



IUT PARIS  
DESCARTES

# RAPPORT MEDIAWEB

---

Dylan RAKOTOARIVELO (201),  
Julien ZHANG (201)

2020

Dylan RAKOTOARIVELO  
Julien ZHANG

# LE PROJET

---

Le projet consistait à simuler certains services que peuvent proposer une bibliothèque. Plus précisément, il s'agissait de simuler les services : emprunt et retour de documents (réservation n'est pas encore disponible sur cette bêta). Dans le cadre de la matière, il était important de mettre en relation le Backend, le Frontend et la Base de données. Il était également important de mettre en œuvre un découplage strict entre les services servant aux échanges HTTP, la médiathèque contenant en grande partie le domaine stable, et enfin les éléments de persistance des données.

## STRUCTURATION DE CODE

---

Notre architecture est composée de 3 packages structurée ainsi : *Services* -> *Mediatek2020* <- *Persistent*, avec *Mediatek2020* étant un package stable faisant le lien entre les Servlets de *Services* et les requêtes vers la base de données de *Persistent* vers PhpMyAdmin.

(La page suivante vous montre, grâce au software « ObjectAid » la structuration du code en détail)



# SERVLET, JSP, SESSION ET TRANSFORMATION OBJET-RELATIONNEL

---

Chaque classe du package services a son rôle. Il y a actuellement pour cette version bêta 4 classes (Servlets) et chacune fait appel à la base de données via l'intermédiaire d'une méthode de la classe *MédiathequeData.java* ou d'une méthode d'une autre classe du package *Mediatek2020* puis vers une de la classe *MédiathequeData.java*. Voici alors ci-dessous les fonctionnalités de chaque Servlets.

*LoadOnInitServlet.java*

- Initialiser la ***MediathequeData***
- Authentifier
- Créer la **SESSION**
- Maintenir la **SESSION**

*AjoutDocument.java*

- Ajouter des documents

*EmpResDocument.java*

- Emprunter des documents
- Réserver des documents

*RetourDocument.java*

- Rendre des documents

Par ailleurs, chaque Servlet et en plus de la classe *Document.java* du package *Mediatek2020*, sont utilisées dans les JSP qui ont eux ont pour rôle d'être l'IHM. Nous pour l'instant disposons 5 vues JSP :

- *ident.jsp* :
  - *LoadOnInitServlet.java*
- *librarian.jsp* :
  - *AjoutDocument.java*
  - *MediatequeData.java*
- *user.jsp* :
  - *EmpResDocument.java*
  - *MédiathequeData.java*
- *dashboard.jsp* : En fonction du statut de l'utilisateur, inclura *librarian.jsp* s'il s'agit d'une bibliothécaire ou *user.jsp* s'il s'agit d'un abonné.
- *header.jsp* : Entête de la page qui l'inclura

# GESTION DE LA BASE DE DONNEES

---

Pour accéder à la base de données PhpMyAdmin, il est obligatoire de faire appelle à la classe `MediathequeData.java` étant la seule à pouvoir recueillir, ajouter, modifier ou supprimer des données. Pour cela, il a fallu établir une connexion entre celle-ci et notre Base en important le Driver *mysql-connector-java-8.0.15.jar* et en faisant appelle à la fonction static `DriverManager.getConnection(...)` une seule fois lors de l'instanciation de la Classe. Ainsi, pour chaque méthode est attribué une ou plusieurs requêtes selon le contexte. Enfin, pour optimiser ces dernières, les fonctions utilisent toute u `PreparedStatement` qui rendra les requêtes à la fois plus rapides, mais aussi plus lisibles.

## BILAN

---

Les fonctionnalités que nous avons réalisées :

- Authentification
- Création de la SESSION
- Maintien de la SESSION
- Ajouter des documents
- Emprunter des documents
- Rendre des documents
- Messages d'alerte pour la Réservation des documents

Les fonctionnalités que nous n'avons pas réalisées :

- Concurrency
- Fermeture correcte de la SESSION
- Réelle Réservation des documents