

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 ?

L'installation et l'utilisation du module Sagemath sur Python est un problème à résoudre puisque l'installation de ce module est prévue principalement sur Linux. En soit l'installation de Sagemath n'est pas compliqué sur Windows il suffit de télécharger le paquet sur internet destiné à Windows. Or pour utiliser Sagemath sur Python avec un OS Windows c'est plus compliqué car l'installation se fait en commandes Linux. Il faut donc trouver un moyen d'installer Sagemath Python sur Windows.

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

Comment installer Sagemath Python sous Windows ?

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.

Une preuve de concept minimale serait de pouvoir utiliser une fonction Sagemath en langage Python (exemple création d'une matrice) en étant sous un OS Windows. Un simple script Python utilisant une fonction Sagemath est suffisant à la PoC.

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.

SageWindows : <https://wiki.sagemath.org/SageWindows>

Description : guide d'installation de Sagemath 8.0 pour Windows et utilisation simple des différents éléments fournis lors de l'installation de Sagemath.

askSagemath : <https://ask.sagemath.org/question/39742/make-pycharm-recognise-the-sage-python-interpreter/>

Description : Comment faire pour que PyCharm reconnaisse l'interpréteur Python de Sagemath.

docSagemath : <https://doc.sagemath.org/html/en/faq/faq-usage.html>

Description : FAQ sur l'installation et l'utilisation de Sagemath.

Reddit : <https://doc.sagemath.org/html/en/faq/faq-usage.html>

Description : Question sur des problèmes d'importation du module Sagemath sur Python.

Le Crabeinfo : <https://lecrabeinfo.net/installer-wsl-windows-subsystem-for-linux-sur-windows-10.html>

Description : Installer WSL (Windows Subsystem for Linux) sous Windows 10.

DocSagemath : <https://doc.sagemath.org/pdf/en/installation/installation.pdf>

Description : Guide détaillé d'installation de Sagemath.

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 : Le module SageMath fonctionne Python

URL Github : <https://github.com/cegepmatane/projet-specialise-2022-AJTheEnder/commit/6372a5c9f9efc54f089307671daa126096aeded1>

EXPLIQUEZ VOTRE POC

Décrivez la POC en détails.

Cette PoC vérifie la compatibilité entre le module SageMath Python et l'OS Windows. En effet sous Windows le seul moyen d'installer SageMath est le .zip que l'on peut trouver sur le site de SageMath or ce .zip ne donne que des outils comme le créateur de script SageMath ou le shell SageMath et les commandes Windows ne permettent pas d'installer les packages nécessaires à SageMath Python. Cela

n'est possible que sur Linux où le package manager s'occupe d'installer les dépendances entre Python et SageMath. Le but est donc d'utiliser Linux sous Windows grâce à WSL. C'est à partir de là que l'on peut installer SageMath pour Python. Au final le but sera de pouvoir utiliser des fonctions comme `sage.all.mod(a, b)`, `sage.all.matrix(...)`, `sage.all.factorial(n)` etc...

Que PROUVE la Poc ?

La PoC prouve que le module SageMath a bien été importer dans un script Python sur un OS Windows et qu'on le peut utiliser les différentes fonctions de SageMath à l'intérieur d'un Script.

Que reste-t-il à prouver ?

Il ne reste rien à prouver.

Quels sont vos résultats de la Poc ?

Des résultats concluants ont été obtenue après l'installation de WSL sur Windows 10, l'installation de Ubuntu sur ma machine et l'importation des pakages SageMath. En effet j'ai obtenu un script Python utilisant des fonctionnalités de SageMath, ce script s'exécute via lignes de commande Ubuntu et le script ne présente pas d'erreur. Je peux donc actuellement utiliser le module SageMath sur Python.

Script :

```
testSagemath.py 5 ×
CryptoBib > testSagemath.py > ...
1  from sage.all import *
2
3  factorielTest = sage.all.factorial(9)
4
5  matriceTestA = sage.all.matrix([[1, 2, 3], [3, 2, 1], [5, 0, 2]])
6  matriceTestB = sage.all.matrix([[5, 6, 1], [0, 0, 4], [3, 4, 2]])
7
8  moduloTest = sage.all.mod(30, 5)
9
10 print(factorielTest)
11 print(matriceTestA * matriceTestB)
12 print(moduloTest)
```

Résultat :

```
ajtheender@LAPTOP-0GRVT9L6:/mnt/d/Cours/2èmeAnnée/Projet Spe/projet-specialise-2022-AJTheEnde/CryptoBib$ python3 testSage
math.py
362880
[14 18 15]
[18 22 13]
[31 38 9]
0
```

PREMIÈRE TECHNOLOGIE SÉLECTIONNÉE (LA NOUVELLE)

Technologie : Sagemath

URL : <https://www.sagemath.org/>

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 car c'est devenu une référence en module mathématique au côté de géant comme Matlab ou Wolfram qui sont certes plus puissant, avec plus de possibilités mais également beaucoup plus compliqué à apprendre et à maîtriser. De plus pour un projet d'une durée de 10 semaines sans des besoins en mathématiques d'un niveau très avancé, utiliser un module comme Matlab ou Wolfram qui est très demandant en espace serait un peu « overkill ».

DEUXIÈME TECHNOLOGIE SÉLECTIONNÉE (LA CONNUE)

Technologie : Windows

URL : <https://docs.microsoft.com/en-us/windows/>

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.

Windows étant l'OS installé sur mon ordinateur l'utilisation de celui-ci semble évidente, pour un projet aussi court installer un autre OS ne serait pas judicieux. Une autre solution serait d'installer une machine virtuelle Linux pour pouvoir travailler sur le projet mais cela affectera les performances car cela donnera de la programmation par interposition. De plus Windows est tout de même l'OS utilisé par environ 75% des utilisateurs de PC de bureau, Windows est un OS fiable et performant pour la programmation.