

Rapport du projet de JEE - Medialibs

Nicolas GILLE

Version 0.2, 03 janvier 2018

Architectures

```
interface IMedia
class Media
class Book
class Comic
class VideoGame
class Album
class Anime
class Cartoon
class Movie
class Series
class Video

IMedia <|-- Media
Media <|-- Video
Media <|-- Book
Media <|-- Comic
Media <|-- VideoGame
Media <|-- Album
Video <|-- Anime
Video <|-- Cartoon
Video <|-- Movie
Video <|-- Series
```

Présentation des Interfaces de Services

Présentation du service d'authentification

Table 1. Service d'authentification

Méthode	URL	Body	Description	Code de retour
POST	/auth	User	Permet de se connecter à l'application	200 en cas de réussite lors de l'inscription. 204 en cas d'absence de l'utilisateur dans la base de données. 500 en cas de problème majeure.
POST	/register	User	Permet de s'inscrire sur à l'application avec un rôle d'utilisateur	201 en cas de réussite lors de l'enregistrement de l'utilisateur. 409 en cas de conflit d'existence.

Présentation du service de gestion utilisateur

Table 2. Service de gestion utilisateurs

Méthode	URL	Body	Description	Code de retour
GET	/users/	/	Récupération de l'ensemble des utilisateurs inscrits.	200 en cas de succès de la requête. 204 en cas de liste vide.
GET	/users/{id}	/	Récupération des informations d'un utilisateur unique en fonction de son identifiant.	200 en cas de succès de la requête. 204 en cas d'utilisateur inconnu.
PUT	/users/edit/{id}	User	Modification des informations d'un utilisateur passé dans le corps de la requête et pour identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si l'utilisateur n'existe pas.
DELETE	/users/delete/{id}	User	Suppression de l'utilisateur ayant pour identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si l'utilisateur devant être supprimé n'existe pas.

Présentation des services de gestion des médias

Service de gestion des Livres

Table 3. Service de gestion des livres

Méthode	URL	Body	Description	Code de retour
GET	/books/	/	Récupération de l'ensemble des livres présent dans la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence de livres présent dans la médiathèque.
GET	/books/{id}	/	Récupération des informations d'un livre en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence de livre ayant pour identifiant <i>id</i> .
GET	/books/title/{title}	/	Récupération des informations d'un livre en fonction de son <i>titre</i> .	200 en cas de succès de la requête. 204 en cas d'absence de livre ayant pour titre <i>title</i> .
POST	/books/	Book	Ajout d'un nouveau livre dans la médiathèque.	201 en cas de succès de la requête. 409 en cas d'échec de l'ajout du livre dans la médiathèque.
PUT	/books/{id}	Book	Modification des informations d'un livre possédant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si le livre à modifier n'existe pas.

Méthode	URL	Body	Description	Code de retour
DELETE	/books/{id}	Book	Suppression du livre ayant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si le livre a supprimer n'existe pas.
GET	/books/genres/	/	Retourne la liste de tout les genres de livres disponible dans la médiathèque.	Renvoie toujours un code 200.
GET	/books/formats/	/	Retourne la liste de tout les formats de livres disponible dans la médiathèque.	Renvoie toujours un code 200.

Services de gestions des Bandes Dessinées / Comics / Mangas

Table 4. Service de gestion des bandes dessinées, comics et mangas

Méthode	URL	Body	Description	Code de retour
GET	/comics/	/	Récupération de l'ensemble des bandes dessinées, comics et mangas présent dans la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence de bande dessinés, comics ou mangas présent dans la médiathèque.
GET	/comics/{id}	/	Récupération des informations d'une bandes dessinées, comics ou mangas en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence de bande dessinée, comic ou manga ayant pour identifiant <i>id</i> .
GET	/comics/title/{title}	/	Récupération des informations d'une bande dessinée, comic ou manga en fonction de son <i>titre</i> .	200 en cas de succès de la requête. 204 en cas d'absence de bande dessinée, comic ou manga ayant pour titre <i>title</i> .
POST	/comics/	Comic	Ajout d'une nouvelle bande dessinée, comic ou manga dans la médiathèque.	201 en cas de succès de la requête. 409 en cas d'échec de l'ajout de la bande dessinée, du comic ou du manga dans la médiathèque.
PUT	/comics/{id}	Comic	Modification des information de la bande dessinée, comic ou manga possédant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si la bande dessinée, du comic ou du manga a mettre à jour n'existe pas.

