



PROJET 4

Contexte

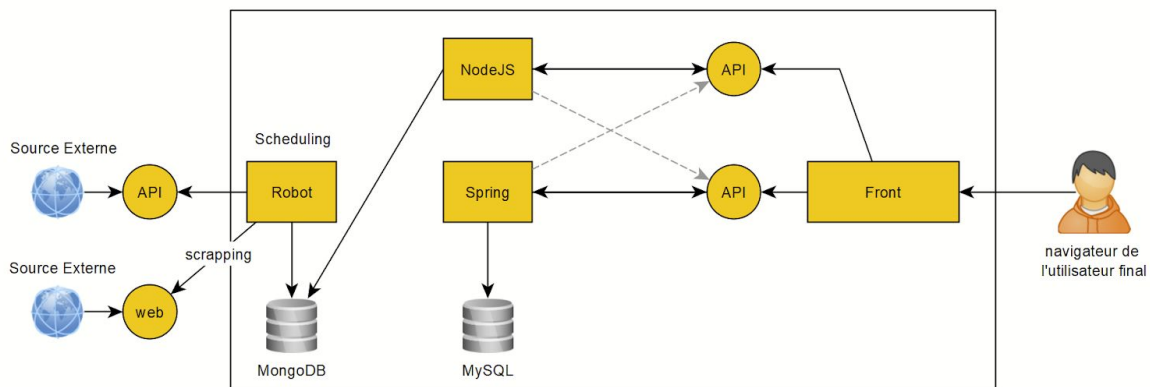
Vous devrez réaliser une solution complète sur le sujet / thème de votre choix.

Ce projet se déroulera individuellement. En effet, nous tenons à ce que vous soyez le plus libre possible dans vos choix, et que vous tiriez parti de cette autonomie.

De plus, vous devrez pour ce projet, monter en compétence par vous même sur certains sujets. Les exigences sur ce projet est d'aboutir à un projet complet, avec des tests, multi technos, hébergé en production, avec des déploiements automatisés.

L'architecture

Le principe est de développer une application web complète, avec différentes briques et micro services séparés qui communiquent entre eux suivant les besoins. Ces différents éléments peuvent être schématisés comme ci dessous.



Front

Le Front est la partie visible de votre application.

Cette partie devra être développée à l'aide d'un framework Javascript moderne et de haut niveau. Vous êtes libre de choisir le framework que vous voulez, par exemple Angular, React ou Vue. Cependant, vous allez voir en cours le framework Angular, et vous devriez disposer d'aide sur cette techno durant le projet, il est donc plutôt recommandé de le préférer à React ou Vue. Vous **devez** utiliser un framework, les implémentations d'une solution Ajax "à la main", avec ou sans jQuery sont donc interdites.

Votre partie front devra être **utilisable et cohérente** et un minimum d'esthétique serait plutôt une bonne chose. Ne négligez pas que l'aspect graphique de votre interface est la partie visible de votre projet. Néanmoins rassurez-vous, vous n'avez aucune obligation de créer vous même la totalité de cet aspect graphique, et il est même plutôt recommandé de trouver le template adéquat pour votre application.

Cette partie front étant uniquement côté client, les données qui seront affichées viendront du côté serveur : votre partie front devra donc se connecter au backend via une API rest (voir détails dans les parties backend)

Backend Utilisateur

Votre application devra comprendre un micro service développé en Java / Spring qui contiendra toute la logique utilisateur de votre application, comme la connexion à votre application, les préférences utilisateurs, etc etc. Pour rappel, ce micro service devra exhiber une **API Rest**.

Cette partie stockera les données à persister dans une base de données **MySQL**. Les mots de passe stockés en base **doivent impérativement être cryptés**.

La mise en place d'un micro service d'authentification central, qui permettra de partager un token avec les autres microservices, est une façon très professionnelle de gérer l'authentification. Cependant, la mise en place peut être très fastidieuse. Vous pouvez adopter cette stratégie si vous le voulez, l'effort sera apprécié, mais ce n'est pas une obligation. Attention à ne pas perdre de temps la dessus :)

Backend Data

Votre application devra comprendre un micro service développé en NodeJS qui servira à exploiter les données présentes une base de données **MongoDB**. De même que le micro service utilisateur, ce service devra être requêtable par le biais d'une **API Rest**.

A noter : si pour le besoin de votre application, vos services doivent communiquer entre eux, ils devront le faire par le biais de leurs API. Par conséquent, la partie NodeJS **ne doit jamais se connecter directement à MySQL**, et de son côté, la partie Java **ne doit jamais se connecter directement à MongoDB**.

Robot

Cette partie “robot” est peut être la partie qui vous est le moins familière.

Pour l'instant, vous êtes habitués au traitement “synchrone”, avec résultat immédiat : le fait de cliquer sur un bouton appelle tout de suite la fonction qui s’y rapporte, ou également, le fait de passer une requête HTTP à une adresse donnée déclenche tout de suite la génération de la réponse.

