en XML

---

- Ajoutez une ressource dans le service Web avec la méthode **GET** et le *path* local « **membres/{nomUtil}** » où « nomUtil » est un paramètre dans le *path* (`@PathParam`). Le *path* complet ressemblera à ceci (« sackid » est un nom d'utilisateur):  
`http://localhost:8080/twitface_ws_SL_JPB/webresources/twitface/membres/sackid`
- Cette ressource retourne une représentation XML du membre correspondant au nom d'utilisateur dans le *path*.
- S'il y a une exception dans le code Java, le service Web doit lancer une « `WebApplicationException` » en ne modifiant pas la cause de l'exception.
- Si le nom d'utilisateur n'existe pas, le service Web doit lancer une « `WebApplicationException` » de type 404 (*Not Found*).
- Notez que cette ressource ne sera pas utilisée dans la partie 2 de ce travail (client du service Web).

---

## Partie 2 – Construction du client du service Web (40%)

---

- Vous devez créer un nouveau projet NetBeans (Java EE 6 pour GlassFish V3) à partir de rien. Le nom du projet et du dossier doit être « **twitface\_ws\_SL\_JPB\_client** » en remplaçant « SL » et « JPB » par vos initiales.
- Pour ce projet, utilisez toujours le nom de *package* suivant :

**VotreDomaineInverse.twitface.ws.rest.client**

Remplacez "VotreDomaineInverse" par votre nom de domaine créé dans le cadre du cours en ordre inverse; par exemple, "com.dinfoarneau.twitface.ws.rest.client".

- Vous devez construire une mini-application selon l'architecture MVC qui permettra de rechercher des membres en fonction du nom partiel du membre seulement. Par contre, seulement les dix premiers membres (en ordre de nom de membre) devront être retournés pour la première page de résultats.

- Vous devez aussi implanter un mécanisme permettant de contrôler le changement de pages à l'aide d'une **barre de navigation** en JSTL. Dans cette barre de navigation, il doit seulement y avoir des liens pour la **page suivante** et la **page précédente**. Ces **liens** permettant de changer de page doivent seulement être présents lorsque cela est nécessaire.
- Vous devez traiter les trois **paramètres** suivants (qui peuvent être optionnels) dans l'**URL** : **nom**, **debut** et **max**. Ainsi, il sera possible de changer l'URL directement pour tester le bon fonctionnement de votre application Web.
- Le point d'entrée de toute requête à une ressource de cette application Web est la **servlet "Controleur"**.
- Vous devez avoir un modèle qui s'occupe de communiquer avec le service Web pour obtenir les informations nécessaires. Le nom du modèle doit être **"ModeleRechMembres"**.
- Pour l'affichage du formulaire de recherche ainsi que du résultat de recherche (lorsque le formulaire a été soumis uniquement avec la méthode GET), vous devez utiliser une seule vue appelée **"rech-membres.jsp"**.
- Les paramètres de recherche doivent toujours être conservés dans le formulaire de recherche. La **persistance** des paramètres de recherche est de niveau **requête**.
- Vous devez ignorer et enlever les espaces superflus dans le champ du formulaire de recherche.
- Vous ne devez pas utiliser la session utilisateur pour ce projet.
- Ce projet utilisera uniquement la ressource du service Web qui retourne une liste de membres. L'autre ressource du service Web qui retourne l'information sur un membre en fonction de son nom d'utilisateur ne sera pas utilisée.

---

## Directives générales

---

1. Pour les requêtes à la BD à partir de code Java, vous devez utiliser la classe Java « **RequeteBD.java** ».
2. Les nouvelles classes Java créées doivent être placées dans le bon package.
3. Pour la génération des vues, vous devez seulement utiliser EL-JSTL ainsi que les éléments d'actions en JSP. Vous ne pouvez pas utiliser les éléments de scripts en JSP.
4. Respectez les règles de base de la programmation: utilisation de méthodes, utilisation des instructions appropriées, noms des identifiants significatifs, présentation du code, etc.
5. Mettez suffisamment de commentaires dans toutes les parties du code.
6. Portez une attention particulière à la structure de votre programme; pensez à la *parcimonie* et à la *simplicité* avant tout !

---

## Précisions sur l'évaluation

---

Outre les éléments à réaliser dans le cadre du travail, les aspects suivants viendront influencer la note finale:

- Structure du code (modularité, clarté, simplicité, concision, uniformité, etc.)
- Présentation du code: saut de ligne, décalage, etc.
- Nom des identifiants de toutes sortes (variables, méthodes, classes CSS, etc.)
- Commentaires.
- Qualité du français (structure des phrases et fautes).
- D'autres éléments pourraient s'ajouter au besoin.

---

## À remettre

---

- Aucune remise papier.
- Les deux projets créés dans le cadre de ce travail (deux dossiers) doivent être placés dans un dossier principal portant le nom suivant « **TP5\_P1\_NomFamille1\_P2\_NomFamille2** » (utilisez les initiales de vos prénoms et vos noms de famille complet, sans accents et caractères spéciaux).
- Ajoutez un fichier texte portant le nom "**InfoPourProfesseur.txt**" dans le dossier principal qui précise si des éléments n'ont pas pu être implémentés correctement. Le fichier doit être présent même si tout fonctionne correctement.
- **Remettez le dossier compressé sur LÉA**; il est important de conserver le dossier que vous venez de créer lors de la compression des fichiers avant la remise.
- Aucune remise papier.

*Bon travail et bonne fin de session !*