Méthode	URL	Body	Description	Code de retour
DELETE	/comics/{id}	Comic	Suppression de la bande dessinée, comic ou manga ayant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si la bande dessinée, le comic ou le manga a supprimer n'existe pas.
GET	/comics/genres/ /		Retourne la liste de tout les genres de bandes dessinées, comics ou mangas disponible dans la médiathèque.	Renvoie toujours un code 200.
GET	/comics/format s/	/	Retourne la liste de tout les formats de bandes dessinées, comics ou mangas disponible dans la médiathèque.	Renvoie toujours un code 200.

Service liés aux Jeux Vidéos

Table 5. Service de gestion des jeux vidéos

Méthode	URL	Body	Description	Code de retour
GET	/video-games/	/	Récupération de l'ensemble des jeux vidéos présent dans la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence de jeux vidéos présent dans la médiathèque.
GET	/video-games/{id}	/	Récupération des informations d'un jeu vidéo en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence de jeu vidéo ayant pour identifiant <i>id</i> .
GET	/video-games/title/{titre}	/	Récupération des informations d'un jeu vidéo en fonction de son <i>titre</i> .	200 en cas de succès de la requête. 204 en cas d'absence de jeu vidéo ayant pour titre <i>titre</i> .
POST	/video-games/	VideoGame	Ajout d'un nouveau jeu vidéo dans la médiathèque.	201 en cas de succès de la requête. 409 en cas d'échec de l'ajout du jeu vidéo dans la médiathèque.
PUT	/video-games/{id}	VideoGame	Modification des information d'un jeu vidéo possédant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si le jeu vidéo a mettre à jour n'existe pas.

Méthode	URL	Body	Description	Code de retour
DELETE	/video-games/{id}	VideoGame	Suppression du jeu vidéo ayant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si le jeu vidéo a supprimer n'existe pas.
GET	/video-games/genres/	/	Retourne la liste de tout les genres de jeux vidéos disponible dans la médiathèque.	Renvoie toujours un code 200.
GET	/video-games/platforms/	/	Retourne la liste de toutes les plates-formes de jeux vidéos disponible dans la médiathèque.	Renvoie toujours un code 200.

Services de gestions des Albums musicaux

Table 6. Service de gestion des albums de musique

Méthode	URL	Body	Description	Code de retour
GET	/musics/	/	Récupération de l'ensemble des albums de musique présent dans la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence d'albums de musiques présent dans la médiathèque.
GET	/musics/{id}	/	Récupération des informations d'un album de musique en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'album ayant pour identifiant <i>id</i> .
GET	/musics/title/{title}	/	Récupération des informations d'un album de musique en fonction de son <i>titre</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'album ayant pour titre <i>title</i> .
POST	/musics/	Album	Ajout d'un nouvel album de musique dans la médiathèque.	201 en cas de succès de la requête. 409 en cas d'échec de l'ajout d'un nouvel album dans la médiathèque.
PUT	/musics/{id}	Album	Modification des information d'un album de musique possédant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si l'album a mettre à jour n'existe pas.
DELETE	/musics/{id}	Album	Suppression de l'album de musique ayant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si l'album a supprimer n'existe pas.

Méthode	URL	Body	Description	Code de retour
GET	/musics/genres/ /		Retourne la liste de tout les genres musical disponible dans la médiathèque.	Renvoie toujours un code 200.

