

STEAMULO

Lyon



Livret Bootcamp

1. Prérequis

1.1. Connexion Google

- Demandez vos identifiants pour votre compte Google Steamulo et configurer votre compte sur Ubuntu dans Paramètres > Comptes en ligne > Google
- Changez les identifiants de connexion WiFi avec ceux trouvés dans le mail du Wojo sur Gmail

1.2. Liens Utiles

- Gitlab - gitlab.steamulo-lyon.com
- Reporting - reporting-lyon.steamulo.com
- Congés - conges-lyon.steamulo.com
- MyThune - mythune.steamulo-lyon.com

2. Installation du poste

2.1. Changement du mot de passe

Un mot de passe a été configuré par défaut sur votre session, n'hésitez pas à le changer en effectuant la commande suivante : `passwd`

2.2. Structure du File System

En dehors des dossiers par défaut d'Ubuntu, il est proposé d'avoir l'architecture suivante pour les dossiers :

```
~ (/home/[session_name])
|
+--- /tools les logiciels installés par download de binaires
+--- /workspaces l'ensemble des répertoires projet
```

2.3. Script d'installation

À la racine de votre home, vous trouverez un dossier intitulé `setup_ubuntu` avec plusieurs scripts dedans.

Exécutez le script `install.sh` qui installera tous les packages et applications nécessaires au développement. Tout est à installer, suivez-bien les étapes, vous aurez à interagir quelques fois (et probablement à entrer votre mot de passe plusieurs fois).

Redémarrez votre PC et vérifiez que tout fonctionne en exécutant ces commandes :

- `nvm install lts/gallium`
- `docker run hello-world`
- `docker compose`
- `code` (Activez Material icon theme comme thème par défaut)
 - PS: Si vous n'aimez pas l'extension ErrorLens, désactivez-la

Demandez votre fichier de configuration wireguard, téléchargez-le puis exécutez le script *wireguard_install.sh*.
Testez en allant sur [GitLab](#).

Vous pouvez télécharger les IDE de JetBrains en initialisant la toolbox via le script *jetbrains-toolbox.sh*.

Normalement c'est tout bon, envoyez votre clé ssh publique à votre parrain pour la renseigner dans l'annuaire des clefs de STEAMULO LYON.

3. Le Bootcamp

3.1. Introduction

3.1.1. Concept général

Le Bootcamp est un parcours initiatique à travers lequel doit passer chaque nouveau développeur non expérimenté arrivant chez Steamulo Lyon (stagiaire ou jeune diplômé).

Le but de ce parcours est de :

- vous donner un aperçu des méthodes de travail utilisées ;
- vous faire découvrir la stack applicative : Spring Boot, Hibernate, architecture REST, Keycloak, front en ReactJs.

Il s'agit en quelque sorte d'un mini plan de formation accéléré à l'environnement de travail des projets Steamulo Lyon.

3.1.2. Durée

Le Bootcamp doit normalement durer deux semaines. Mais c'est une donnée indicative, il faut y consacrer tout le temps nécessaire à l'assimilation des différents concepts abordés.

3.1.3. Conseils et méthodologies

Afin de terminer votre Bootcamp, il vous est proposé de réaliser le projet suivant. Celui-ci permettra de mettre en pratique les différentes techniques, outils et frameworks utilisés à Steamulo Lyon.

Ne vous laissez pas déborder par la multitude des actions à entreprendre (dont certaines vous sont inconnues). L'objectif est d'intégrer pas à pas le maximum de connaissances dans le temps imparti.

Prenez juste le temps de décomposer le projet en tâches élémentaires afin de traiter chaque problème indépendamment des autres, et ayez recours au maximum de ressources disponibles pour vous débloquer en cas de problème.

