

# Implémentation des logs

On a deux fichiers de logs, un contenant tous les envois, et un autre contenant que les erreurs.

Name	Date modified	Type	Size
2024-02-20-log	20/02/2024 15:50	Text Document	1 KB
2024-02-20-logErreur	20/02/2024 15:50	Text Document	1 KB
2024-02-21-log	21/02/2024 16:29	Text Document	9 KB
2024-02-21-logErreur	21/02/2024 14:38	Text Document	1 KB
2024-02-22-log	22/02/2024 14:47	Text Document	6 KB
2024-02-22-logErreur	22/02/2024 09:26	Text Document	0 KB
2024-02-23-log	23/02/2024 16:44	Text Document	5 KB
2024-02-23-logErreur	23/02/2024 16:44	Text Document	3 KB

2024-02-23-logErreur

File Edit View

```
2024-02-23 14:20:09,384 - ERROR - FA24/00143: Il y a des données manquantes dans les lignes et colonnes:
(1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
(1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
(1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
(1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
(1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:41:54,450 - ERROR - FA24/00206: Il y a des données manquantes dans les lignes et colonnes:
(1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
(2, 'SIRET')
(2, 'CountryID')
(2, 'CountryIDLivr')
(3, 'SIRET')
(3, 'CountryID')
(3, 'CountryIDLivr')
```

2024-02-23-log

File Edit View

```
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
(1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
(1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
(1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:38:25,032 - INFO - fa24_00204FactX.pdf
Réponse de l'API : {
  "codeRetour" : 0,
  "libelle" : "GCU_MSG_01_000",
  "numeroFluxDepot" : "CPP001111700000000305658",
  "dateDepot" : "2024-02-23",
  "syntaxeFlux" : "IN_DP_E2_CII_FACTURX"
}
2024-02-23 16:38:25,032 - INFO - fa24_00204FactX.pdf
Réponse de l'API : {
  "codeRetour" : 0,
  "libelle" : "GCU_MSG_01_000",
  "numeroFluxDepot" : "CPP001111700000000305658",
  "dateDepot" : "2024-02-23",
  "syntaxeFlux" : "IN_DP_E2_CII_FACTURX"
}
```

Initialisation des logs :

```
import logging
from datetime import datetime

current_datetime = datetime.now()
# Formater la date et l'heure
formatted_datetime = current_datetime.strftime("%Y-%m-%d")

# Configure logging for the first log file
log = logging.getLogger('log')
log.setLevel(logging.INFO)
log_formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')
file_handler = logging.FileHandler(fr'C:\temp\v4\factures\logs\{formatted_datetime}-log.log')
file_handler.setFormatter(log_formatter)
log.addHandler(file_handler)

logErr = logging.getLogger('logErr')
logErr.setLevel(logging.ERROR)
logErr_formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')
fileErr_handler = logging.FileHandler(fr'C:\temp\v4\factures\logs\{formatted_datetime}-logErreur.log')
fileErr_handler.setFormatter(logErr_formatter)
logErr.addHandler(fileErr_handler)
```

Ajout des messages dans les logs (exemple getXML.py):

```
# Vérification des données pour s'assurer qu'il n'y a pas de valeurs manquantes
def verifData(data, noFacture, file_path, errorPath):
    vide = []
    for i in range(len(data)):
        for key, val in data[i].items():
            if val == '':
                vide.append((i+1, key))

    if vide:
        error_message = f"{noFacture}: Il y a des données manquantes dans les lignes et colonnes:\n " + '\n'.join([f"({idx}, '{key}')" for idx, key in vide])
        destination_path = os.path.join(errorPath, os.path.basename(file_path))
        shutil.move(file_path, destination_path)
        print("err " + error_message)
        logErr.error(error_message)
        log.error(error_message)
        return False
    return True
```

(exemple sendFact.py) :

```
# Afficher la réponse de l'API
print(f"{file_name} \n Réponse de l'API :", response.text)
log.info(f"{file_name} \n Réponse de l'API : {response.text}")
```

