



SER - Laboratoire 2 – Une séance

1/ Sérialisation XML avec Java jdom

2/ Sérialisation Json avec Java & Google/Gson

ERIC LEFRANÇOIS - 1^{er} Avril 2016

Table des matières

1	OBJECTIFS DU LABORATOIRE	1
2	RENDU	2
3	CONTRÔLE DU LABO	2
4	LA STRUCTURE DU PROGRAMME PLEXADMIN	3
5	INSTALLATION DE PLEXADMIN ET PRISE EN MAIN	5

1 Objectifs du laboratoire

L'objectif de ce deuxième laboratoire consiste en la génération des documents XML et JSON élaborés dans le cadre du 1^{er} laboratoire.

Dans les deux cas (XML comme JSON), les fichiers obtenus obéiront à un « pretty format » (retours de ligne et indentation des structures).

En particulier, il s'agira de reprendre l'application existante « Plex_Admin » en complétant le code de deux contrôleurs :

Génération du fichier XML

- ➔ Contrôleur `ControleurXMLCreation`, avec comme point d'entrée, - pour la création du fichier XML-, la méthode `createXML()`.

Génération du fichier JSON

- ➔ Contrôleur `ControleurMedia`, avec comme point d'entrée, - pour la création du fichier JSON-, la méthode `sendJSONToMedia()`.

2 Rendu

En début de séance no 4 (début du laboratoire no 3), rendre un rapport papier contenant :

- Petite introduction
- Les modifications éventuelles apportées à vos structures XML et JSON élaborées dans le 1^{er} laboratoire. Décrire alors l'objectif de ces modifications.
- Présenter le code des contrôleurs `ControleurXMLCreation` et `ControleurMedia`
Commentez votre solution (méthodes rajoutées, complétées, leur objectif, points remarquables).
- Présenter un extrait représentatif des deux fichiers que vous aurez générés.
- Bilan - Conclusion

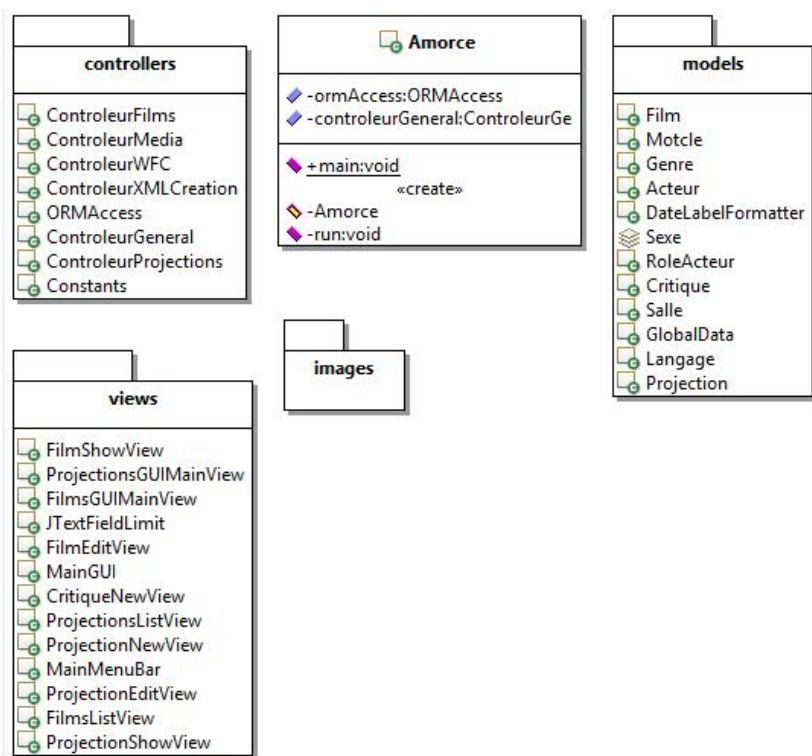
En plus du rapport au format papier, seront communiqués par email :

- Les fichiers source des contrôleurs `ControleurXMLCreation.java` et `ControleurMedia.java`
- Le fichier XML et le fichier JSON qui auront été générés par votre application `PlexAdmin`.