3.2. Le Sujet

Dead line : environ 2 semaines (Back + front)

Sujet : la ville de Trou-perdu-en-Isère désire réaliser une version web de sa bibliothèque municipale. Lors de la consultation du site, 3 parties doivent être accessibles :

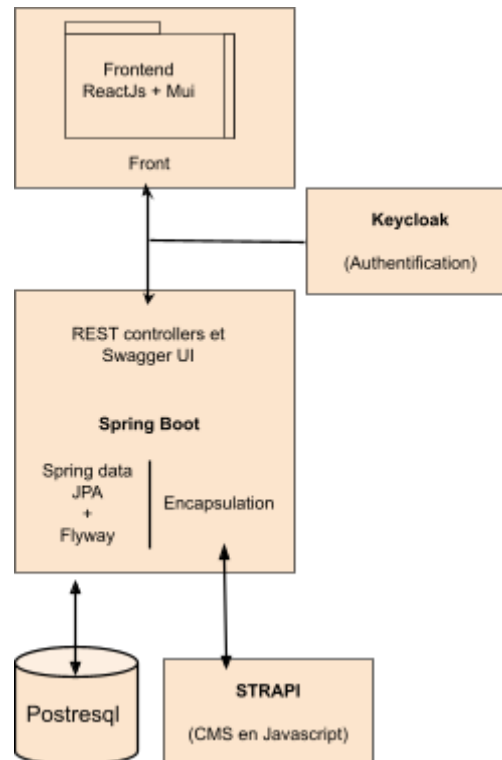
- une partie institutionnelle (contenus éditoriaux) ;
- une partie permettant la consultation des Livres ;
- une partie « mon compte » affichant les informations de l'utilisateur identifié en front-office.

Le site ne sera accessible que pour les utilisateurs identifiés et autorisés (en front-office).

Plan du site (menu du front office) :

- Présentation (article quelconque provenant du CMS)
- Bibliothèque (liste des sous menus représentant les thèmes des livres)
 - littérature (liste des livres (Articles) classés dans le dossier Littérature)
 - bandes dessinées
 - utilitaires
 - livres pour enfants
- Mon compte (affichage des infos de l'utilisateur)

Stack Applicative imposée :



On verra par la suite qu'il y a deux entités principales à gérer : des utilisateurs et des livres. Les livres seront mis à disposition via le CMS (titre, résumé, auteur, couverture, édition, date d'édition), les utilisateurs via le serveur d'authentification Keycloak et toutes les autres données persistantes (les emprunts par exemple) seront stockées dans la base.

3.3. L'échauffement - Préparation de la matière grise

Avant de commencer le sujet, des exercices préliminaires sont proposés :

3.3.1. Exercice 1 : Revue de code

Dead Line : 30 minutes

Objectif :

Analyser et comprendre la classe "[Livres.java](#)". Relever tout ce qui semble étrange, incorrect et proposer une correction (indiquer les modifications de façon succincte en commentaire).

3.3.2. Exercice 2 : La norme tu respecteras

Dead Line : 1 heure

Objectif :

Sur la base des corrections relevées lors du précédent exercice, proposer un document de norme de codage qui servira de référence dans la suite des développements du Bootcamp.

3.3.3. Exercice 3 : Tes connaissances de base on évaluera

Dead Line : 1 à 2 heures

Objectif : Sur cette phase, ton parrain doit te présenter ou te proposer des exercices pour valider que tu connais un certain nombre de concepts de base de notre métier :

- le client/serveur,
- l'AjAx,
- le CSS, le responsive design,
- le SQL, et le NoSQL et la notion de persistance,
- la pile J2EE
- les différents OS mobiles
- les webservices (les formats de flux, la doc Open API)
- Maven & NPM (gestion de dépendances en général)

3.3.4. Exercice 4 : Tes connaissances des configurations de projet Front

Dead Line : 1 à 2 heures

Objectif : Tu devras expliquer à ton parrain les différentes lignes de configuration ou concept listés ci-dessous

- ESLint
- Prettier / EditorConfig
- Husky

Un peu plus général :

- Utilité d'un linter de fichiers en pré commit
- Utilité d'un linter de message de commit

