

Nom :
Prénom:

TP n° 10

Ce TP est ramassé et noté
Durée 2h00 (tiers temps additionnel : 40 minutes)

Consignes : Les documents sont autorisés (ainsi que l'accès à Internet et aux corrigés des TP précédents). Les réponses aux questions marquées d'un * doivent être écrites dans le fichier reponses.txt se trouvant dans le projet Eclipse (en bas dans l'arborescence du projet).

Le code doit être rendu via un formulaire de dépôt disponible ici :

https://www.lri.fr/~kn/rendu_ecs.php

Vous devez générer l'archive de rendu depuis Eclipse :

- Export
- General/Archive File
- L'archive doit être au format (.tar.gz) et le nom de l'archive ne doit contenir ni espaces, ni accents. Vérifiez bien que les options « Save in tar format » et « Create directory structure for files » sont sélectionnées.

Vous pouvez utiliser le formulaire autant de fois que vous le souhaitez et toutes les versions soumises seront conservées. Nous corrigerons la version la plus récente. Le site de soumission se verrouille de manière automatique à l'heure indiquée sur la page (fin du TP + 5 minutes). Passé ce délai, il sera **impossible de soumettre**. Il est **vivement conseillé de ne pas soumettre à la dernière minute pour éviter tout problème**.

Notation : Les questions rédigées sont sur 6 points. La qualité du code (commentaires, lisibilité et fait que le code compile) seront sur 4 points. Les 10 points restants seront attribués par des tests automatiques sur votre site Web. **Il est interdit de renommer les packages ou les classes fournis.**

Où manger à San Francisco

Le TP repose sur une base de données de lieux de restauration de la ville de San Francisco (restaurants, cafés, cantines scolaires, ...). La base contient deux tables :

```
CREATE TABLE BUSINESS (bid INTEGER PRIMARY KEY, name VARCHAR(100), address VARCHAR(150))
```

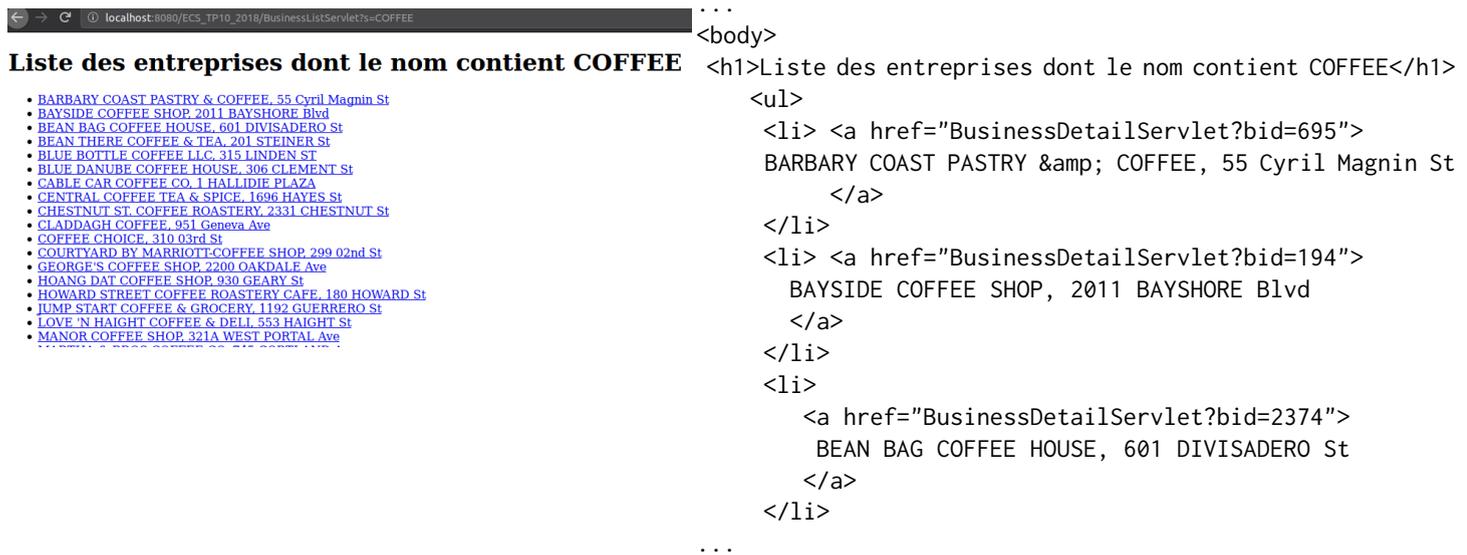
```
CREATE TABLE INSPECTION (bid INTEGER, score INTEGER, date VARCHAR(8),  
reason VARCHAR(100), FOREIGN KEY (bid) REFERENCES BUSINESS)
```

La première table donne la liste des entreprises (business) considérées (leur identifiant unique bid, leur nom name et leur adresse address¹). La seconde liste toutes les inspections de l'agence sanitaire dans ces entreprises. Une inspection est donnée par : le bid de l'entreprise inspectée, le score de l'inspection (entre 0 et 100, 100 signifiant hygiène irréprochable), la date de l'inspection sous la forme d'une chaîne de 8 caractères AAAAMMJJ et la raison (reason) de l'inspection (inspection de routine, plainte d'un client, ...).

