

# Projet 4 : Interface Web service traitement batch sur Docker

---

## DockerINT - Revue numéro 1 - 12.03.2021

---

### Encadrant :

Olivier BERGER - olivier.berger@telecom-sudparis.eu

### Étudiants :

Marion SAVES - marion.saves@telecom-sudparis.eu

Nordine MARIE - nordine.marie@telecom-sudparis.eu

### Dépôt du projet :

<https://github.com/nordine-marie/dockerint>

## 1. Description du projet

---

### Présentation

DockerINT a l'ambition de créer une interface web permettant à tout membre du personnel ou étudiant du campus de lancer des programmes diverses, qui s'exécuteront sur un serveur distant du type : - Récupération d'enregistrement de salons BBB pour post-processing vidéo ;

- Extraction de rendus Moodle pour intégration de noms d'étudiants avant assemblable en PDF imprimables ;
- Tests automatisés sur des rendus de projets dans Moodle, pour notation etc.

Depuis notre interface, on pourra avoir accès à l'état de la tâche. Ce projet deviendrait accessible depuis Moodle, ecampus ou autres et pourrait nécessiter une authentification par CAS.

**Attention les profs n'aurait pas à savoir utiliser Docker**, Docker est une technologie qu'on utilise derrière l'interface web où ils ont juste à cliquer sur le programme voulu, importer leur .zip et attendre que la tâche soit terminée pour ensuite télécharger le résultat.

À terme, il faut aussi que des professeurs puissent ajouter des nouveaux programmes et tester leurs fonctionnements. Pour les profs expérimentés nous leur laisserons la possibilité de télécharger le container aussi.

**Pourquoi DockerINT ?** 1. Toutes nos machines sont différentes et la **dockerisation** des programmes lancées à distance règle ce problème ; 2. Ne pas utiliser son propre ordinateur pour effectuer des tâches qui ralentirait ensuite notre **productivité**. 3. **Mutualisation et partage** de programmes utiles au campus à un seul et même endroit.

### Objectifs :

- Créer une interface utilisateur web accessible à tous ;
- Mettre en place le serveur distant s'occupant d'identifier l'utilisateur et d'exécuter/gérer les tâches en réponse à l'interface web.

### Technologies & Outils utilisées :

**a. Technologies** : Docker, Go-Docker, Unix, HTML, CSS, Javascript, AngularJS **b. Outils** : Git, GitHub, Pandoc, Stackedit, Framadate

## 2. Planning

---

En accord avec notre encadrant, nous utilisons une **méthodologie agile** se composant en une succession de sprints.

Sprint	Sous-Taches	Durée prévu
<b>Veille technologique</b>		<b>1 mois</b>
	Formation à Docker	1 semaine
	Recherche sur le web processing	1 semaine
	Formation à Go-Docker	2 semaine
<b>Interface Utilisateur</b>		<b>2 semaines</b>
	Définition de la charte graphique	1 semaine
	Modélisation graphique (Figma)	1 semaine
	Intégration web (HTML, CSS, Javascript)	1 semaine
<b>Serveur Distant</b>		<b>1 mois</b>
	Mise en place de Go-Docker	2 semaines
	Mise en place du back-end	1 semaine
	Connecter l'interface utilisateur au serveur distant	2 semaines
<b>Création de programmes types</b>		<b>1 mois</b>
	Pas encore défini	

### Fonctionnement des réunions avec l'encadrant

**Chaque lundi** nous planifions une réunion avec notre tuteur afin de lui parler de notre avancement et pour lui poser des questions.

La réunion du lundi marquant le début d'un sprint sera particulièrement importante. En effet, durant cette réunion **nous ferons un point sur nos objectifs passés et nous définirons les suivants.**

## 3. Travail effectué

---

Nous nous sommes formé à **Docker** et documenté sur le batch processing au moyen de containers docker. Après recherches, nous avons donc conclut que **Go-Docker** s'apparentait comme l'outil le plus adapté à notre projet et nous permettait de partir d'une base adéquate. À la suite de cela nous avons commencé à concevoir une **interface utilisateur** que nous avons finalement intégré grâce aux technologies web **HTML, CSS, Javascript et AngularJS**. En ce moment même, nous travaillons sur la **mise en place du serveur distant**.