Services de gestions des Dessins Animés

Table 7. Service de gestion des dessins animés

Méthode	URL	Body	Description	Code de retour
GET	/cartoons/	/	Récupération de l'ensemble des dessins animés présent dans la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence de dessin animés présent dans la médiathèque.
GET	/cartoons/{id}	/	Récupération des informations d'un dessin animé en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence de dessin animé ayant pour identifiant <i>id</i> .
GET	/cartoons/title/{title}	/	Récupération des informations d'un dessin animé en fonction de son <i>titre</i> .	200 en cas de succès de la requête. 204 en cas d'absence de dessin animé ayant pour titre <i>title</i> .
POST	/cartoons/	Cartoon	Ajout d'un nouveau dessin animé dans la médiathèque.	201 en cas de succès de la requête. 409 en cas d'échec de l'ajout du dessin animé dans la médiathèque.
PUT	/cartoons/{id}	Cartoon	Modification des information d'un dessin animé possédant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si le dessin animé a mettre à jour n'existe pas.
DELETE	/cartoons/{id}	Cartoon	Suppression du dessin animé ayant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si le dessin animé a supprimer n'existe pas.
GET	/cartoons/genres/	/	Retourne la liste de tout les genres de dessins animés disponible dans la médiathèque.	Renvoie toujours un code 200.

Services de gestions des Films

Table 8. Service de gestion des films

Méthode	URL	Body	Description	Code de retour
GET	/movies/	/	Récupération de l'ensemble des films présent dans la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence de films présent dans la médiathèque.
GET	/movies/{id}	/	Récupération des informations d'un film en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence de film ayant pour identifiant <i>id</i> .
GET	/movies/title/{title}	/	Récupération des informations d'un film en fonction de son <i>titre</i> .	200 en cas de succès de la requête. 204 en cas d'absence de film ayant pour titre <i>title</i> .
POST	/movies/	Movie	Ajout d'un nouveau film dans la médiathèque.	201 en cas de succès de la requête. 409 en cas d'échec de l'ajout du film dans la médiathèque.
PUT	/movies/{id}	Movie	Modification des informations d'un film possédant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si le film à mettre à jour n'existe pas
DELETE	/movies/{id}	Movie	Suppression du film ayant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si le film à supprimer n'existe pas.
GET	/movies/genres /	/	Retourne la liste de tous les genres de films disponible dans la médiathèque.	Renvoie toujours un code 200.

Services de gestion des Japanimes

Table 9. Service de gestion des japanimes

Méthode	URL	Body	Description	Code de retour
GET	/animes/	/	Récupération de l'ensemble des animes présent dans la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence d'anime présent dans la médiathèque.
GET	/animes/{id}	/	Récupération des informations d'un anime en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'anime ayant pour identifiant <i>id</i> .

Méthode	URL	Body	Description	Code de retour
GET	/animes/title/{title}	/	Récupération des informations d'un anime en fonction de son <i>titre</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'anime ayant pour titre <i>title</i> .
GET	/animes/title/{title}/{currentSeason}	/	Récupération des informations d'un anime en fonction de son <i>titre</i> et de sa <i>saison actuelle</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'anime ayant pour titre <i>title</i> et pour saison actuelle <i>currentSeason</i> .
POST	/animes/	Anime	Ajout d'un nouvel anime dans la médiathèque.	201 en cas de succès de la requête. 409 en cas d'échec de l'ajout du nouvel anime dans la médiathèque.
PUT	/animes/{id}	Anime	Modification des informations d'un anime possédant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si l'anime a été mis à jour n'existe pas
DELETE	/animes/{id}	Anime	Suppression de l'anime ayant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si l'anime a été supprimé n'existe pas.
GET	/animes/genres	/	Retourne la liste de tous les genres d'animes disponibles dans la médiathèque.	Renvoie toujours un code 200.

