**DST Airlines – ETAPE 01**

1. **Récolte des données : Consignes**
   1. **Objectifs**

Passer par l’API de [Lufthansa](https://developer.lufthansa.com/docs) pour récupérer des données sur les vols.

Vous pouvez tester les différentes routes de l’API de Lufthansa à l’aide de ce [lien](https://developer.lufthansa.com/io-docs). Vous serez amené à aller puiser différentes informations comme les codes IATA (il faudra les récupérer via Webscraping).

Cette étape est importante, vous devez comprendre les données que vous pouvez récupérer et faire un choix des routes à utiliser.

Il y a aussi l’API de [Internatinal Airlines](https://developer.iairgroup.com/), mais il se peut que vous ayez des soucis pour y accéder..

1. **Modules du parcours**

Utilisation de la librairie requests ou de l’outil Postman (pour tester) ou Techniques de webscraping

1. **Attendu**

Fichier explicatif du traitement et des différentes données accessible (doc / pdf)  
Un exemple de données collectées

1. **Récolte des données : livrables** 
   1. **Le périmètre des sources de données :**

Utilisation de 3 APIs :

* API LUFTHANSA :<https://api.lufthansa.com/v1>
* API AVIATION STACK : http://api.aviationstack.com/v1
* API AIR LABS : https://airlabs.co/api/v9
  1. **Extraction des données avec la librairie Requests :**

Exemple extraction avec l’API Lufthansa:

Fichier : testLUF.py (Fichier d’utilisation du module DST.apiLufthansa et des fonctions getToken et getData et récupération des fichiers JSON)

from DSTModules.apiLufthansa import getToken, getData

from DSTModules.tools import writeFile

# Déclaration constantes

pathOut = "./FilesOut\_Lufthansa"

# Récupération du token

token = getToken()

#----------------------------------------

# Récupération des données de références

#----------------------------------------

#v1/mds-references/countries/{countryCode}

#v1/mds-references/cities/{cityCode}

#v1/mds-references/airports/{airportCode}

#v1/mds-references/airports/nearest/{latitude},{longitude}

#v1/mds-references/airlines/{airlineCode}

#v1/mds-references/aircraft/{aircraftCode}

listReference = ["countries", "cities", "airports", "airlines", "aircraft"]

for reference in listReference:

    result = getData(token, "mds-references/" + reference)

    writeFile(pathOut+"/"+reference.replace("/","\_")+".json", result)

#----------------------------------------

# Récupération des données Opérations

#----------------------------------------

#v1/operations/customerflightinformation/{flightNumber}/{date}

#v1/operations/customerflightinformation/route/{origin}/{destination}/{date}

#v1/operations/customerflightinformation/arrivals/{airportCode}/{fromDateTime}

#v1/operations/customerflightinformation/departures/{airportCode}/{fromDateTime}

#v1/operations/schedules/{origin}/{destination}/{fromDateTime}

listOperations = ["customerflightinformation/LH400/2022-10-30",

    "customerflightinformation/route/FRA/JFK/2022-10-30",

    "customerflightinformation/arrivals/ZRH/2022-10-30T10:00",

    "customerflightinformation/departures/HAM/2022-10-30T10:00",

    "schedules/ZRH/FRA/2022-10-30"]

for operation in listOperations:

    result = getData(token, "operations/" + operation)

    writeFile(pathOut+"/"+operation.replace("/","\_")+".json", result)

#----------------------------------------

# Récupération des données de Vols

#----------------------------------------

params={

    'airlines':'LH',

    'flightNumberRanges':'400-405',

    'startDate':'05DEC22',

    'endDate':'10DEC22',

    'daysOfOperation':'1234567',

    'timeMode':'UTC'

}

result = getData(token, "flight-schedules/flightschedules/passenger", params)

writeFile(pathOut+"/flightschedules\_passenger.json", result)

#result = getData(token, "flight-schedules/flightschedules/cargo", params)

#writeFile(pathOut+"/flightschedules\_cargo.json", result)

#result = getData(token, "flight-schedules/flightschedules", params)

#writeFile(pathOut+"/flightschedules.json", result)

Fichier : apiLufthansa.py (Fichier contenant le code python avec usage de la bibliothèque requests et des données de paramètrages)