Le but de cet exercice est d'utiliser Java/JSP pour faire un affichage des entreprises, puis pour chaque entreprise un affichage détaillé donnant la liste de ses inspections.

1. Attention pour vos requêtes address prend deux « d » en anglais

Affichage en liste (Questions 1.1 à 1.5)

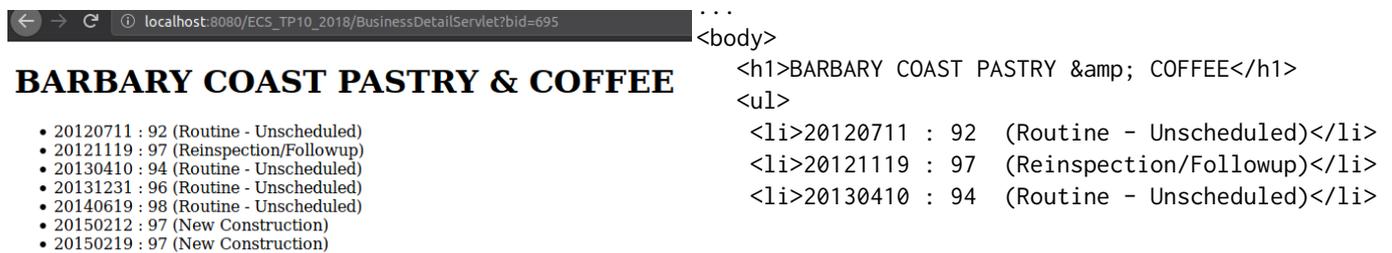


The screenshot shows a web browser at localhost:8080/ECS_TP10_2018/BusinessListServlet?s=COFFEE. The page title is "Liste des entreprises dont le nom contient COFFEE". The browser displays a list of coffee shops with their names and addresses. The HTML code on the right shows the structure of the list, including the title, a list of items, and links to detailed views for each item.

```
...
<body>
<h1>Liste des entreprises dont le nom contient COFFEE</h1>
<ul>
<li> <a href="BusinessDetailServlet?bid=695">
  BARBARY COAST PASTRY & COFFEE, 55 Cyril Magnin St
  </a>
</li>
<li> <a href="BusinessDetailServlet?bid=194">
  BAYSIDE COFFEE SHOP, 2011 BAYSHORE Blvd
  </a>
</li>
<li>
  <a href="BusinessDetailServlet?bid=2374">
    BEAN BAG COFFEE HOUSE, 601 DIVISADERO St
  </a>
</li>
...

```

Affichage détaillé (Questions 1.6 à 1.8)

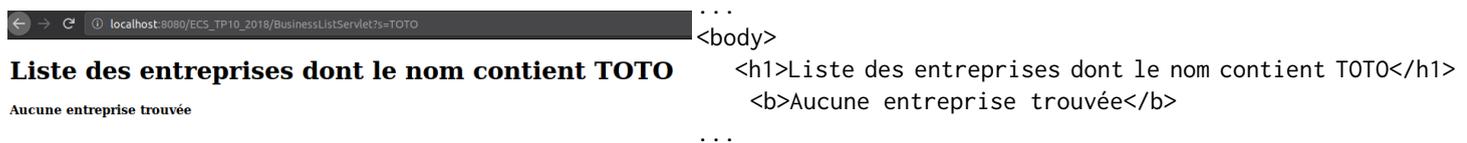


The screenshot shows a web browser at localhost:8080/ECS_TP10_2018/BusinessDetailServlet?bid=695. The page title is "BARBARY COAST PASTRY & COFFEE". The browser displays a list of inspections for this shop, including dates, times, and statuses. The HTML code on the right shows the structure of the detailed view, including the title, a list of items, and the inspection details.

```
...
<body>
<h1>BARBARY COAST PASTRY & COFFEE</h1>
<ul>
<li>20120711 : 92 (Routine - Unscheduled)</li>
<li>20121119 : 97 (Reinspection/Followup)</li>
<li>20130410 : 94 (Routine - Unscheduled)</li>
...