Services de gestion des Séries

Table 10. Service de gestion des séries

Méthode	URL	Body	Description	Code de retour
GET	/series/	/	Récupération de l'ensemble des séries présentes dans la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence de séries présentes dans la médiathèque.
GET	/series/{id}	/	Récupération des informations d'une série en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence de série ayant pour identifiant <i>id</i> .
GET	/series/title/{title}	/	Récupération des informations d'une série en fonction de son <i>titre</i> .	200 en cas de succès de la requête. 204 en cas d'absence de série ayant pour titre <i>title</i> .

Méthode	URL	Body	Description	Code de retour
GET	/series/title/{titre}/{currentSeason}	/	Récupération des informations d'une série en fonction de son <i>titre</i> et de sa <i>saison actuelle</i> .	200 en cas de succès de la requête. 204 en cas d'absence de série ayant pour titre <i>titre</i> et pour saison actuelle <i>currentSeason</i> .
POST	/series/	Series	Ajout d'une nouvelle série dans la médiathèque.	201 en cas de succès de la requête. 409 en cas d'échec de l'ajout de la série dans la médiathèque.
PUT	/series/{id}	Series	Modification des informations d'une série possédant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si la série a été mise à jour n'existe pas
DELETE	/series/{id}	Series	Suppression de la série ayant l'identifiant <i>id</i> .	200 en cas de succès de la requête. 204 si la série a été supprimée n'existe pas.
GET	/series/genres/	/	Retourne la liste de tous les genres de série disponibles dans la médiathèque.	Renvoie toujours un code 200.

Présentation des services liés aux personnes relatives aux médias

Présentation du service de gestion des acteurs

Table 11. Service de gestion des acteurs

Méthode	URL	Body	Description	Code de retour
GET	/actors/	/	Récupération de la liste de tous les acteurs enregistrés sur la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence d'acteurs présents dans la médiathèque.
GET	/actors/{id}	/	Récupération des informations d'un acteur en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'acteur ayant pour identifiant <i>id</i> .

Méthode	URL	Body	Description	Code de retour
GET	/actors/search/? / fname={fname} &lname={lname}		Récupération des informations d'un acteur en fonction de son <i>nom</i> et son <i>prénom</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'acteur ayant pour nom <i>lname</i> et pour prénom <i>fname</i> .

Présentation du service de gestion des auteurs

Table 12. Service de gestion des auteurs

Méthode	URL	Body	Description	Code de retour
GET	/authors/	/	Récupération de la liste de tout les auteurs enregistrés sur la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence d'auteurs présent dans la médiathèque.
GET	/authors/{id}	/	Récupération des informations d'un auteur en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'auteur ayant pour identifiant <i>id</i> .
GET	/authors/search / /?fname={fname} &lname={lname}		Récupération des informations d'un auteur en fonction de son <i>nom</i> et son <i>prénom</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'auteur ayant pour nom <i>lname</i> et pour prénom <i>fname</i> .

Présentation du service de gestion des directeurs

Table 13. Service de gestion des directeurs

Méthode	URL	Body	Description	Code de retour
GET	/directors/	/	Récupération de la liste de tout les directeurs enregistrés sur la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence directeurs présent dans la médiathèque.
GET	/directors/{id}	/	Récupération des informations d'un directeur en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence directeur ayant pour identifiant <i>id</i> .
GET	/directors/search / /?fname={fname} &lname={lname}		Récupération des informations d'un directeur en fonction de son <i>nom</i> et son <i>prénom</i> .	200 en cas de succès de la requête. 204 en cas d'absence directeur ayant pour nom <i>lname</i> et pour prénom <i>fname</i> .

Présentation du service de gestion des illustrateurs

Table 14. Service de gestion des illustrateurs

Méthode	URL	Body	Description	Code de retour
GET	/illustrators/	/	Récupération de la liste de tout les illustrateurs enregistrés sur la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence d'illustrateurs présent dans la médiathèque.
GET	/illustrators/{id}	/	Récupération des informations d'un illustrateur en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'illustrateur ayant pour identifiant <i>id</i> .
GET	/illustrators/search/?fname={fname}&lname={lname}	/	Récupération des informations d'un illustrateur en fonction de son <i>nom</i> et son <i>prénom</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'illustrateur ayant pour nom <i>lname</i> et pour prénom <i>fname</i> .

