

Dans cette page, vous pourrez y trouver les pages qui semblaient les plus intéressantes pour notre projet :

- **OrthancException.cpp** : Déclare les évènements erreurs qu'on pourraient utiliser.
- **DicomTag.cpp/h** : Déclaration des tags qui seront utilisés dans Orthanc.
- **DicomMap.cpp** : Placement des tags dans les différentes catégories (patient, study, series, instance), ainsi que la déclaration pour savoir comment récupérer les tags.
- **Dossier OrthancExplorer** : Contient le coté client d'Orthanc avec les fichiers JS et HTML.
- **DatabaseWrapper.cpp** : Requêtes SQL
- **OrthancFindRequestHandler.cpp** : Va chercher les tags recherchés.
- **OrthancInitialization.cpp** : Définit l'emplacement de la BDD.
- **DicomUserConnection.cpp** : Connexion au SCU.
- **DicomServer.cpp** : C'est le SCP.
- **Main.cpp** : Certains réglages sont fait comme pour l'AE Title, ou encore le Dicomport, ainsi que le load des plugins (Par défaut Orthanc) :

```
466 // DICOM server
467 DicomServer dicomServer;
468 OrthancApplicationEntityFilter dicomFilter(context);
469 dicomServer.SetCalledApplicationEntityTitleCheck(Configuration::GetGlobalBoolParameter("DicomCheckCalledAet", false));
470 dicomServer.SetStoreRequestHandlerFactory(serverFactory);
471 dicomServer.SetMoveRequestHandlerFactory(serverFactory);
472 dicomServer.SetFindRequestHandlerFactory(serverFactory);
473 dicomServer.SetPortNumber(Configuration::GetGlobalIntegerParameter("DicomPort", 4242));
474 dicomServer.SetApplicationEntityTitle(Configuration::GetGlobalStringParameter("DicomAet", "ORTHANC"));
475 dicomServer.SetApplicationEntityFilter(dicomFilter);
```

- **Tous les fichiers .sql** : Structure SQL.
 - PrepareDatabase.sql
 - Upgrade3to4.sql
 - Upgrade4to5.sql
- **OrthancRestResources.cpp** : C'est ici que sont déclarés les URIs qui seront appelés par l'API REST.
- **Plugin.c** : C'est un exemple de plugin qui peut aider à utiliser des fonctions comme par exemple POST.

Pour le plugin **WebViewer** qu'on s'inspire pour 4vet :

- **OrthancExplorer.js** : c'est ici qu'est déclaré la page d'accès lorsqu'on clique sur le bouton Webviewer.
- **CacheManager.cpp** : c'est ici qu'est déclaré la structure de la BDD
- **Plugin.cpp** : partie principal du plugin, y est déclaré le dossier de la BDD, de plus possède les 4 fonctions importantes d'un plugin qui peut servir d'exemple.