# BIG DATA PROJECT

## Synopsis

The goal of the 'Big Data project' is to create a simple end-to-end data architecture, includingdata ingestion, data transformation and data exposition.

In this project, you are free to decide which data you would like to fetch and determine the finaloutput that will create value from your data inputs.The only necessary thing to respect is a Datalake architecture, which will permit us to structurethe data correctly, to have a clean data pipeline and shareable data.

## Architecture

Here are the jobs you need to do:

1.Create jobs that extract data from the Internet via REST API.

2. Create jobs that format / normalize these data to transform them into a format suitable toa Datalake /data sharing /data analysis.

3. Create a job that joins / combines your different data sources into a final interesting3.

usage.

4. Index your data in a dashboard to expose your final output to end users

The final architecture will be like this:

图示

描述已自动生成

## Steps

Here we detailed the necessary steps to realize your project:

### Step 1.1: Decide your theme for data

Choose a theme that you like. One of your passions, or something that interests you.

Example: Health, Sport, Cinema, Security (Crimes), Environment, real estate ...

Default: lf you don't have any idea, we suggest doing it with Cinema.

### Step 1.2: Find Datasource to fetch on Internet

Find on Internet data sources you would like to fetch. You should have at least 2 different datasources because the goal of the Datalake is to cross data.One of the 2 data sources should be “realtime" (at least daily refresh), so the dashboard can beinteresting to follow every day.

Example:

* open data (https://www.data.gouv.fr/fr/ as example)
* data scraping
* free data apis.
* https://rapidapi.com/
* A list of public apis: https://github.com/toddmotto/public-apis#newsSocial Network data(Twitter tweets, Google trends ...)

### Step 2: Data Pipeline

We advise you to create your whole data pipeline as 1 Airflow DAG

### Step 2.1: Ingestion

Here you'l create scripts to fetch data on the dedicated APls and store them into your datalake

Tips:

* With Python, you can find an SDK to help you fetch data from different platforms. Example with Twitter: https://developer.twitter.com/en/docs/twitter-api/tools-and-libraries/v2 this will be easier than doing manual HTTP requests (not possible since 2023 because of Elon Musk).
* There is many tools to help scrap the web, beautiful Soup is one them:https://python.doctor/page-beautifulsoup-html-parser-python-library-xm

### Step 2.2: Formatting

When your data arrives in the Data Lake, it's in CSV, jSON... lt's not a good format to do data analysis. You should therefore prepare your data in the layer “formatted/normalize" to convert in the necessary format (parquet ?), apply transformation (date in UTC for example), and also normalize the columns (each column should be clean and have clean values, to be easily re-usable)

You can do it in pure python with the pandas library but the best would be to use parallel scalable processing (Spark or DBT).

### Step 2.3: Combination

This is all the goal of the Datalake, being able to cross multiple different data sources to create value.

Your goal is to create a job that combines / joins your different data sources in one final output that creates value.

The job can be a simple join and then do some aggregation to generate KPl (sum, count, average...).

Exampl:

* Do recommandation
* Anomaly detection
* Fraud analysis
* Resources optimization

Links to example:

* https://www.upgrad.com/blog/big-data-project-ideas-beginners/
* https://www.projectpro.io/article/top-20-big-data-project-ideas-for-beainners-in-2021/426https:/medium.com/@LearnBay.co/big-data-is-an-exciting-subject-of-information-technoogy-cae55cb583ab

Default:

* Recommendation of movies

### Step 2.4: Indexing

Index your result in Elasticsearch to expose the results in a Kibana dashboard.

图形用户界面, 应用程序

描述已自动生成

### Your deliverable

You should deliver your work in 3 parts:

* One video where you show your work (10 min maximum)
* One zip containing all your code
* One PDF where you explain the different parts of your work (10 pages maximum)

表格

描述已自动生成

表格

描述已自动生成

表格

描述已自动生成

\* Ingest in a distributed file system: lt could be S3 / OVH storage / GCS. In local you emulate local S3 with https://github.com/ocalstack/localstack