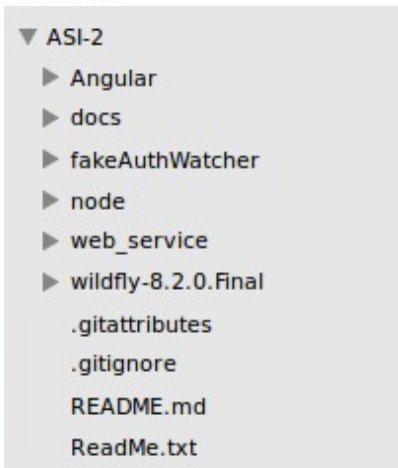


**Projet ASI-2 :**

**Mise en place d'un diaporama en ligne**

## I] Introduction

Soit la structure du projet :



**Angular** : Application Web basé sur Angular.js

**docs** : contient toutes nos remarques sur les différentes parties du projet.

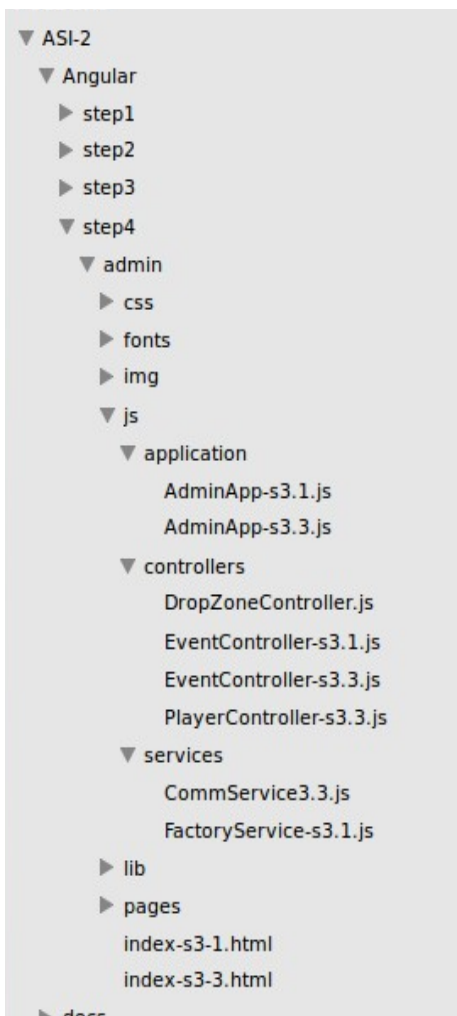
**Node** : Communication côté serveur.

**web\_service** : Authentification via J2E.

**wildfly-8.2.0.Final** : contient tous les fichiers permettant la mise en place d'un serveur avec Wildfly.

## II] Angular

Dans le dossier réservé aux modules Angular.js, nous avons séparé les dossiers en 4 steps, correspondant chacun à ce que nous devons réaliser d'après l'énoncé.



**Css** : contient les fichiers .css pour la mise en page (notamment les bibliothèques bootstrap)

**fonts** : contient les glyphicons

**img** : contient les différentes images à afficher sur la page web

**js** : contient les fichiers javascript, où sont codées les fonctions angular.js. Ce dossier est segmenté en 3 sous-dossiers :

- **application** : qui correspond au main de notre application

- **controllers** : qui alimentent le scope par des variables ou des fonctions. DropZoneController traite le drag&drop des images de la barre latérale. EventController alimente le scope des différentes slides. PlayerController traite du défilement du diaporama.

- **services** : fournit des fonctionnalités au controller. CommService traite du chargement et de la sauvegarde d'images. FactoryService traite la création de contenu (crée de nouvelles slides, génère les ID uniques...).