Présentation du service de gestion des producteurs

Table 15. Service de gestion des producteurs

Méthode	URL	Body	Description	Code de retour
GET	/producers/	/	Récupération de la liste de tout les producteurs enregistrés sur la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence de producteurs présent dans la médiathèque.
GET	/producers/{id}	/	Récupération des informations d'un producteur en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence de producteur ayant pour identifiant <i>id</i> .
GET	/producers/search/?fname={fname}&lname={lname}	/	Récupération des informations d'un producteur en fonction de son <i>nom</i> et son <i>prénom</i> .	200 en cas de succès de la requête. 204 en cas d'absence de producteur ayant pour nom <i>lname</i> et pour prénom <i>fname</i> .

Présentation du service de gestion des chanteurs et groupes

Table 16. Service de gestion des chanteurs et groupes

Méthode	URL	Body	Description	Code de retour
GET	/singers/	/	Récupération de la liste de tout les chanteurs et groupes enregistrés sur la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence de chanteurs et/ou groupes présent dans la médiathèque.
GET	/singers/{id}	/	Récupération des informations d'un chanteur ou groupe en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence de chanteur et/ou groupe ayant pour identifiant <i>id</i> .
GET	/singers/search/?fname={fname}&lname={lname}	/	Récupération des informations d'un chanteur ou groupe en fonction de son <i>nom</i> et son <i>prénom</i> .	200 en cas de succès de la requête. 204 en cas d'absence de chanteurs et/ou groupe ayant pour nom <i>lname</i> et pour prénom <i>fname</i> .

Présentation des services liés aux entreprises

Présentation du service de gestion des Label Records

Table 17. Service de gestion des labels records

Méthode	URL	Body	Description	Code de retour
GET	/label-records/	/	Récupération de la liste de tout les labels records enregistrés sur la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence de label records présent dans la médiathèque.
GET	/label-records/{id}	/	Récupération des informations d'un label records en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence de label records ayant pour identifiant <i>id</i> .
GET	/label-records/{name}	/	Récupération des informations d'un label records en fonction de son <i>nom</i> .	200 en cas de succès de la requête. 204 en cas d'absence de label records ayant pour nom <i>name</i> .

Présentation du service de gestion des développeurs

Table 18. Service de gestion des développeurs

Méthode	URL	Body	Description	Code de retour
GET	/developers/	/	Récupération de la liste de tout les développeurs enregistrés sur la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence de développeur présent dans la médiathèque.
GET	/developers/{id}	/	Récupération des informations d'un développeur en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence de développeur ayant pour identifiant <i>id</i> .
GET	/developers/{name}	/	Récupération des informations d'un développeur en fonction de son <i>nom</i> .	200 en cas de succès de la requête. 204 en cas d'absence de développeur ayant pour nom <i>name</i> .

Présentation du service de gestion des éditeurs

Table 19. Service de gestion des éditeurs

Méthode	URL	Body	Description	Code de retour
GET	/publishers/	/	Récupération de la liste de tout les éditeurs enregistrés sur la médiathèque.	200 en cas de succès de la requête. 204 en cas d'absence d'éditeurs présent dans la médiathèque.
GET	/publishers/{id}	/	Récupération des informations d'un éditeur en fonction de son <i>id</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'éditeur ayant pour identifiant <i>id</i> .
GET	/publishers/{name}	/	Récupération des informations d'un éditeur en fonction de son <i>nom</i> .	200 en cas de succès de la requête. 204 en cas d'absence d'éditeurs ayant pour nom <i>name</i> .