3 Contrôle du labo

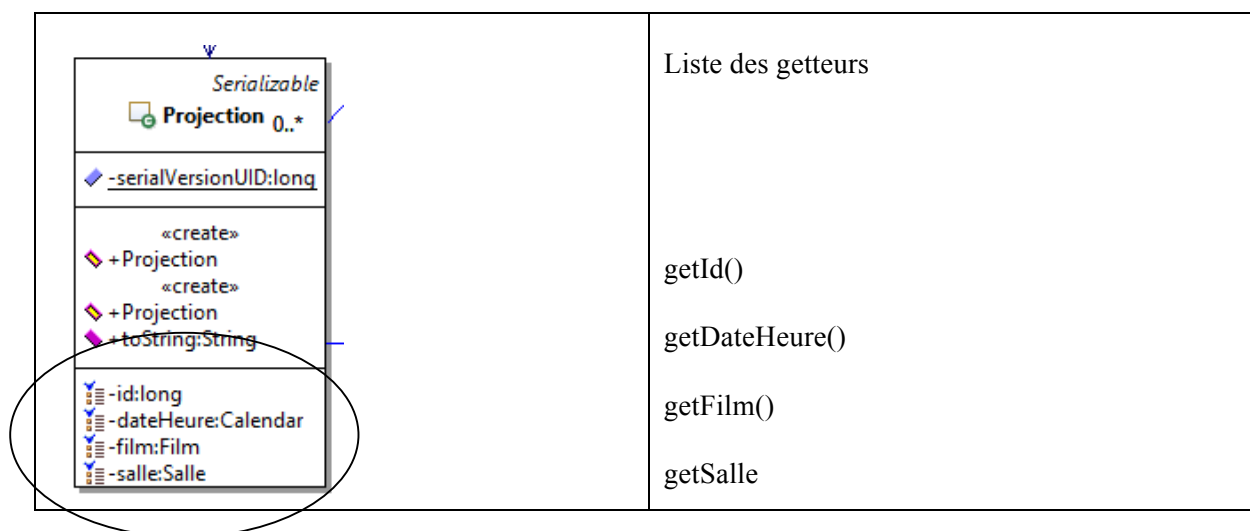
En plus du rendu décrit dans le paragraphe précédent, la génération des fichiers XML et JSON sera présentée au professeur ou à l'assistant (prévoir environ 5 mn de démonstration).

