

Suivi des candidatures dans une entreprise tech:

`https://github.com/Arbia-riahi/projet-web-services_graphql.git`

1. Solution proposee

- **Analyse du Problème:**

Actuellement, l'entreprise gère les candidatures par email. Cela pose plusieurs problèmes :

- **perte de données** ou d'historique
- **réponses tardives** ou oubliées
- **aucune traçabilité** des statuts ou décisions prises
- **difficulté à prioriser** ou filtrer les candidats

-> Une API centralisée via GraphQL permettra de suivre précisément les candidatures, leur statut et les actions prises à chaque étape du processus

Entités et Relations

- **Candidat** : personne qui soumet une candidature
- **Candidature** : enregistrement de la demande pour un poste
- **Statut** : en cours, accepté, refusé, en attente d'entretien, etc.
- **Décision** : note ou commentaire du responsable RH ou technique

Relations:

- Un **candidat** peut avoir plusieurs **candidatures** (stage, emploi, relance...)
- Une **candidature** est liée à un **candidat** et à un **statut**

- **Fonctionnalités sous forme de Web Services**

Requêtes (Queries)

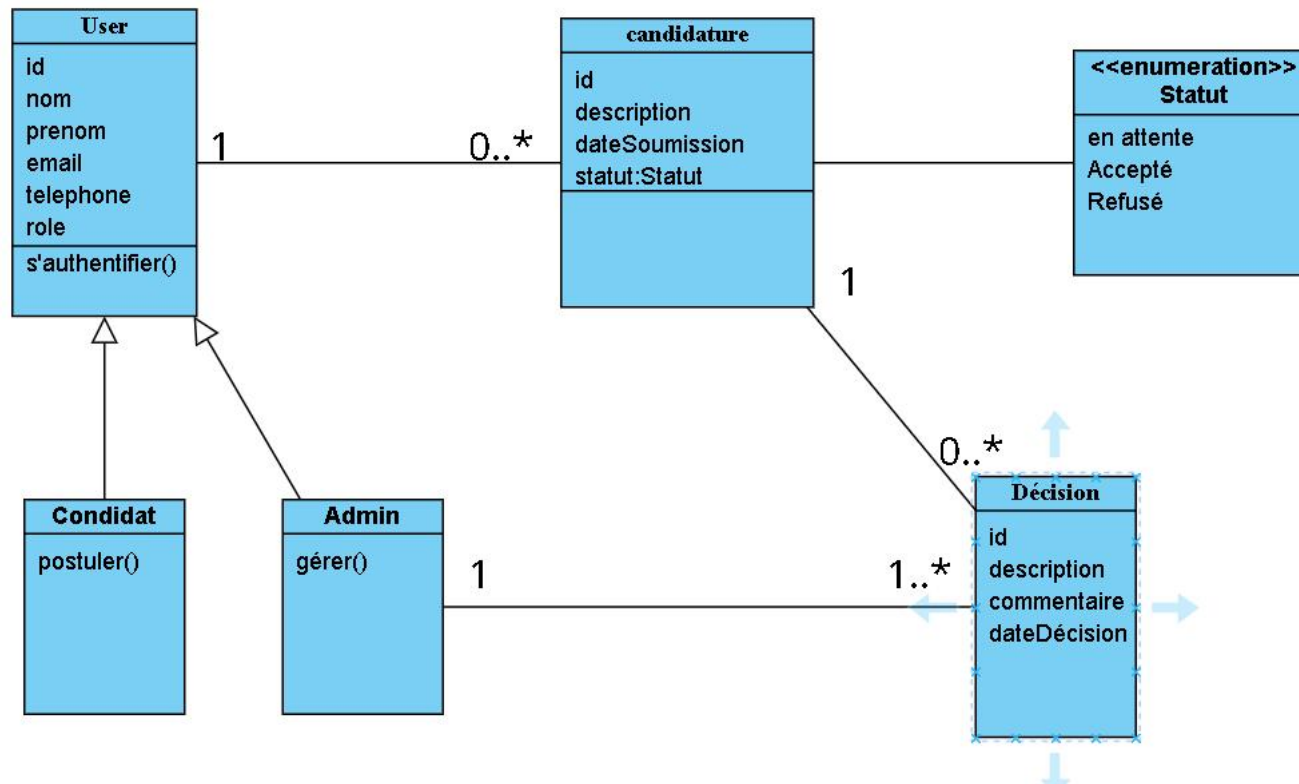
- ✓ **Candidatures()** : lister toutes les candidatures
- ✓ **candidatureById(id: ID!)** : consulter une candidature spécifique
- ✓ **candidaturesParStatut(statut: String)** : filtrer les candidatures par statut