#----------------------

# Client API LUFTHANSA

#----------------------

import requests

import json

from pprint import pprint

#

# Déclaration des constantes

urlToken = "https://api.lufthansa.com/v1/oauth/token"

clientId = "wxrnjrr63u4pwy5xb6kp3be4"

clientPass = "8ykm4YUp4Da27TNnj8Uy"

urlLuft = "https://api.lufthansa.com/v1"

#

# GET TOKEN

def getToken():

    headers={

        'Accept': 'application/json',

        'Content-Type':'application/x-www-form-urlencoded'

    }

    data = {'client\_id':clientId,'client\_secret': clientPass, 'grant\_type': 'client\_credentials'}

    response = requests.post(urlToken, headers=headers, data=data)

    if response.ok:

        result = response.json()

        return result["access\_token"]

    else:

        raise Exception("Erreur dans la récupération du token:\n"+response.reason)

#

# GET DATA

def getData(token, uri, params={}):

    headers = {

            "Authorization" : "Bearer "+token,

            "Accept" : "application/json",

            "Content-type" : "application/json"

    }

    #url = urlLuft+"/"+uri+"/"+filtre

    url = urlLuft+"/"+uri

    print(url)

    response = requests.get(url, headers=headers, params=params)

    if response.ok:

        result = json.dumps(response.json(), indent=4)

        return result

    else:

        raise Exception("Erreur get "+uri+":\n"+str(response.status\_code)+" "+response.reason)

* 1. **Sources de données API Lufthansa :**

Ressources :

https://api.lufthansa.com/v1/mds-references/countries

https://api.lufthansa.com/v1/mds-references/cities

https://api.lufthansa.com/v1/mds-references/airports

https://api.lufthansa.com/v1/mds-references/airlines

https://api.lufthansa.com/v1/mds-references/aircraft

https://api.lufthansa.com/v1/operations/customerflightinformation/LH400/2022-10-30

https://api.lufthansa.com/v1/operations/customerflightinformation/route/FRA/JFK/2022-10-30

https://api.lufthansa.com/v1/operations/customerflightinformation/arrivals/ZRH/2022-10-30T10:00

https://api.lufthansa.com/v1/operations/customerflightinformation/departures/HAM/2022-10-30T10:00

https://api.lufthansa.com/v1/operations/schedules/ZRH/FRA/2022-10-30

https://api.lufthansa.com/v1/flight-schedules/flightschedules/passenger

Exemple de résultat pour la ressource airports:

{

"AirportResource": {

"Airports": {

"Airport": {

"AirportCode": "CDG",

"Position": {

"Coordinate": {

"Latitude": 49.0097,

"Longitude": 2.5478

}

},

"CityCode": "PAR",

"CountryCode": "FR",

"LocationType": "Airport",

"Names": {

"Name": [

{

"@LanguageCode": "DE",

"$": "Paris/ Ch.de Gaulle"

},

{

"@LanguageCode": "EN",

"$": "Paris/ Ch.de Gaulle"

},

{

"@LanguageCode": "ES",

"$": "Paris/Ch.de Gaulle"

},

{

"@LanguageCode": "FR",

"$": "Paris/Ch. de Gaulle"

},

{

"@LanguageCode": "IT",

"$": "Parigi/Ch.de Gaulle"

},

{

"@LanguageCode": "JA",

"$": "\u30d1\u30ea"

},

{

"@LanguageCode": "KO",

"$": "\ud30c\ub9ac"

},

{

"@LanguageCode": "NL",

"$": "Parijs/Ch. de Gaulle"

},

{

"@LanguageCode": "PL",

"$": "Pary\u017c"

},

{

"@LanguageCode": "PT",

"$": "Paris/Ch. de Gaulle"

},

{

"@LanguageCode": "RU",

"$": "\u041f\u0430\u0440\u0438\u0436"

},

{

"@LanguageCode": "ZH",

"$": "\u5df4\u9ece"

}

]

},

"UtcOffset": "+01:00",

"TimeZoneId": "Europe/Paris"

}

},

"Meta": {

"@Version": "1.0.0",

"Link": [

{

"@Href": "https://api.lufthansa.com/v1/mds-references/airports/CDG",

"@Rel": "self"

},

{

"@Href": "https://api.lufthansa.com/v1/mds-references/cities/PAR",

"@Rel": "related"

},

{

"@Href": "https://api.lufthansa.com/v1/mds-references/countries/FR",

"@Rel": "related"