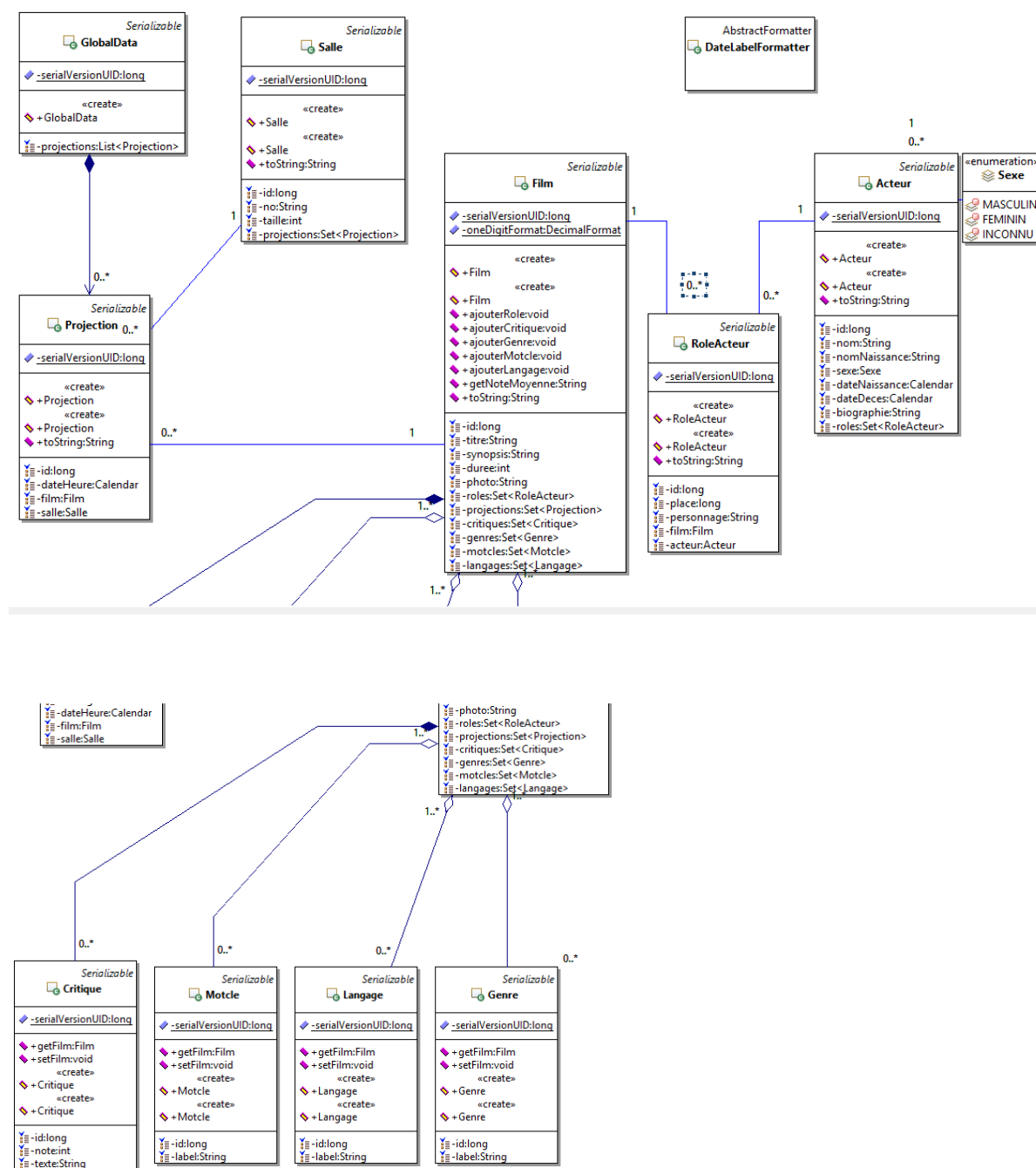
4 La structure du programme PlexAdmin



En particulier, dans les classes « models », la classe `GlobalData` est une structure qui permet d'obtenir la liste des projections actuellement planifiées par l'administrateur du complexe « Pathé-Flon ».

Voir le diagramme de classes présenté ci-après. Chacune des classes comporte une partie inférieure qui décrit la liste de ses « propriétés ». Ces propriétés sont accessibles par le biais d'accesseurs (getters et setters). Par exemple, la class `Projection` possèdent les getters suivants :





Pour générer vos structures XML et Json, vous aurez à disposition une variable de type GlobalData dont la valeur aura été directement chargée depuis la base de données.

Vous pourrez ainsi, pour chacune des projections, obtenir la salle et le film. Pour chacun des films projetés, obtenir la liste des critiques, mots-clés, langages et genres, ainsi que la liste des rôles, associés chacun à un acteur.

⚠ Attention ! Seules les données relatives aux projections actuellement planifiées seront chargées dans la variable GlobalData. Par contre, toute la base de données n'est pas chargée dans cette structure.

Par exemple, si vous planifiez une seule projection, la projection du film « Ocean Eleven », l'acteur Brad Pitt sera présent dans la structure de données. Si vous envoyez le message `getRoles()` à cet acteur, la méthode vous retournera la liste des rôles de Brad Pitt pour d'autres films enregistrés dans la base de données qui ne font pas partie de vos projections actuelles. L'accès à ces différents rôles – non présents dans la structure `GlobalData` -, retournera une erreur de type « Hibernate – Proxy Error – Lazy Access ».

5 Installation de PlexAdmin et prise en main

Dans Eclipse, vous allez importer les deux projets `SER_PlexAdmin` et `SER_WFC_Interface`.

Comme le projet `SER_PlexAdmin`, - qui contient le code que vous allez compléter -, référence une interface décrite dans le projet `SER_WFC_Interface`, le projet `SER_WFC_Interface` doit lui-même être importé.