Trouvez la définition de ces 2 lignes de configuration dans le fichier .npmrc (fichier de configuration pour npm), et expliquez ce qu'elles empêchent dans un projet :

- save-exact=true
- engine-strict=true

3.4. Le parcours du combattant

3.4.1. Cahier des charges

Back-office

Page / Fonctionnalité	Commentaires
Gestion de l'authentification	<ul style="list-style-type: none"> La page Back Office n'est pas authentifiée dans une 1ère version. Une authentification HTTP BASIC sera mise en place par l'hébergeur de Trou-perdu-en-Isère
Gestion des inscriptions	<ul style="list-style-type: none"> Permet de visualiser la liste des utilisateurs qui se sont inscrits. Permet à un utilisateur du back-office d'autoriser le prêt de livres à un utilisateur inscrit. Tant que l'utilisateur n'a pas été autorisé par un utilisateur Back-Office, il ne peut pas emprunter de livre. Un moteur de recherche permettra de rechercher les utilisateurs par leur nom, leur prénom, leur email et / ou par leur activation.

Front-office

Page / Fonctionnalité	Commentaires
Page d'accueil	<ul style="list-style-type: none"> La page d'accueil de la bibliothèque de Trou-perdu-sur-Isère accueille les utilisateurs sur son site avec un contenu simple et épuré permettant de comprendre simplement son fonctionnement
Liste des livres	<ul style="list-style-type: none"> Permet de visualiser la liste des livres organisés par catégories. Doit être simple, efficace et performant Permet à l'utilisateur de prendre connaissance du livre, et d'en demander le prêt
Gestion de la connexion	<ul style="list-style-type: none"> Un utilisateur doit se connecter pour demander l'emprunt d'un livre Il doit pouvoir se connecter, ou se créer son compte qui sera par défaut validé mais interdit d'emprunt Il doit pouvoir gérer son oubli de mot de passe
Emprunt d'un livre	<ul style="list-style-type: none"> Chaque livre doit pouvoir être emprunté s'il n'a pas été emprunté par ailleurs
Mon compte	<ul style="list-style-type: none"> L'utilisateur voit ses infos L'utilisateur voit ses emprunts et peut rendre un livre

3.4.2. Exercice 1 : La charge tu supporteras

À partir du cahier des charges, vous devez réaliser un plan de charge. Il s'agit d'un découpage du travail en une liste de tâches auxquelles devront être affectées des durées en fraction de jour de travail (0,25 j, 0,5 j, 0,75 j, 1 j, etc...).

Ton parrain doit te fournir le [template de plan de charge vide](#) que tu pourras remplir.

Organisez-vous afin de déterminer l'ensemble et l'ordre des tâches à réaliser afin de mener votre projet à bien.

On ne commence pas le développement tant que le plan de charge ne tient pas la route. A valider avec l'parrain donc.

3.4.3. Exercice 2 : Le Bootcamp tu développeras

C'est maintenant qu'il va falloir apprendre à nager, pendant toute cette phase, tu vas réaliser la partie back-end présentée plus haut.

Les Checkpoints

Pour s'assurer que tout se passe bien, voici les deux points de passages pendant toute la durée de l'exercice :

- avant de commencer le développement de chaque page / développement : il faut présenter à ton parrain la solution technique que tu vas mettre en place, il te guidera sur le bon chemin ou te fera creuser des pistes techniques pour t'éviter de te noyer ;
- selon ton avancement, des revues de code régulières seront effectuées par ton parrain pour te faire des retours constructifs sur ton code.

De l'aide !