Présentation des Technologies du projet

Technologies liées au Backend

Le framework Spring-Boot

Le framework Spring-Boot est un framework développé en **Java** et ayant pour but de simplifier la réalisation d'application web en Java.

Etant une extension du célèbre **Spring Framework**, celui-ci ajoute un ensemble d'annotations facilitant la mise en place et la configuration parfois complexe qu'imposé Spring Framework. En

sus de ceci, Spring-Boot possède un ensemble de plugin supplémentaire ajoutant de nouvelles fonctionnalités au *core* initial du projet. On retrouvera par exemple dans le projet un module de gestion de la sécurité, ou encore de la manipulation de données via la présence de JPA et Hibernate. Vous trouverez plus d'information relative à Spring-Boot sur le site officiel : <https://projects.spring.io/spring-boot/>

La librairie nv-i18n

La librairie NV-I18N est une librairie facilitant la mise en place du respect de la norme **ISO 639-1**, qui permet une identification des pays par un couple de deux lettres. Par exemple, la France est représentée par le couple de lettres *fr*, tandis que les Etats-Unis sont représentés par les lettres *us*.

Cette librairie est utilisée dans l'optique de définir les langues et sous-titres disponibles dans les films ou encore les jeux vidéos gérés par la Médiathèque.

Cette librairie est disponible via le lien suivant : <https://github.com/TakahikoKawasaki/nv-i18n>

Technologies liées au Frontend

Le framework ReactJS

ReactJS est un framework développé en **JavaScript** permettant la mise en place d'une interface utilisateur. Couplé à de nombreuses extensions, ReactJS permet la réalisation d'application web interactive utilisable facilement sur téléphone, tablette ou pc. L'un des nombreux avantages d'utiliser ReactJS est qu'il est l'un des frameworks Front les plus utilisés à l'heure actuelle dans le monde professionnel, et représente donc le framework le plus à même de me servir dans mon futur professionnel.+ Vous trouverez davantage d'information relative à ReactJS ici : <https://reactjs.org/>

Technologies liées aux Tests

Le framework JUnit5

JUnit est un framework de **test unitaire** pour le langage Java. Il est utilisé afin de tester de manière unitaire l'ensemble des services et classes développés dans le cadre du projet, afin de valider leur bon fonctionnement avant la mise en production de ces éléments. Vous trouverez plus d'information sur ce framework à l'adresse suivante : <http://junit.org/junit5/>

Le framework AssertJ

Le framework **AssertJ** est un ensemble de classes et de méthodes facilitant le test de classe en Java. Ce framework, couplé à JUnit permet une rédaction plus rapide et plus concise des tests unitaires.

Cette librairie est disponible via le lien suivant : <http://joel-costigliola.github.io/assertj/>

La librairie Given-When-Then

La librairie **Given-When-Then** est une représentation basique du principe du *Given-When-Then* que l'on retrouve dans la rédaction de test unitaire. Cette librairie est ici utilisée afin de faciliter l'écriture et la compréhension des tests unitaires par une personne ayant une connaissance moins étendue du Java.

Technologies supplémentaires

Le logiciel de versionning Git

Git est un logiciel de versionning de code source. Il est utilisé dans le cadre du projet afin de pouvoir gérer plus facilement l'ensemble du code. De plus, via l'utilisation du **Git-Flow**, Git permet une gestion plus aisée du développement et de la correction des différentes *features* via un ensemble de *branche* dédié à une feature donnée.

Le langage de présentation de document AsciiDoctor

L'**AsciiDoctor** a été utilisé pour la rédaction du rapport du projet *Medialibs*. En effet, un rapport final est demandé afin de présenter le projet. Ainsi, plutôt que d'utiliser les outils classiques tels qu'un *traitement de texte* ou l'utilisation du langage *LaTeX*, j'ai pu découvrir l'AsciiDoctor, qui est une excellente alternative au LaTeX car plus simple à écrire, et davantage compréhensible pour un "non-information".

Auteurs

- Nicolas GILLE <nic.gille@gmail.com>