Décompresser les projets Eclipse `SER_PlexAdmin` et `SER_WFC_Interface`, sur votre bureau ou dans un dossier créé pour l'occasion. Vous avez la possibilité de copier les projets importés dans le workspace d'Eclipse ou alors de travailler directement dans le projet importé.

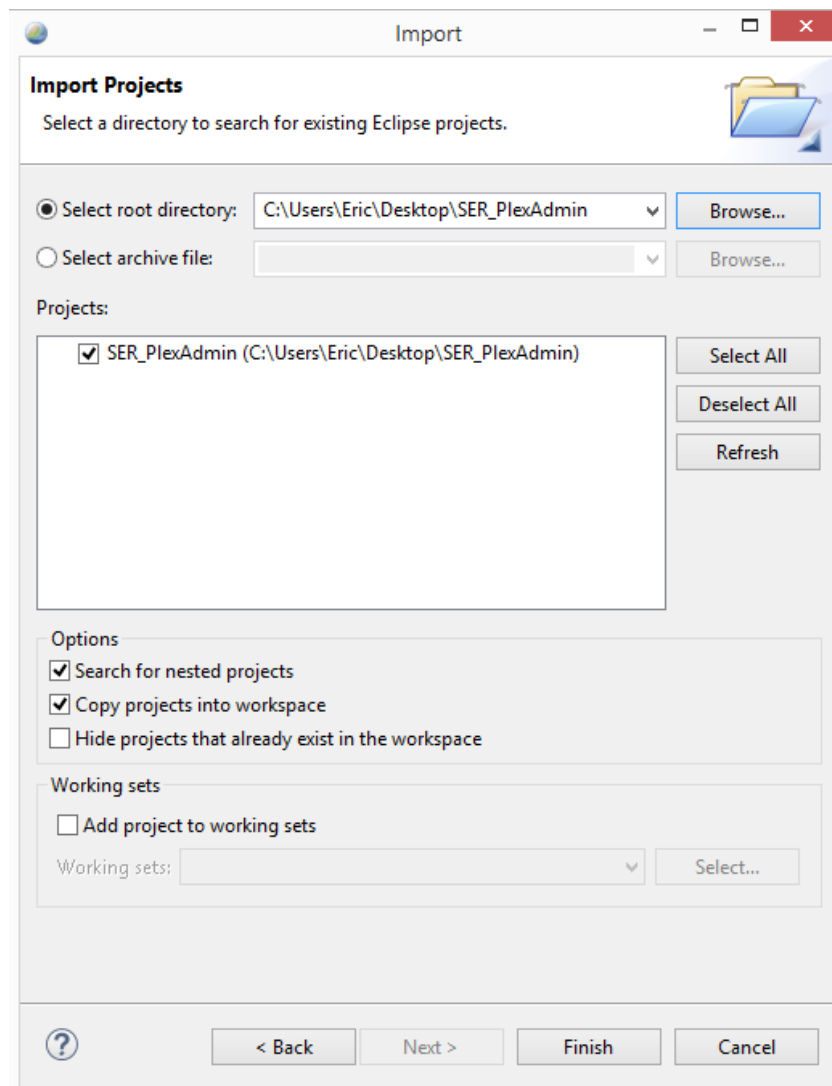
Remarque : suivant la version d'Eclipse que vous possédez, vous pouvez directement importer l'archive sans avoir besoin de la décompresser.



Pour importer un projet dans Eclipse :

File > Import... > General > Existing Projects into Workspace

- Sélectionner `SER_PlexAdmin`
- Typiquement, cocher « Copy projects into workspace » si vous désirez que le projet soit copié dans le workspace de Eclipse.
- Appuyer sur finish
-



Une fois les projets importés, les projets seront copiés dans le workspace de Eclipse.