Quelques repères sont là pour t'aider :

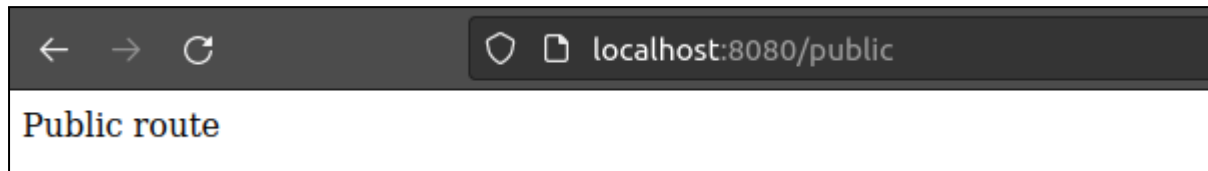
- ton parrain : il a l'air occupé, il l'est sûrement mais va le voir, pose lui des questions, il est là pour t'expliquer les grands concepts du projet, pour t'aiguiller quand tu pars dans la mauvaise direction, bref tu as compris ;
- l'aide et les ressources présentés ci-après.

3.5. Aide

3.5.1. Commencement - Initialisation du projet

Objectif

Avoir une application back-end qui tourne, avec une page de test "Public route" :



Indices

La première chose à faire est de récupérer la structure des projets de départ.

Les projets se trouvent aux adresses suivantes :

- Le projet back-end : [Starter Back](#)
- Le projet strapi : [Starter Strapi](#)
- Le projet front-end : [Starter Front](#)

Aide Technique

- [Git Cheat Sheet](#)
 - Chacune de vos fonctionnalités doit être développée sur une branche distincte de votre repository git
 - Lorsqu'une fonctionnalité est prête, vous devez effectuer une "merge-request" de votre branche vers le master à votre parrain (possible directement dans l'UI gitlab). Celui-ci pourra alors faire une revue de code
 - Le master représente donc la version la plus à jour de votre projet, avec les dernières évolutions validées

3.5.2. Préparation du terrain - Création des données STRAPI

Objectif

Créer une liste de livres dans STRAPI, et la montrer à ton parrain :

```
{
  "data": {
    "id": 1,
    "attributes": {
      "title": "Watchmen",
      "summary": "New York, 1985. Alors qu'une guerre nucléaire menace d'éclater...",
      "author": "Alan Moore",
      "edition": "Deluxe Edition",
      "editionDate": "1986-11-01",
      "createdAt": "2022-07-07T10:10:50.735Z",
      "updatedAt": "2022-07-25T07:30:12.445Z",
      "publishedAt": "2022-07-07T10:11:21.419Z",
      "category": "BD"
    }
  },
  "meta": {}
}
```

Indices

Strapi est un CMS *Headless* (solution clef en main, qui expose des API) qui propose une interface web de création de contenu (pour nous ce sera des livres), et qui gère toute la partie stockage en base de données.

La création de livre passe par la configuration de templates afin de définir les différents attributs d'un livre.

Lorsque tu as fini de créer tes entités, une API est à ta disposition pour récupérer ce que tu viens de créer.

Aide Technique

Documentation de Strapi : <https://docs.strapi.io/developer-docs/latest/getting-started/introduction.html>

3.5.3. Consommer STRAPI

Objectif

Afficher la liste des livres dans un webservice de l'application Spring-boot :

GET /books

```
{
  "books": [
    {
      "id": "1",
      "title": "Le rouge et le noir",
      "image": "http://....",
      ...
    },
    {
      "id": "2",
      "title": "Le rouge et le noir",
      "image": "http://....",
      ...
    }
  ]
}
```