Mutations

- ✓ **addCandidature(nom, email, statut)** : ajouter une nouvelle candidature
- ✓ **updateCandidature(id, statut, commentaire)** : mettre à jour le statut ou les commentaires
- ✓ **deleteCandidature(id)** : supprimer une candidature
- ✓ **deleteAllCandidatures** : supprimer toutes les candidatures

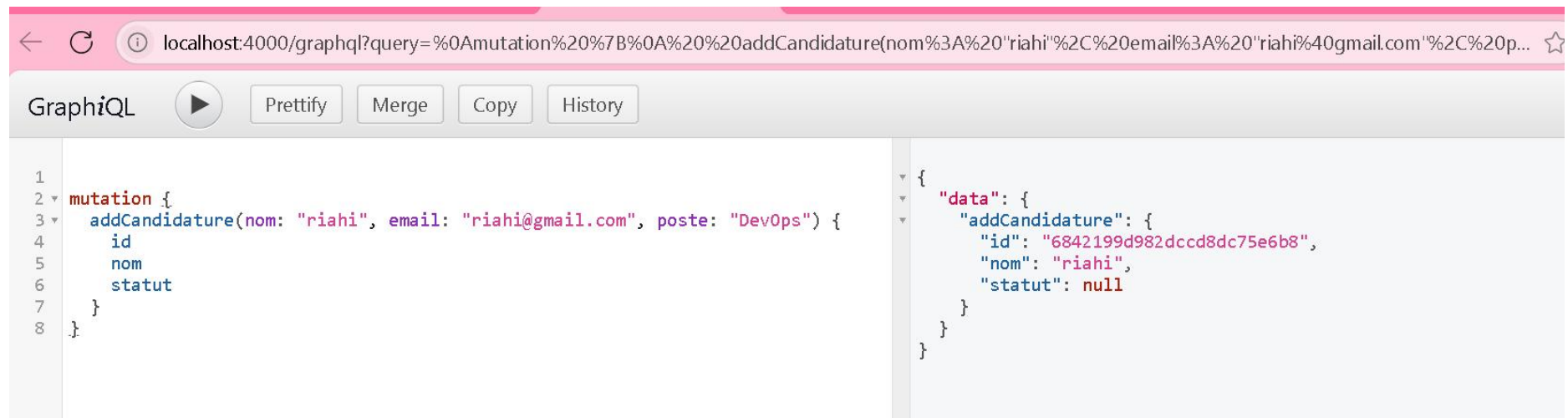
2. Diagrams

- Diagramme de classe:



• Schema des requetes && reponses du service

Ajouter une candidature:



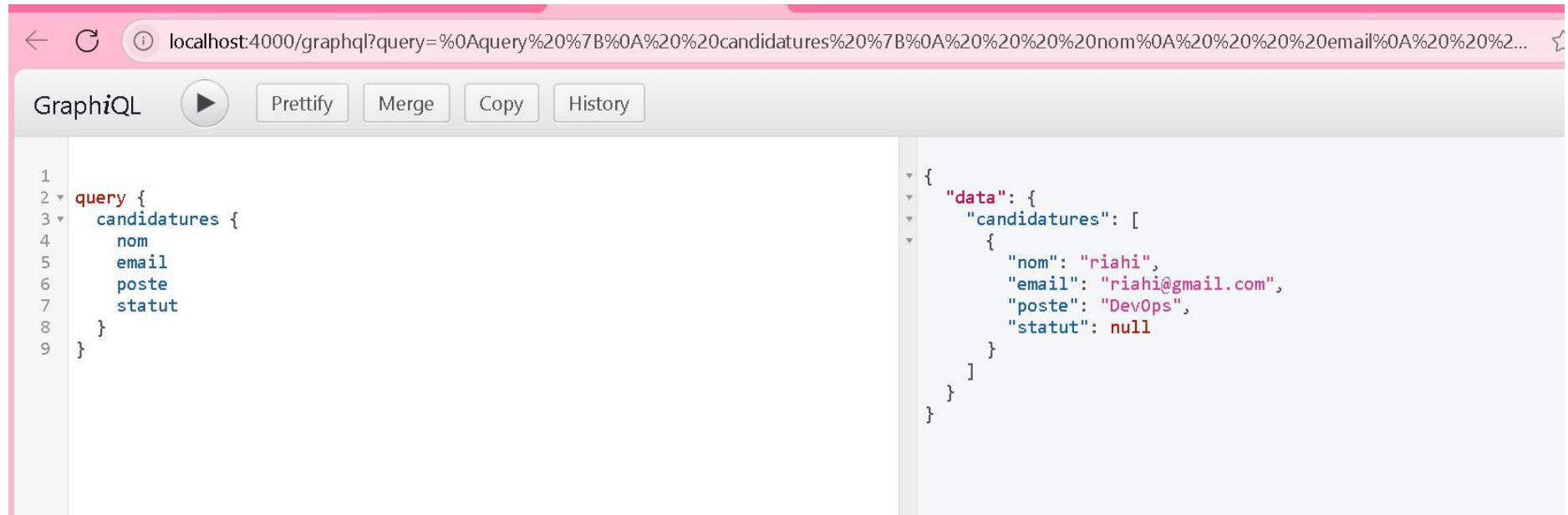
The screenshot shows the GraphQL Playground interface. The URL bar at the top displays the endpoint `localhost:4000/graphql?query=%0Amutation%20%7B%0A%20%20addCandidature(nom%3A%20%22riahi%22%2C%20email%3A%20%22riahi%40gmail.com%22%2C%20p...`. The interface includes a toolbar with buttons for 'Prettify', 'Merge', 'Copy', and 'History'. The left pane contains the following GraphQL query:

```
1
2 mutation {
3   addCandidature(nom: "riahi", email: "riahi@gmail.com", poste: "DevOps") {
4     id
5     nom
6     statut
7   }
8 }
```

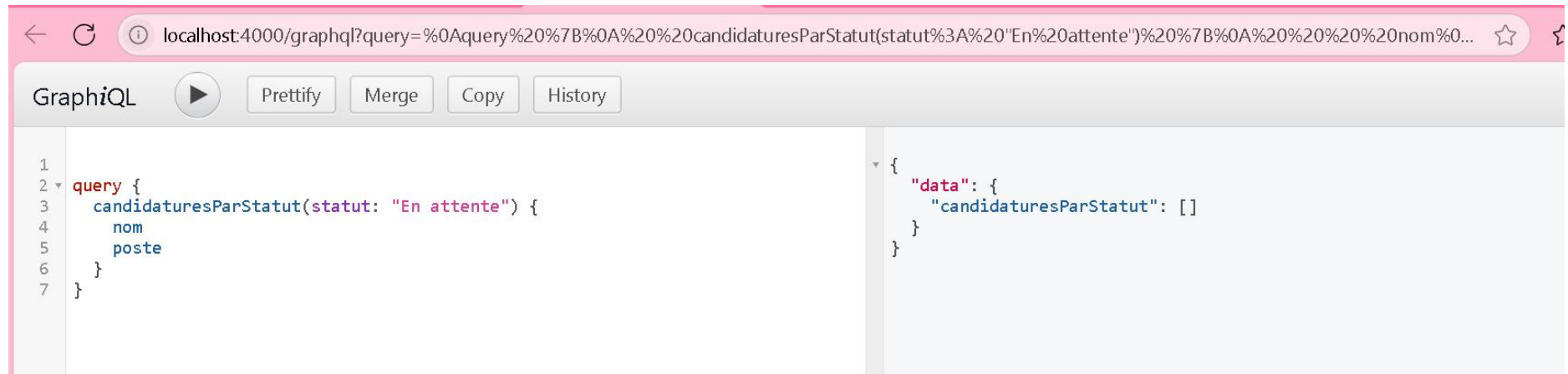
The right pane displays the JSON response:

```
{
  "data": {
    "addCandidature": {
      "id": "6842199d982dccd8dc75e6b8",
      "nom": "riahi",
      "statut": null
    }
  }
}
```