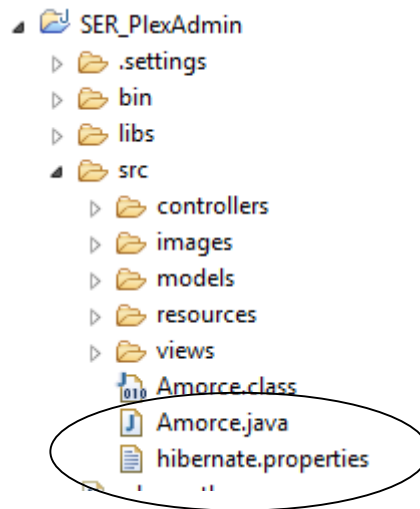
Créer une base de données nommée « plex_admin » dans MySQL

Créer cette base avec un type d'encodage UTF-8 (Collation : UTF-8)



Configurer l'application SER_Plex_Admin pour qu'elle puisse se connecter à votre base de données.

A cette fin, ouvrez le fichier hibernate.properties :



Et configurez les valeurs des champs `hibernate.connection.username` & `hibernate.connection.password` avec le username et le password de votre connexion MySQL.

```
hibernate.connection.driver_class=com.mysql.jdbc.Driver
hibernate.connection.url=jdbc:mysql://localhost:3306/plex_admin?
hibernate.connection.username=votre_user_name
hibernate.connection.password=votre_password
hibernate.connection.pool_size=1
hibernate.dialect=org.hibernate.dialect.MySQLDialect
hibernate.show_sql=false
hibernate.hbm2ddl.auto=update
hibernate.connection.isolation=2
```



Pour compiler et exécuter le projet, créer une configuration d'exécution en sélectionnant le projet SER_PlexAdmin dont la classe Amorces sera choisie comme point d'entrée (fonction main).



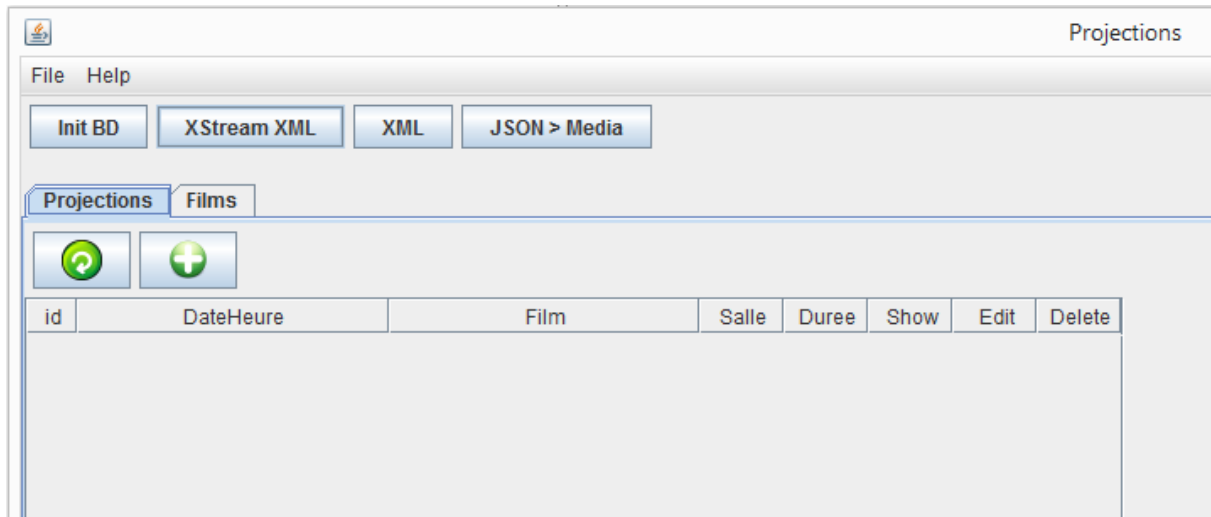
Exécuter le programme.