Indices

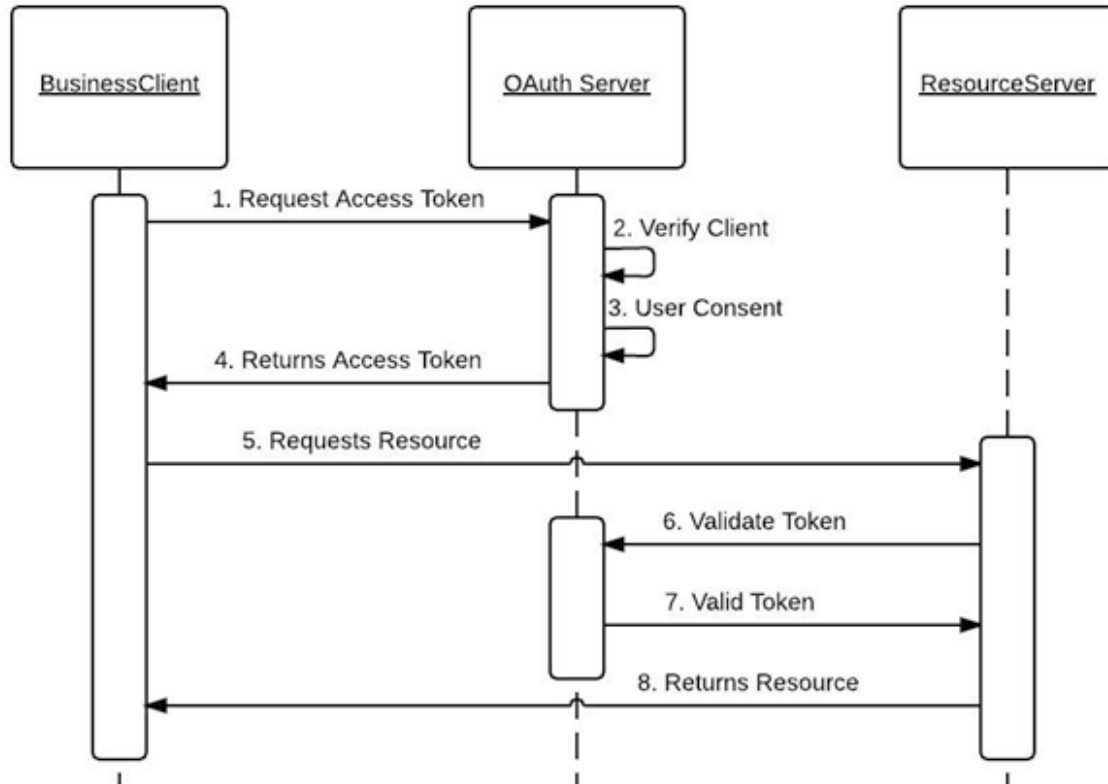
Pour se faire, procéder par étape :

1. Afficher dans la console Spring-boot la liste des livres de STRAPI
2. Exposer un Webservice qui rend le résultat attendu

3.5.4. Authentification

Objectif

Mettre en place l'authentification :



Indices

Si vous avez bien suivi le README.md du starter back-end, il ne reste plus qu'à créer des rôles et protéger vos routes.

Aide Technique

Documentation Keycloak : <https://www.keycloak.org/>

3.5.5. Autres fonctionnalités

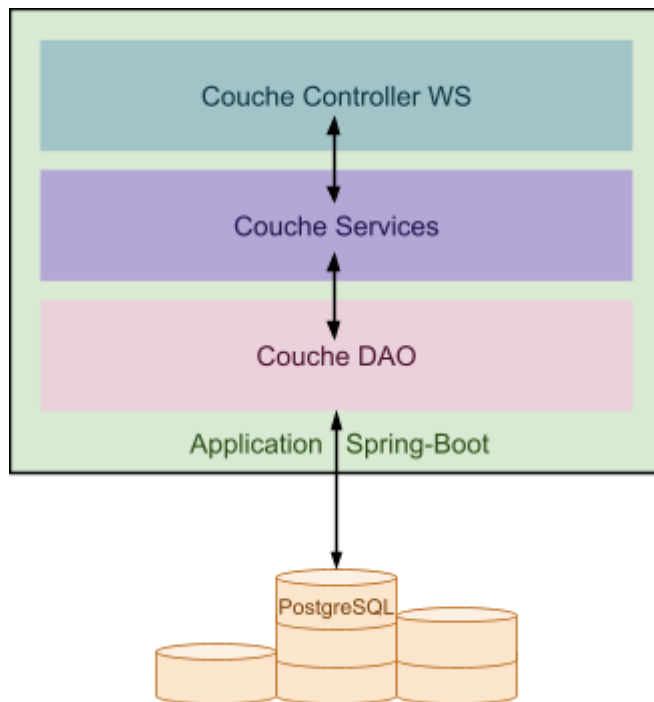
Objectif

Implémenter les autres fonctionnalités du cahier des charges (emprunts, gestion des droits...).

