

Enseignant(s)

**ZELLOU Charaf**

Email(s)

[team@blockchain-xpertise.com](mailto:team@blockchain-xpertise.com)

## 4IBC - Langage GO

### 1 Matières, formations et groupes

Matière liée au projet :

Formations : -

Nombre d'étudiant  
par groupe :

**1**

Règles de constitution des groupes: **Imposé**

Charge de travail  
estimée par étudiant : **6,00 h**

### 2 Sujet(s) du projet

Type de sujet : **Imposé**

#### Énoncé 4A IBC - API Github

Ce projet a pour but de tester les compétences des étudiants de 4ème année à créer une application en utilisant leurs connaissances sur le langage Go.

### 3 Détails du projet

**Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)**

Ce projet a pour but de tester les compétences des étudiants de 4ème année à créer une application en utilisant leurs connaissances sur le langage Go.

### Descriptif détaillé

Le but de ce projet annuel est de créer une application en Golang complète permettant de cloner les repositories depuis Github, selon les critères ci-dessous.

Les étudiants seront amenés à développer les fonctionnalités suivantes :

Créer une application qui requête l'API Github pour récupérer:

la liste de repositories d'un utilisateur,

ou la liste de repositories d'une organisation,

Trier ces repositories par dernière modification.

L'application doit récupérer au minimum TOUS les repositories spécifiés, ou au minimum les 100 derniers repositories par date de modification.

L'application doit écrire un CSV de cette liste, avec l'ensemble des informations récupérées sur l'API.

L'application doit cloner ces repositories en local.

L'application doit exécuter un Git Pull sur la dernière branche modifiée (dernier commit) en local.

L'application doit aussi exécuter un Git Fetch pour récupérer toutes les références de branches en local.

L'application doit créer une archive (ZIP ou 7z) de ces repositories à la fin du traitement en local.

Une fois déployée, la dApp aura comme fonctionnalités de :

Spécifier le pseudo Github d'un utilisateur ou une organisation,

Lister et cloner les repositories publiques de l'utilisateur ou l'organisation,

Si un Token API Github est fourni, l'application doit en supplément cloner les repositories privés de l'utilisateur ou l'organisation,

Rendre disponible le téléchargement de ces repositories via une API.

L'utilisation des notions Golang suivantes est obligatoire :

Webserver HTTP pour le téléchargement de l'archive,

Goroutines & Waitgroups pour optimiser l'exécution du code.

La dApp doit être Dockerisée afin de faciliter son déploiement.

Des volumes persistants pour la BDD sont à prévoir.

### Ouvrages de référence (livres, articles, revues, sites web...)

<https://pkg.go.dev/>

<https://docs.github.com/fr/rest?apiVersion=2022-11-28>

### Outils informatiques à installer

<https://pkg.go.dev/>

## 4 Livrables et étapes de suivi

1

Rendu final

jeudi  
28/09/2023  
17h00

## 5 Soutenance

Durée de présentation  
par groupe :

**0 min**

Audience :

Type de présentation :

Précisions :