Partie extraction pour l’Excel :

La solution technique retenue est de faire un ou plusieurs prompt pour chaque case du tableau Excel. Cela permet d’être précis sur chaque prompt et simplifie la démarche d’ajouter une case si nécessaire. L’avocat n’a pas besoin de traiter énormément de bail en même temps : le nombre de « invoke\_model » excessif n’est pas un problème si le nombre de requête possible en même temps est débridé., sinon il faut compter 20minutes pour traiter un document.

Chaque case est associée à un tag présent dans excel\_tag\_column\_correspondace.py

Pour chaque tag il y a une ou plusieurs prompt dans excel\_prompt\_page\_x.py

Toute les prompt sont exécuté de manière sucessive serverless avec « invoke\_model », l’entrée du dernier prompt est la sortie du précédent, leur résultat mis en valeur d’un dictionnaire où la clef est le tag

generate\_excel.py prend ensuite ce dictionnaire et les correspondances case-tag pour remplir le excel.

Partie extraction pour le word :

La structure du word est plus compliquée.

Ici il y a une série de prompt (dans prompts.py) pour chaque grande catégorie du excel. La sortie du premier prompt est l’entrée du suivant. En général il y a un prompt qui collecte toutes les information et un prompt qui formate ces informations au format json.

A la fin l’on dispose d’un fichier json compliqué. Chaque valeur est remplacée dans le excel model par la valeur trouvée.