

FICHE : PoC & CHOIX TECHNOLOGIQUES

=> Après avoir rempli cette fiche : convertir ce document en PDF et le téléverser dans votre dépôt Github dans un répertoire nommé 'doc'.

COMPTE RENDU DE LA VEILLE TECHNOLOGIQUE PoC

QUESTIONS DE RECHERCHE SUR LA PoC

Quel est l'opération la plus risquée de votre projet, l'élément qui risque de ne pas fonctionner ? Quelle est l'interaction entre deux technologies ? Quel est le PROBLÈME technique à résoudre ?

La difficulté du projet est de réussir à faire communiquer le background du site en python et le FrontEnd du site afin d'afficher les données sur le site.

Quelle est la QUESTION que vous vous posez et que vous demandez à internet de répondre ?

Est-il possible d'afficher des données de Python sur un site Web ?

POC = PREUVE DE CONCEPT

Quel genre de preuve de concept minimale pourrait valider que le problème n'existe pas ou qu'une solution a été trouvée ? Décrivez chaque élément du code requis.

Il faudrait que j'arrive à afficher sur mon site les données récupérer grâce aux fonctions Backend Python grâce à un système de requête HTTP par exemple.

LES MARQUE-PAGES IDENTIFIÉS LORS DE VOS RECHERCHES

Lien vers une page publique contenant vos marque-pages collaboratifs ou lister les marque-pages directement ici. Pour chaque lien : URL, nom de la page et description sommaire.

https://docs.google.com/document/d/1R_JbCxeOHsvfyqKL2hiUFDRRtiEtjiQ54liFv5IVnlo/edit?usp=sharing

LES PREUVES DE CONCEPT

Pour chaque preuve de concept réalisée : identifier le but de la preuve de concept (ce qu'elle vérifie), le lien vers le sous-répertoire de votre dépôt GitHub qui contient le code de la preuve de concept ainsi que les résultats de votre expérimentation, puis, finalement, vos conclusions.

Au moins une preuve de concept doit être documentée et réalisée.

PREMIÈRE POC RÉALISÉE

Preuve : Afficher des données sur l'interface web

URL Github : <https://github.com/cegepmatane/projet-specialise-2022-Lechiffre4>

EXPLIQUEZ VOTRE POC

Décrivez la Poc en détails.

Pour prouver que le projet est possible, il faudrait que je code une interface web capable de communiquer avec des fonctions Python. Il faudrait aussi que l'interface soit dynamique (modifie le HTML afin d'afficher les informations). Pour faire cela, il faut établir une communication entre Python Django et le HTML/CSS/JS avec des requêtes Ajax par exemple.

Que PROUVE la Poc ?

Qu'il est possible de faire le site web dynamique en Python avec des requêtes effectuées avec AJAX (javascript).

Que reste-t-il à prouver ?

Il reste a prouver que le nombre de requêtes de Web Scrapping n'est pas trop élevé. (Risque de Ban ip)

Quels sont vos résultats de la Poc ?

La cohabitation du Python et du JavaScript avec est possible. J'ai réussi a afficher des données récupérer avec Python sur mon site web grâce a des requêtes JS. L'affichage dynamique s'effectue avec Jinja.

PREMIÈRE TECHNOLOGIE SÉLECTIONNÉE (LA NOUVELLE)

Technologie : BeautifulSoup4

URL :

JUSTIFIER VOTRE CHOIX TECHNOLOGIQUE POUR CETTE TECHNOLOGIE

Expliquer à l'aide d'une argumentation rationnelle votre choix technologique. Établir votre justification à l'aide de liens avec les fonctionnalités, contraintes et risques de votre projet. Un tableau comparatif permettant de synthétiser votre réflexion pourrait être un apport judicieux à vos explications.

J'utilise cette technologie parce qu'elle allie la simplicité et l'efficacité pour le WebScrapping. Elle est aussi beaucoup documentée et utilisée par d'autres développeurs ce qui pourra m'être utile pour développer et résoudre mes erreurs et bugs.

Avantage

- Simple d'utilisation
- Bien documenté
- Grande communauté

Inconvénient

- effectue beaucoup de requêtes
- effectue uniquement du webscrapping

DEUXIÈME TECHNOLOGIE SÉLECTIONNÉE (LA CONNUE)

Technologie : JavaScript (Ajax)

URL :

JUSTIFIER VOTRE CHOIX TECHNOLOGIQUE POUR CETTE TECHNOLOGIE

Expliquer à l'aide d'une argumentation rationnelle votre choix technologique. Établir votre justification à l'aide de liens avec les fonctionnalités, contraintes et risques de votre projet. Un tableau comparatif permettant de synthétiser votre réflexion pourrait être un apport judicieux à vos explications.

Pour effectuer les requêtes entre le background de mon site et le frontend je dois effectuer des requêtes HTML. Pour cela, je compte utiliser AJAX. AJAX est très documenté et utilisé, de plus la majorité des tutos web en Python l'utilise. Il me sera plus facile de trouver de l'aide. En plus de cela, l'alternative FETCH est une technologie que je connais déjà ainsi je voudrais découvrir AJAX.

Avantage

- très documenté
- beaucoup utilisé
- Utilisé dans tout les tutos Python

Inconvénient

- Développement plus long (technologie inconnue pour moi)
- Plus lourd que Fetch a cause de JQuery