

PROJET DATA, DÉVELOPPEMENT ET ARCHITECTURE DE DONNÉES

URBAN DATA EXPLORER

MERIEM BENNACER



PROJET: URBAN DATA EXPLORER

O B J E C T I F :

Construire une plateforme complète (pipeline + API + dashboard)

T H É M A T I Q U E :

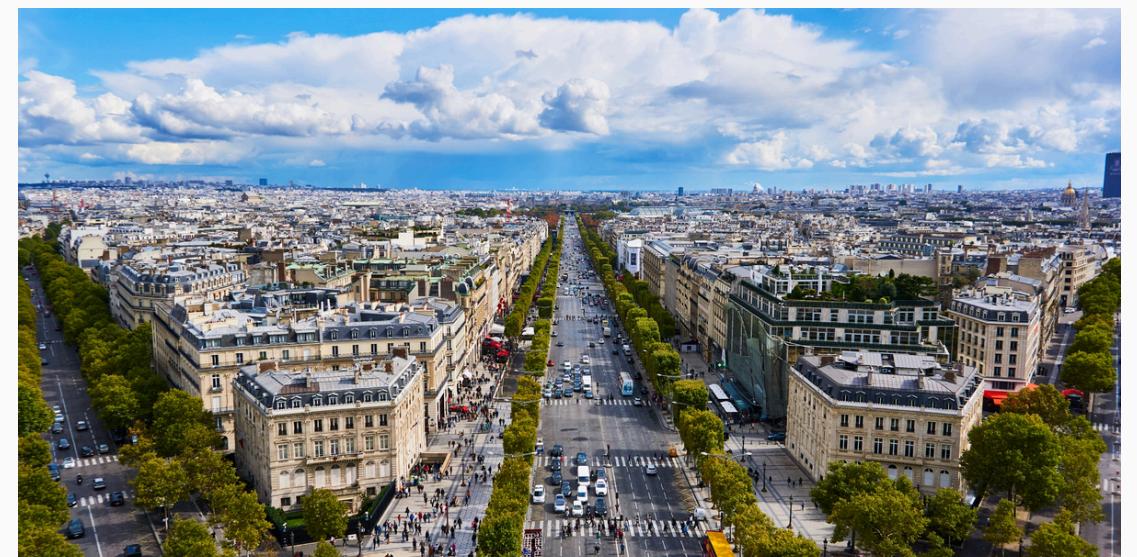
Le marché immobilier à Paris

F I N A L I T É

Analyser, comparer et visualiser des indicateurs clés (prix, logements sociaux, typologie, évolution...)

R É S U L T A T A T T E N D U :

Un dashboard interactif explorant la ville arrondissement par arrondissement



OBJECTIF PÉDAGOGIQUES DU PROJET

Points clés :

- Concevoir une architecture data robuste (zones Bronze / Silver / Gold)
- Intégrer plusieurs sources d'API publiques (DataGouv, INSEE, etc.)
- Créer une API performante pour exposer vos données
- Développer une interface cartographique interactive en JS
- Travailler la qualité, traçabilité, versioning et automatisation
- Construire un dashboard narratif et accessible



APERÇU DU RÉSULTAT ATTENDU:

Contenu :

Identifier ce que vous allez mesurer et d'où viennent vos données.

- Un dashboard web déployé, affichant :
 - Prix/m² médian par arrondissement
 - Part de logements sociaux
 - Typologie (Studios, T2, etc.)
 - Évolution dans le temps
- Interactivité :
 - Survol ou clic sur un arrondissement
 - Timeline animée
 - Mode comparaison (arrondissement A vs B)
- Données servies via API maison
- Code et documentation sur GitHub



ARCHITECTURE TECHNIQUE GLOBALE

Sources ouvertes (APIs, CSV, JSON)



Zone Bronze (Raw)



Zone Silver (Cleaned & Géocodée)



Zone Gold (Agrégée & Analytique)



API Web (FastAPI / Flask)



Dashboard interactif (JS + Mapbox / Deck.gl)

- Avoir un pipeline data automatisé
- Des données propres, enrichies et versionnées
- Un serveur backend qui alimente le frontend dynamique

ÉTAPE 1 : CADRAGE ET CHOIX DES INDICATEURS

Objectif :

Identifier ce que vous allez mesurer et d'où viennent vos données.

À faire :

1. Choisir les indicateurs principaux :

- Prix/m² médian -Valeur médiane des ventes par arrondissement - Suivre le marché
- Évolution temporelle - Variation du prix dans le temps -Identifier les tendances
- Logements sociaux (%) - Part des logements sociaux dans chaque zone - Mesurer la mixité sociale
- Typologie des logements - Type (appartement, maison, nombre de pièces) - Comprendre la structure du parc

2. Proposer 4 indicateurs personnalisés, par ex :

Pollution / qualité de l'air

Délits enregistrés

Revenus moyens

Densité de population

3. Identifier les sources de données :

Data.gouv, OpenData Paris, INSEE, Airparif,

ÉTAPE 2: INGESTION & COLLECTE DES DONNÉES

Objectif :

Automatiser la récupération de plusieurs sources.

À faire :

- Utiliser Python + requests + pandas pour appeler des APIs.
- Stocker les données brutes dans un dossier /data/bronze.
- Gérer les erreurs (codes HTTP, données manquantes).

```
import requests, pandas as pd

url = "https://data.gouv.fr/api/dvf"
data = requests.get(url).json()
df = pd.DataFrame(data)
df.to_csv("data/bronze/transactions.csv")
```

ÉTAPE 3 : NETTOYAGE, NORMALISATION ET GÉOCODAGE

Objectif :

Transformer vos données brutes en données exploitable.

À faire :

- Nettoyer les colonnes (types, doublons, NA...)
- Géocoder les adresses via API BAN ou Nominatim (OpenStreetMap)
- Uniformiser les formats (dates, prix, surfaces)
- Fusionner les jeux de données
- Sauvegarder les données.

ÉTAPE 4 : AGRÉGATION ET ENRICHISSEMENT

Objectif :

Produire des tables prêtes pour la visualisation.

À faire :

- Calculer le prix médian par arrondissement et par année.
- Calculer la variation annuelle (%).
- Fusionner les données socio-éco (revenus, densité...).
- Exporter au format Parquet / CSV / GeoJSON.

Arrondissement	Année	Prix_m2	Variation	Logements_sociaux	Revenu_moyen

ÉTAPE 5 : API BACKEND

Objectif :

Créer une API REST pour que le dashboard accède aux données.

Technos :

- FastAPI ou Flask

Endpoints types :

- /arrondissements
- /prix?annee=2023
- /comparaison?arr1=1&arr2=6
- /timeline?arr=6

ÉTAPE 6: FRONTEND & VISUALISATION

Objectif :

Rendre vos données vivantes sur une carte dynamique.

Technos :

- JavaScript + Mapbox / MapLibre / Deck.gl pour la carte
- Plotly.js / Chart.js pour les graphes
- HTML + CSS pour le design

Fonctionnalités clés :

- Carte choroplète → couleurs selon prix
- Survol → affiche les infos
- Sélecteur d'année
- Comparaison entre deux zones
- Timeline d'évolution

ÉTAPE 7 : DOCUMENTATION & DÉPLOIEMENT

A faire:

- Documenter le code et l'architecture
- Fournir un data catalog
- Déployer :
 - Backend : Render / Railway / Heroku
 - Frontend : GitHub Pages / Netlify
- Publier le code sur GitHub (README + schémas)

Livrables finaux :

- Code complet (pipeline + API + front)
- Documentation technique
- Jeu de données test
- Dashboard déployé
- Présentation / storytelling