A l'inverse, votre partie “robot” devra se déclencher non pas manuellement ou de manière synchrone, mais automatiquement et régulièrement, à une fréquence définie qui aura du sens pour votre application, par exemple toutes les 5 minutes, tous les soirs à minuit, une fois par mois, etc. En d'autres termes, le lancement de votre partie robot devra être “**schedulée**”.

> On parle en terme UNIX technique de tâche en “cron” ou encore de cronjob.

Dans notre cas, le rôle de ce robot sera de se connecter à une **API externe** afin de récupérer des données. Si vous ne trouvez pas d'APIs qui vous conviennent, vous avez également la possibilité de récupérer les données externes sur d'autres sites internet, en extrayant les informations directement des pages. Cette opération s'appelle le “scrapping” ou encore le “crawling”

Bien évidemment la source de vos données est très importante. Il faut que vous trouviez une API qui est un intérêt à être requêtée régulièrement, par exemple une API météo, et non une API qui donne des informations à la demande.

Sauf exception à justifier, **seul le robot doit requêter cette API externe**

La technologie utilisée pour cette partie robot est laissée libre. Comme vous aborderez NodeJS et la connexion à MongoDB avec NodeJS en cours, il sera plus simple de réaliser cette partie en NodeJS. Cependant si vous le souhaitez, si votre application le nécessite, vous pouvez réaliser cette partie dans le langage et avec les librairies de votre choix, par exemple en Go, en Python, etc. Même remarque que précédemment : Attention à ne pas y passer trop de temps !

Infrastructure

Hébergement

Votre projet devra exister en plusieurs versions. Bien sûr, il devra fonctionner sur votre poste, mais il devra aussi être hébergé en production. Cela signifie que n'importe qui devra pouvoir s'y connecter via une URL que vous communiquerez. Le choix des solutions d'hébergement est laissé libre.

La livraison du code

Vous devrez mettre en place un / des pipelines de livraisons de votre code basé sur Git, depuis votre environnement de développement local jusqu'à la production.

Complément d'information

Vous devrez être dans la possibilité de faire une modification et de la pousser jusqu'en production avec votre pipeline d'automatisation de livraison du code

Vous devrez pouvoir faire la démonstration du bon fonctionnement de votre site.

Vous devrez respecter les bonnes pratiques, notamment les 12 Factors

Encadrement

Contexte

Le projet 4 est une occasion unique pour vous de découvrir de nombreuses technologies et de pouvoir par la suite faire la démonstration de vos savoirs, notamment pendant votre recherche d'emploi. Enfin, il est une occasion précieuse de gérer un projet de bout en bout, et ce, sur tous les volets, développement, hébergement, front/back, gestion de votre projet et de votre temps.

Le projet 4 est également le dernier de votre cursus à Isika. Il vient valider pour vous la possibilité de présenter le diplôme final.

Soutenance

Votre projet va faire l'objet de plusieurs soutenances et notations, et ce afin de vous permettre de valoriser chaque aspect de votre travail.

1. Soutenance Intermédiaire : nous ferons un point, à mi-parcours, sur la mise en place de votre projet. Il s'agira pour vous de présenter votre sujet, une première implémentation "naïve" de chacune des briques, et surtout, de prendre en main la production, le déploiement, et de mettre en place vos propres outils d'automatisation des tests et des déploiements. Cette soutenance arrivera à mi-parcours
2. Soutenance Technique : nous ferons une évaluation individuelle, durant 1h, de votre projet. Préparez votre présentation, et idéalement, préparez un petit support de quelques slides qui viendront vous aider à faire un tour exhaustif de votre travail.
3. Soutenance Finale : il s'agit des soutenances de la promotion précédente auxquelles vous avez assisté. L'occasion vous est donnée de présenter votre projet d'un point de vue fonctionnel, durant 10 à 15 minutes, en présence de :
 - a. Vos camarades
 - b. L'équipe pédagogique Isika
 - c. La promotion qui vous suivra
 - d. Des externes, entrepreneurs, recruteurs.

Rendu

Attention, la dernière semaine n'est pas la dernière semaine de code ! Nous ramasserons votre travail le dimanche précédent la semaine des soutenances, et ce, afin que tout le monde soit sur un pied d'égalité en terme de temps de travail. La bonne gestion de votre temps fait partie de la notation !

La dernière semaine de projet sera répartie comme suit :

- J - 5 à 23h59 : Ramassage de votre travail
- J - 4 : Vous préparez votre soutenance technique

- J - 3 / J - 2 : Soutenance technique d'1h
- J - 1 : Vous préparez votre soutenance finale
- Le Jour J : matin et après midi : Soutenance finale 10min + 10min question

Suivi

Durant toute la durée du projet, vous serez encadrés dans vos choix dans les différentes étapes de votre projet.