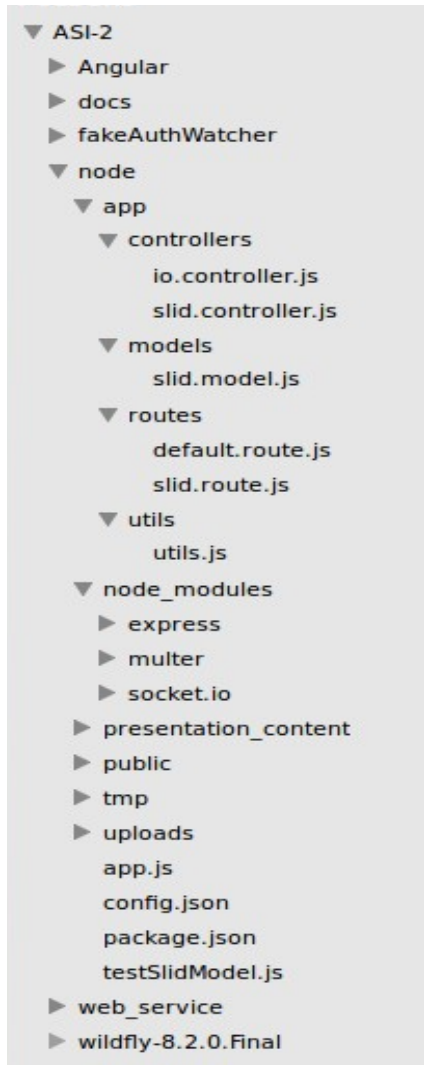
**lib** : Contient les librairies : JQuery, ngDraggable, Bootstrap...

**pages** : Contient la page html de la dropZoneArea.

Enfin, nous avons la page html index, qui correspond à la vue de ce diaporama en ligne.

### III] Node

Dans le dossier Node, nous avons aussi séparé les logiques en différents sous-dossiers :



**app** : dossier séparant les différentes logiques :

- **controllers** : contient les controllers de l'application. io.controller.js traite de la connexion par socket. slid.controller.js crée, lit et écrit les slides

- **models** : liée à la représentation des données, on parse le fichier json pour l'envoyer à la vue.

- **routes** : sous-dossier lié au routage des pages web. Default.route.js définit la route par défaut, et slid.route.js s'occupe du renvoi vers une certaine page en fonction de l'URL.

- **utils** : contient utils.js qui gère la métadonnée.

**node\_modules** : contient tous les modules de Node, installés via l'outil npm. Ici, nous avons importé expres, multer et socket.io

**presentation\_content** : contient un fichier json contenant toute la présentation à afficher.

**Public** : contient la step4 de la partie angular précédente.

**Tmp** : censé contenir les fichiers temporaires.

**Uploads** : contient les images en ligne que l'on peut afficher drag&drop dans le diaporama.

Aussi, **app.js** définit les routes statiques ; **config.json** définit les configurations (numéro de port, dossiers d'upload) ; et enfin **package.json** définit les paramètres de l'application (dépendances, numéro version, nom...).

### **III] Fonctionnalités de l'application Web**

L'application web que nous avons réalisé permet à l'utilisateur de concevoir un diaporama en ligne en temps réel.

Avant d'effectuer cela, l'utilisateur doit se connecter via un formulaire de login, pour s'assurer de son authentification.

Il est ainsi possible d'ajouter des images aux slides, du contenu et un titre.

Il existe aussi un système d'upload d'images dans le serveur, si l'utilisateur souhaite ajouter d'autres images que celles qui sont déjà en place par défaut.

Les raccourcis clavier ont été implémentés, c'est à dire que l'on peut parcourir les slides via les touches flèches du clavier, afin de faciliter l'expérience utilisateur.

Des boutons en haut de la page ont aussi été conçus pour :

- ajouter une slide,
- supprimer une slide,
- passer à la slide suivante,
- passer à la slide précédente,
- passer à la première slide,
- passer à la dernière slide,
- faire défiler le diaporama des slides selon un intervalle de temps régulier,
- mettre en pause cette présentation.,
- sauvegarder la présentation.

Une barre horizontale de recherche a été mise en place afin de chercher une image par son nom.