

ÉTAT D'AVANCEMENT DU STAGE PFE

Sujet : Framework de Génération et Automatisation
des Tests d'API Rest

Réalisé par :

Abou el haoul

Houssam Eddine

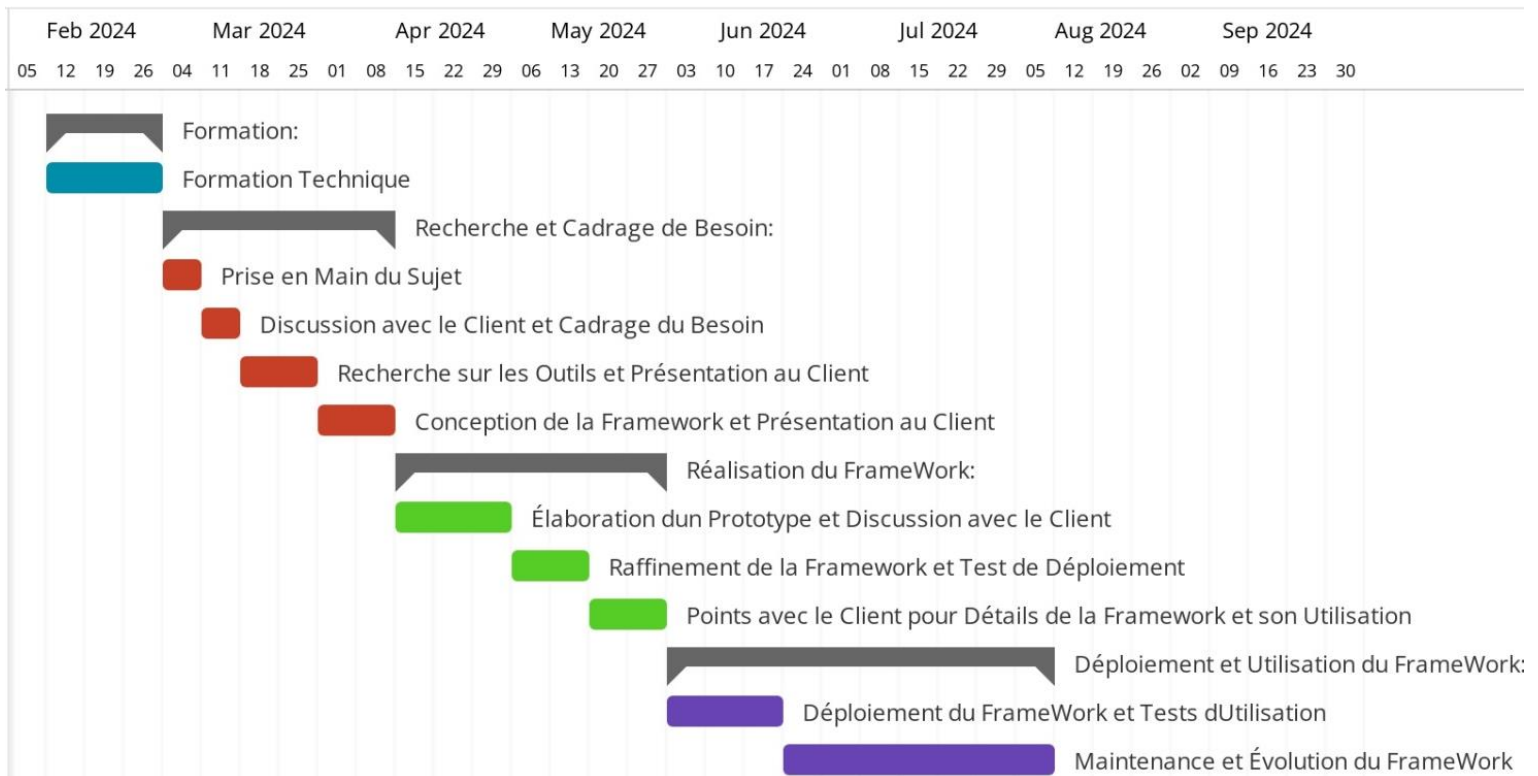
Encadré par :

Pr. Ouatik El Alaoui

Said

Février-Aout 2024

Diagramme de GANTT du projet



Id	Name	Start	Finish	Duration (Cal. Days)
1	Formation:	05/02/2024	01/03/2024	26
1.1	Formation Technique	05/02/2024	01/03/2024	26
2	Recherche et Cadrage de Besoin:	04/03/2024	12/04/2024	40
2.1	Prise en Main du Sujet	04/03/2024	08/03/2024	5
2.2	Discussion avec le Client et Cadrage du Besoin	11/03/2024	15/03/2024	5
2.3	Recherche sur les Outils et Présentation au Client	18/03/2024	29/03/2024	12
2.4	Conception de la Framework et Présentation au Client	01/04/2024	12/04/2024	12
3	Réalisation du Framework:	15/04/2024	31/05/2024	47
3.1	Élaboration d'un Prototype et Discussion avec le Client	15/04/2024	03/05/2024	19
3.2	Raffinement de la Framework et Test de Déploiement	06/05/2024	17/05/2024	12
3.3	Points avec le Client pour Détails de la Framework et son Utilisation	20/05/2024	31/05/2024	12
4	Déploiement et Utilisation du Framework:	03/06/2024	05/08/2024	64
4.1	Déploiement du Framework et Tests d'Utilisation	03/06/2024	21/06/2024	19
4.2	Maintenance et Évolution du Framework	24/06/2024	05/08/2024	43