},

{

"@Href": "http://travelguide.lufthansa.com/de/de/{cityCode}/CDG",

"@Rel": "alternate"

},

{

"@Href": "http://travelguide.lufthansa.com/de/en/{cityCode}/CDG",

"@Rel": "alternate"

},

{

"@Href": "http://travelguide.lufthansa.com/de/cn/{cityCode}/CDG",

"@Rel": "alternate"

},

{

"@Href": "http://travelguide.lufthansa.com/de/pt/{cityCode}/CDG",

"@Rel": "alternate"

}

]

}

}

}

* 1. **Sources de données API Aviation Stack :**

Ressources :

http://api.aviationstack.com/v1/airports

http://api.aviationstack.com/v1/airlines

http://api.aviationstack.com/v1/airplanes

http://api.aviationstack.com/v1/aircraft\_types

http://api.aviationstack.com/v1/taxes

http://api.aviationstack.com/v1/cities

http://api.aviationstack.com/v1/countries

http://api.aviationstack.com/v1/flights

Exemple de résultat pour la ressource filghts :

{

"flight\_date": "2022-10-31",

"flight\_status": "active",

"departure": {

"airport": "Indira Gandhi International",

"timezone": "Asia/Kolkata",

"iata": "DEL",

"icao": "VIDP",

"terminal": "3",

"gate": "26",

"delay": 26,

"scheduled": "2022-10-31T13:35:00+00:00",

"estimated": "2022-10-31T13:35:00+00:00",

"actual": "2022-10-31T14:00:00+00:00",

"estimated\_runway": "2022-10-31T14:00:00+00:00",

"actual\_runway": "2022-10-31T14:00:00+00:00"

},

"arrival": {

"airport": "Frankfurt International Airport",

"timezone": "Europe/Berlin",

"iata": "FRA",

"icao": "EDDF",

"terminal": "1",

"gate": "B44A",

"baggage": null,

"delay": null,

"scheduled": "2022-10-31T18:00:00+00:00",

"estimated": "2022-10-31T18:00:00+00:00",

"actual": null,

"estimated\_runway": null,

"actual\_runway": null

},

"airline": {

"name": "Air India",

"iata": "AI",

"icao": "AIC"

},

"flight": {

"number": "121",

"iata": "AI121",

"icao": "AIC121",

"codeshared": null

},

"aircraft": null,

"live": null

}

* 1. **Sources de données API Air Labs :**

Ressources :

https://airlabs.co/api/v9/airports

https://airlabs.co/api/v9/airlines

https://airlabs.co/api/v9/cities

https://airlabs.co/api/v9/fleets

https://airlabs.co/api/v9/routes

https://airlabs.co/api/v9/countries

https://airlabs.co/api/v9/timezones

https://airlabs.co/api/v9/taxes

https://airlabs.co/api/v9/flights

https://airlabs.co/api/v9/schedules

Exemple de résultat pour la ressource schedules :

{

"airline\_iata": "QT",

"airline\_icao": "TPA",

"flight\_iata": "QT4099",

"flight\_icao": "TPA4099",

"flight\_number": "4099",

"dep\_iata": "MIA",

"dep\_icao": "KMIA",

"dep\_terminal": null,

"dep\_gate": null,

"dep\_time": "2022-10-31 06:00",

"dep\_time\_utc": "2022-10-31 10:00",

"dep\_estimated": "2022-10-31 06:04",

"dep\_estimated\_utc": "2022-10-31 10:04",

"dep\_actual": "2022-10-31 06:04",

"dep\_actual\_utc": "2022-10-31 10:04",

"arr\_iata": "EZE",

"arr\_icao": "SAEZ",

"arr\_terminal": null,

"arr\_gate": null,

"arr\_baggage": null,

"arr\_time": "2022-10-31 16:00",

"arr\_time\_utc": "2022-10-31 19:00",

"cs\_airline\_iata": null,

"cs\_flight\_number": null,

"cs\_flight\_iata": null,

"status": "active",

"duration": 540,

"delayed": null,

"dep\_delayed": null,

"arr\_delayed": null,

"aircraft\_icao": "A332",

"arr\_time\_ts": 1667242800,

"dep\_time\_ts": 1667210400,

"dep\_estimated\_ts": 1667210640,

"dep\_actual\_ts": 1667210640

}