Viendrons alors s'afficher en console quantité de messages du style :

```
mars 27, 2015 5:16:31 PM org.hibernate.tool.hbm2ddl.DatabaseMetadata getTableMetadata
INFO: HHH000262: Table not found: films
mars 27, 2015 5:16:31 PM org.hibernate.tool.hbm2ddl.DatabaseMetadata getTableMetadata
INFO: HHH000262: Table not found: genres
mars 27, 2015 5:16:31 PM org.hibernate.tool.hbm2ddl.DatabaseMetadata getTableMetadata
INFO: HHH000262: Table not found: languages
mars 27, 2015 5:16:31 PM org.hibernate.tool.hbm2ddl.DatabaseMetadata getTableMetadata
INFO: HHH000262: Table not found: motcles
mars 27, 2015 5:16:31 PM org.hibernate.tool.hbm2ddl.DatabaseMetadata getTableMetadata
INFO: HHH000262: Table not found: projections
mars 27, 2015 5:16:31 PM org.hibernate.tool.hbm2ddl.DatabaseMetadata getTableMetadata
INFO: HHH000262: Table not found: roles
mars 27, 2015 5:16:31 PM org.hibernate.tool.hbm2ddl.DatabaseMetadata getTableMetadata
INFO: HHH000262: Table not found: salles
mars 27, 2015 5:16:33 PM org.hibernate.tool.hbm2ddl.SchemaUpdate execute
INFO: HHH000232: Schema update complete
```

Ne pas s'inquiéter ! Le programme cherche des tables dans la base de données qui n'existent pas encore. Il les crée alors automatiquement.

Vous devez obtenir dès lors l'interface suivante :

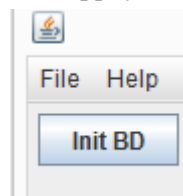


En l'état, vous pouvez constater que les tables ont été créées dans la base de données (salles, projections, films, etc.).

Ces tables sont vides.



Pour les remplir avec les données initiales, appuyez sur le bouton « Init BD »



Cette action a pour effet d'effacer tout le contenu des tables et de les peupler avec le contenu de deux fichiers XML qui se trouvent dans le répertoire « resources » de votre projet : Les fichiers movies_v1.xml et persons_v1.xml



































En console, vous obtenez l'affichage suivant :

```
INFO: HHH000232: Schema update complete
Chargement du contenu XML...
Chargement du contenu XML: OK
Enregistrement des films ...
Enregistrement des films: OK
Enregistrement des acteurs ...
Enregistrement des acteurs: OK
Enregistrement des roles ...
Enregistrement des roles: OK
```

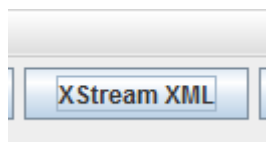

Vous pouvez répéter cette opération(Init BD) autant de fois que le désirez !



En allant dans l'onglet « Films », vous voyez alors apparaître une dizaine de films :

Projections							
File Help							
<div>Init BD XStream XML XML JSON > Media</div>							
<div>Projections Films</div>							
<div>  </div>							
id	Titre	Duree	Photo	Note	Show	Edit	Critique
2053403	Casino Royale	144		-			
2136717	Django Unchained	165		-			
2235127	GoldenEye	130		-			
2304289	Inglourious Basterds	153		-			
2510706	Ocean's Eleven	116		-			
2510708	Ocean's Thirteen	122		-			
2510709	Ocean's Twelve	125		-			
2576878	Quantum of Solace	106		-			
2652003	Skyfall	143		-			
2806844	The World Is Not Enough	128		-			
2822603	Tomorrow Never Dies	119		-			

Appuyez maintenant sur le bouton XStream XML :



Est alors créé un fichier Global_data.xml dans la racine de votre projet qui contient la liste des projections actuelles.



Ce fichier est obtenu en opérant la sérialisation d'un objet Java qui contient la liste actuelle des projections. Cette sérialisation est obtenue en utilisant la librairie XStream (le pendant de la librairie Google/Gson pour XML).

Voir le code de la méthode createXStreamXML du contrôleur ControleurXMLCreation.



Amusez-vous avec l'application, ajoutez quelques commentaires critiques sur vos films et créez une ou deux projections..

Puis appuyez à nouveau sur le bouton XStream XML.



Vous pouvez alors passer à la réalisation de votre laboratoire.

La pression des boutons XML et JSON > Media devront avoir pour effet, respectivement, de générer le fichier XML et le fichier Json.



La pression du bouton XML invoque la méthode `createXML()` du contrôleur `ControleurXMLCreation`.

La pression du bouton JSON > Media invoque la méthode `sendJSONToMedia ()` du contrôleur `ControleurMedia`.