```

Affichage sans résultat (Questions 1.1 à 1.5)



The screenshot shows a web browser at localhost:8080/ECS_TP10_2018/BusinessListServlet?s=TOTO. The page title is "Liste des entreprises dont le nom contient TOTO". The browser displays "Aucune entreprise trouvée". The HTML code on the right shows the structure of the no-results page, including the title and the message.

```
...
<body>
<h1>Liste des entreprises dont le nom contient TOTO</h1>
<b>Aucune entreprise trouvée</b>
...

```

FIGURE 1 – Affichages et codes HTML correspondant

Spécifications L'URL

http://localhost:8080/ECS_TP10_2018_ACOMPLETER/index.html

contient un formulaire permettant de rentrer une chaîne de caractères. Après soumission, une page s'affiche contenant la liste des entreprises dont le nom contient la chaîne donnée, triées par ordre croissant de nom d'entreprise. Chaque entrée de la liste est un lien hyper texte. Cliquer sur le lien hypertexte redirige vers l'affichage détaillé d'une entreprise avec la liste des inspections triées par ordre croissant de date.

(la figure 1 donne le code HTML généré ainsi que l'aperçu). L'affichage doit être **exactement comme détaillé ci-après**. Le non respect de ces consignes entraînera **une perte de points dans les tests automatiques**.

Organisation du code : le code est organisé de la manière suivante :

- une classe Business (**à compléter**) contenant les informations d'une entreprise
- une classe Inspection (**à compléter**) contenant l'information du rôle d'une inspection
- une classe BusinessDB (**à compléter**) encapsulant les interactions avec la base de données
- une classe BusinessListServlet (**à compléter**) représentant le servlet pour l'affichage de la liste des entreprises

une classe `BusinessDetailServlet` (**à compléter**) représentant le servlet pour l'affichage détaillé

un fichier `WEB-INF/jsp/business_list.jsp` (**à compléter**) permettant l'affichage HTML de la liste des entreprises

un fichier `WEB-INF/jsp/business_detail.jsp` (**à compléter**) permettant l'affichage HTML des inspections d'une entreprise.

Exercice 1 Compléter le code aux endroits indiqués. Des informations supplémentaires se trouvent en commentaire dans le code.

1. Rajouter des *getters* à la classe `Business`
2. Rajouter des *getters* à la classe `Inspection`
3. Compléter la méthode `getBusinessByName(String s)` de la classe `BusinessDB`. Cette dernière renvoie un `Vector<Business>` de toutes les entreprises ayant `s` dans leur nom, **triées par ordre de nom croissant. Attention, pour cette requête, vous utiliserez un `Vector<Inspection>` vide pour construire votre objet `Business`.** En d'autres termes, vous n'avez pas, pour cette question à récupérer la liste des inspections pour les entreprises trouvées. La méthode se limite au **100 premiers résultats**. On rappelle la forme générale des requêtes SQL :

```
SELECT ... FROM ... WHERE ... ORDER BY ... LIMIT 100;
```

De plus, on rappelle que pour tester une sous-chaîne `abc` on peut utiliser l'opérateur `LIKE` de la manière suivante :

```
colonne LIKE '%abcd%'
```

dans la clause `WHERE`.

4. Compléter la méthode `doGet` du *servlet* `BusinessListServlet` comme indiqué dans le code.
5. Compléter avec de la syntaxe JSTL le fichier `business_list.jsp` pour qu'il effectue l'affichage demandé.
6. Compléter la méthode `getBusiness(int bid)` de la classe `BusinessDB`. Cette dernière renvoie un `Business` dont le `bid` est passé en argument. Pour cet objet, les inspections doivent être **triées par ordre de date croissantes**. Si un tel `Business` n'existe pas, la méthode renvoie `null`.
7. Compléter la méthode `doGet` du *servlet* `BusinessDetailServlet` comme indiqué dans le code. En particulier, en cas de problème, ce *servlet* effectue une redirection externe vers `index.html`.
8. Compléter avec de la syntaxe JSTL le fichier `business_detail.jsp` pour qu'il effectue l'affichage demandé.

Exercice 2 Répondre aux questions dans le fichier `reponses.txt` se trouvant à côté du fichier `index.html`.

1. * (1.5 point) Dire pour chacun des fichiers `business_detail.jsp`, `BusinessDetailServlet.java` et `BusinessDB.java` s'ils appartiennent à la vue, au modèle ou au contrôleur (sans justifier).
2. * (2.5 points) Expliquer succinctement ce qu'il faudrait modifier pour afficher un message d'erreur explicatif sur la page d'accueil « `index.html` » lorsque l'on effectue une redirection dans le servlet `BusinessDetailServlet`
3. * (2 points) On suppose que la session HTTP expire (donc est détruite) juste après qu'un utilisateur clique sur un lien dans la liste des entreprises générée par `business_list.jsp` (la session est détruite juste avant le chargement de `BusinessDetailServlet`). Cela pose-t'il un problème? Justifier brièvement.