

Haute école d’ingénierie et d’architecture

Fribourg

|  |
| --- |
| **Système d’information**  TP01 – Framework Backend  ISC-IL-3 |

17 novembre 2022

Auteurs :

* Naël Telfser
* Quentin Maradan
* Erwan Sturzenegger

# Introduction

Ce TP a pour but de nous faire découvrir, implémenter et comparer un backend Spring MVC et un backend ASP.Net Core MVC. L'application a pour but de gérer des ligues, des équipes, des matchs et des joueurs.

Nous avons choisi d'implémenter le Frontend en Angular. Nous avons aussi choisi d'utiliser une base de données MySQL pour stocker les données.

Le déploiement de l'application se fera sur une machine virtuelle Ubuntu 22.04 LTS au moyen de conteneurs Docker.

# Comparaison des deux applications

## Charge CPU

## Nous avons choisi ce critère car il nous semble important de savoir quelle est la charge CPU pour une application Spring et respectivement pour une application DotNet. Ce n'est pas très significatif pour une seule application mais cela peut devenir problématique si nous avons plusieurs applications qui tournent en même temps sur une machine. Nous avons donc choisi de comparer les deux technologies sur ce critère.

## Occupation de la mémoire

Ce critère nous paraît au moins aussi important que le précédent. Lorsque de nombreuses applications tournent en même temps sur une machine, il est important de savoir si la mémoire est suffisante pour toutes ces applications. Nous avons donc choisi de comparer les deux technologies sur ce critère.

## Facilité d’extension à des nouvelles fonctionnalités

Nous avons choisi ce critère car une application, le plus souvent, évolue au fil du temps. Il est donc important de pouvoir facilement ajouter de nouvelles fonctionnalités. Nous avons donc choisi de comparer les deux technologies sur ce critère.

# Synthèse

Pour la charge CPU et la RAM, nous avons fait un test avec 100 utilisateurs durant 20 secondes avec Artillery.

## Charge CPU

Graphique de la charge CPU pour DotNet



## Occupation de la mémoire

Graphique de la charge mémoire pour DotNet



## Facilité d’extension à des nouvelles fonctionnalités

Le tableau ci-dessous compare les deux framework selon différents critères que nous avons jugé pertinent concernant la facilité d’extension à des nouvelles fonctionnalités. Les framework sont notés de 1 à 5, 1 étant le moins bien, pour chaque critère.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | ASP.Net Core MVC | Spring |
| Critères | Poids | Note | Note |
| Facilité de mise en place du projet | 3 | 2 |  |
| Facilité d’ajout d’endpoints | 1 | 5 |  |
| Facilité de modification de la DB | 2 | 3 |  |
| Total |  | 17 |  |

Voici la raison de nos choix pour les critères :

* Facilité de mise en place du projet :
  + Afin de pouvoir ajouter des fonctionnalités sans trop perdre de temps, il est important d’avoir un projet bien structuré dès le début du développement.
* Facilité d’ajout d’endpoints :
  + Lors de l’ajout d’une fonctionnalité, il y a généralement des endpoints en plus.
* Facilité de modification de la base de données :
  + Comme pour les endpoints, la base sera certainement modifiée. Est-il facile de modifier celle-ci ? De plus, est-il ausis facile de garder une trace des modifications faites ?

# Automatisation du déploiement

## ASP.Net MVC

Pour ASP.Net MVC, nous appelons un script sur la machine virtuelle exoscale qui va se connecter au registry de l’école, télécharger les nouvelles images avec la commande « docker pull » et recréer les services du docker-compose avec la commande « docker compose up -d ».

# Conclusion