Indices

Il faut y aller par étapes, et sous forme de couches :

- Couche DAO : Création des classes, configuration des entités et du "repository"
- BDD : Création de la structure de Base de données avec Hibernate
- Couche Service : Récupération, création, modification, suppression des entités.
- Couche Controller : Exposition d'un webservice respectant le standard REST



3.5.6. Faire le ménage !

Objectif

Après avoir passé beaucoup de temps à comprendre les diverses couches de l'application, et faire fonctionner le tout ensemble, il est temps de prendre un peu de recul sur ce qui est fait. Et donc prendre le temps de faire le ménage dans le code, et commencer l'interface !

Indices

Nettoyer le code :

- supprimer le code mort
- ajouter des commentaires
- réorganiser les dossiers suivant les couches de l'application

3.5.7. Front-end

Objectif

Suivre le cahier des charges pour les pages à faire

S'interfacer avec Keycloak pour la connexion

Comprendre comment le projet est setup pour être capable de répondre à des questions techniques

Indices

1-10

Aide Technique

Vous trouverez les documentations dans le README du starter front

Bibles du web :

- [W3Schools](#)
- [Mozilla Dev Doc](#)
- [Doc React](#)

Le Saviez-vous ?

Voici rapidement quelques explications des choix techniques :

- **Vite** remplace create-react-app / react-scripts qui est vieux et lent pour lancer le frontend et pour le build principalement, il bénéficie aussi d'une technologie appelée HMR (Hot Module Reloading) qui permet de rendre le dev plus agréable car au lieu de recharger toute l'app à chaque sauvegarde, on recharge uniquement les modules modifiés.
- **pnpm** remplace npm ou même yarn, il ne stocke pas les node_modules dans le répertoire du projet, à la place il les met dans un store situé dans votre \$HOME et ne fait que créer des liens symboliques. Le gros avantage est que vous gagnez beaucoup de places sur votre PC et que si vous supprimez votre node_modules, vous n'aurez pas à tout retélécharger vu qu'ils sont déjà dans votre store pnpm. Les logs sont beaucoup mieux formatés, jolis et compréhensibles. On peut mentionner aussi le fait qu'il peut installer les "peer dependencies" manquantes automatiquement, ce qui d'habitude est plutôt embêtant à gérer.
- **Corepack** est un manager de package manager, pas mal hein. Encore expérimental, il va regarder si l'option "packageManager" est spécifié dans votre package.json et tout gérer pour vous. Plus besoin de vérifier si vous avez la bonne version sur votre PC ou installer à chaque fois le bon package manager dans chaque projet, activez corepack et c'est réglé. On s'assure aussi que tous les membres du projet utilisent le même package manager pour éviter d'éventuels problèmes.

/!\ Étant donné que les versions de node sont gérées par nvm et que corepack n'est pas activé par défaut, il faudra l'activer pour chaque version de node /!\

PS : Le starter front sur lequel vous allez travailler est un des plus avancés en termes de configuration et technologies. Les projets sur lesquels vous allez travailler après seront sûrement plus en retard, il faudra s'adapter

3.6 Divers

3.6.1. Liens utiles

Référence spring boot : <http://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

3.6.2. Astuces

- Il faut utiliser les annotations Spring, mais avant cela il faut bien comprendre ce que l'annotation fait.
- Spring boot comprend des devtools qui peuvent faire gagner un temps considérable grâce au LiveReload et à l'automatic restart. (<https://code.visualstudio.com/docs/java/java-spring-boot>)