## Splash :

Splash est un navigateur Web headless (sans interface graphique) conçu principalement pour l'automatisation de tâches liées au web, telles que le scraping de données depuis des sites Web. Splash est souvent utilisé pour le web scraping, en particulier lorsque vous devez interagir avec des sites Web qui utilisent JavaScript de manière intensive. Splash utilise un moteur de rendu Web basé sur WebKit, ce qui lui permet d'exécuter JavaScript, de charger des pages Web, et de prendre des captures d'écran. Il est généralement utilisé avec des bibliothèques Python telles que Scrapy pour simplifier le processus de collecte de données sur le web.

Splash : Splash est une bibliothèque Python conçue pour le web scraping et le rendu de pages web, en particulier pour les pages web générées de manière dynamique à l'aide de JavaScript. Splash utilise un moteur de rendu basé sur WebKit pour charger et afficher des pages web, ce qui permet de manipuler et de récupérer des données à partir de ces pages, y compris le contenu généré par JavaScript. Splash est souvent utilisé en conjonction avec le framework Scrapy pour faciliter le web scraping.

Pour utiliser Splash avec Scrapy, vous devez d'abord configurer votre projet Scrapy pour qu'il puisse interagir avec Splash en tant que service rendu.
**Pour utiliser Splash, vous devez d'abord l'installer et exécuter un serveur Splash local ou distant. Vous pouvez installer Splash en utilisant Docker ou directement depuis le code source.** Voici un exemple de configuration et d'utilisation de Splash avec Scrapy :

1. Assurez-vous d'avoir Splash installé et en cours d'exécution. Vous pouvez le faire via Docker ou directement sur votre système. Veuillez vous référer à la documentation de Splash pour plus d'informations sur l'installation.
2. Installez Scrapy et scrapy-splash si vous ne l'avez pas déjà fait. Vous pouvez le faire en utilisant pip

Configurez Scrapy pour utiliser Splash en définissant les paramètres dans le fichier de configuration settings.py de votre projet Scrapy :

```
# settings.py

SPLASH_URL = 'http://localhost:8050' # URL de l'API Splash
DOWNLOADER_MIDDLEWARES = {
 'scrapy_splash.SplashCookiesMiddleware': 723,
 'scrapy_splash.SplashMiddleware': 725,
'scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware': 810,
}
SPIDER_MIDDLEWARES = {
 'scrapy_splash.SplashDeduplicateArgsMiddleware': 100,
}
```

Créez un spider Scrapy qui utilise Splash pour effectuer le rendu de pages web. Voici un exemple de spider Scrapy qui utilise Splash pour extraire le contenu d'une page :

python Copy code

```python
import scrapy
from scrapy_splash import SplashRequest

class MySpider(scrapy.Spider):
    name = 'my_spider'

    def start_requests(self):
        url = 'http://example.com'
        yield SplashRequest(url, self.parse, args={'wait': 2})

    def parse(self, response):
        # Traitez la réponse ici
        # Par exemple, extrayez des données à l'aide de Selectors
        title = response.css('title::text').get()
        self.log(f'Titre de la page : {title}')

        # Continuez à naviguer ou à effectuer d'autres actions ici
```

Dans cet exemple, le spider commence par envoyer une requête Splash à l'URL de la page que vous souhaitez extraire. Vous pouvez également passer des arguments supplémentaires à Splash, comme le délai d'attente. Ensuite, le callback parse est appelé pour traiter la réponse rendue par Splash.
**Assurez-vous que Splash est en cours d'exécution avant de lancer votre spider. Vous pouvez exécuter Splash en utilisant la commande Docker ou directement sur votre système.**
N'oubliez pas d'ajuster votre spider Scrapy en fonction de vos besoins spécifiques et d'utiliser des Selectors pour extraire les données souhaitées de la réponse rendue par Splash.

voici un exemple concret pour illustrer pourquoi vous pourriez avoir besoin d'utiliser Splash avec Scrapy. **Supposons que vous souhaitiez scraper un site web de e-commerce où les données des produits sont chargées dynamiquement à l'aide de JavaScript. Sans Splash, l'extraction des données à partir de ce site serait difficile puisque les donnes ne sont pas texte comme  Product 1 ... .**
**Supposons que vous souhaitez scraper un site de vente aux enchères en ligne où les informations sur les enchères sont chargées dynamiquement via JavaScript. Le site utilise une pagination infinie pour charger de nouvelles enchères à mesure que l'utilisateur fait défiler la page.**
**Supposons que vous souhaitiez scraper un site Web de médias sociaux qui charge dynamiquement son contenu à l'aide de JavaScript, et que vous vouliez extraire des commentaires de publications,Ce type de tâche nécessite l'utilisation de Splash avec Scrapy.**
**Supposons que vous souhaitez scraper un site de streaming vidéo où les informations sur les vidéos sont chargées de manière asynchrone via des appels JavaScript après le chargement initial de la page. Le contenu que vous souhaitez extraire n'est pas présent dans le code source HTML initial, mais il est généré dynamiquement à l'aide de JavaScript.**