Recherche et Benchmark des différents outils d'automatisation

Durant le long de mon projet, j'avais une autonomie totale sur quels outils ou langage ou Framework utiliser pour accomplir la tâche d'automatiser les tests des API REST. Du coup j'ai fait plusieurs recherches sur plusieurs outils d'automatisation pour trouver lequel serait le plus adapté à notre problématique. Dont voici une figure de benchmark :

COMPARAISON DES OUTILS D'AUTOMATISATION DE TEST

	<u>Jmeter</u>	<u>Rest Assured</u>	<u>Katalon</u>	<u>Assertible</u>	<u>Postman</u>	<u>Soap UI</u>
Services	<u>Rest</u> + SOAP	<u>Rest</u>	Web + Mobile + <u>Rest</u> + SOAP	<u>Rest</u>	<u>Rest</u> + SOAP	<u>Rest</u> + SOAP
Spécialité	Test de Performance	<u>Rest</u> en JAVA	Polyvalent (mais plus interface graphique)	<u>Rest</u>	<u>Rest</u> + SOAP	<u>Rest</u> + SOAP
Facilité d'utilisation	Avancé	Avancé	Facile	Moyen	Facile	Moyen
Intégration CI/CD	CLI + <u>Maven</u> + plugins Jenkins	<u>Maven</u>	CLI	API d' <u>Integration</u>	CLI	CLI
Tarification	Open Source	Gratuit	Gratuit + Payant	Gratuit + Payant	<u>Gratuit</u> + <u>Payant</u>	Open Source + Payant

Choix des outils et langage du projet

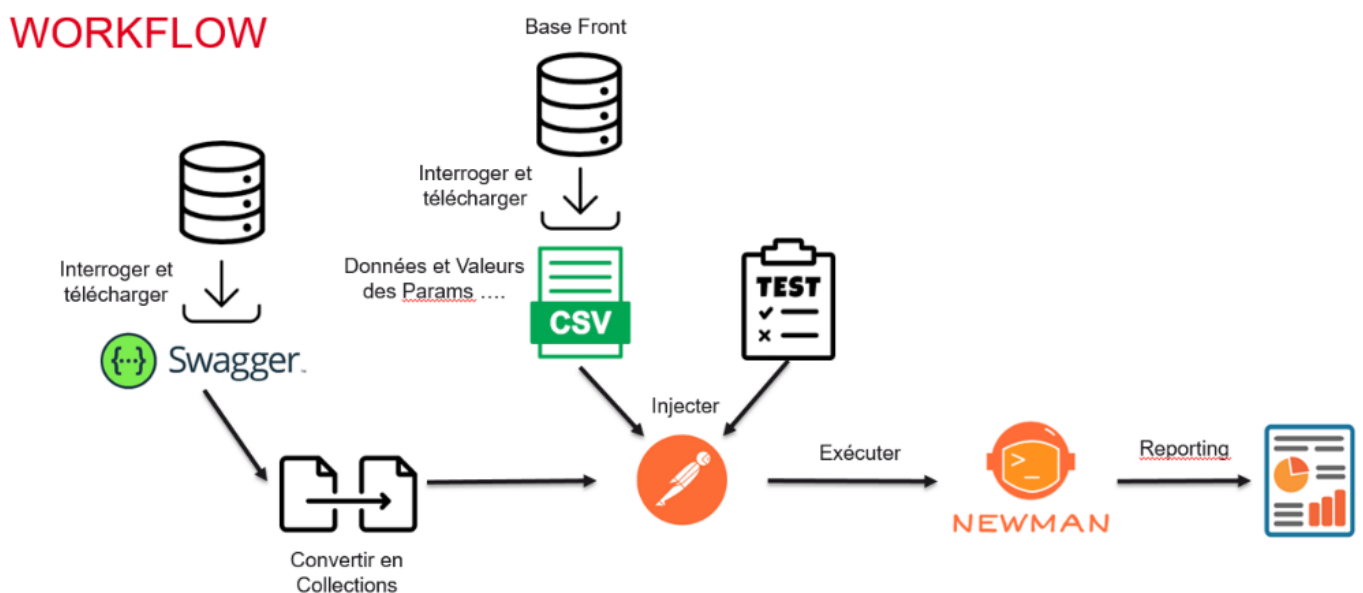
En prenant en considération mes recherches, je suis arrivé à cette conclusions : Il n'y a pas d'outil qui satisfait toutes les demandes de la problématique a 100%, surtout tous ce qui touche à la génération automatique des cas de tests et leur mise à jour automatique.

Donc, je devrais élaborer une solution 'from scratch' vu la nature spécifique de la problématique. En sachant ceci, voici les spécifications techniques de mon Framework :

- La solution sera élaborée en JavaScript, grâce à ses nombreuses librairies et la facilité de manipulation des fichiers et surtout les fichier JSON.
- La solution sera basée sur Postman puisqu'il offre des librairies JavaScript très utiles dans notre cas, en plus les Collections Postman sont des fichiers JSON donc plus facile à manipuler.
- La solution sera déployée dans une Pipeline CI/CD pour faire les exécution automatique.

Workflow de la Framework (mode opératoire)

Voici un diagramme des étapes que mon Framework suit pour automatiser le processus des tests :



Voici les étapes en détails :

- **Étape 1** : Télécharger les fichiers Swagger (documentation des API) et les enregistrer dans répertoire.

- **Étape 2** : Convertir les fichiers Swagger en Collections Postman.
- **Étape 3** : Interroger la base de donnée pour extraire les valeurs nécessaires pour tester les requêtes API.
- **Étape 4** : Injecter les valeurs dans les Collections Postman ainsi que les assertions pour préparer les Collections pour exécution.
- **Étape 5** : Exécuter les Collections Postman et générer les rapports.

N.B. : Ce processus s'applique à chaque module testé