Lister tous les candidatures:



Filtrer par statut:



The screenshot shows the GraphQL Playground interface. The browser address bar displays the URL: `localhost:4000/graphql?query=%0Aquery%20%7B%0A%20%20candidaturesParStatut(statut%3A%20%22En%20attente%22)%20%7B%0A%20%20%20%20nom%0...`. The interface includes a toolbar with buttons for "Prettify", "Merge", "Copy", and "History".

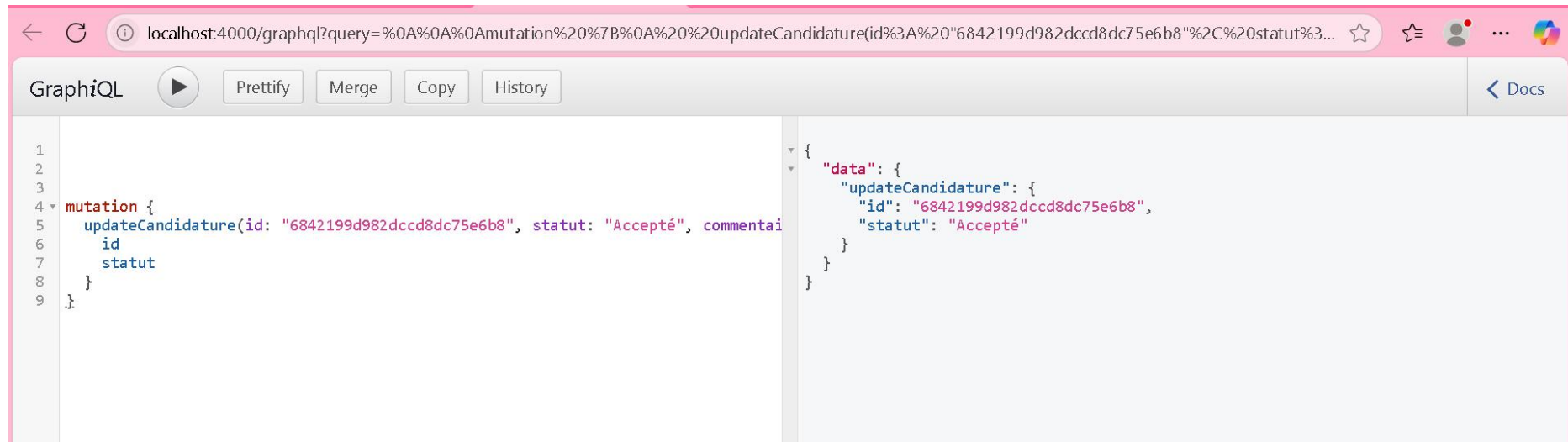
The query editor on the left contains the following query:

```
1
2 query {
3   candidaturesParStatut(statut: "En attente") {
4     nom
5     poste
6   }
7 }
```

The result pane on the right displays the JSON response:

```
{
  "data": {
    "candidaturesParStatut": []
  }
}
```

Modifier une candidature:



Supprimer une candidature:



**3. Implementation && documentation : voir ce lien
github :https://github.com/Arbia-riahi/projet-web-services_graphql.git**