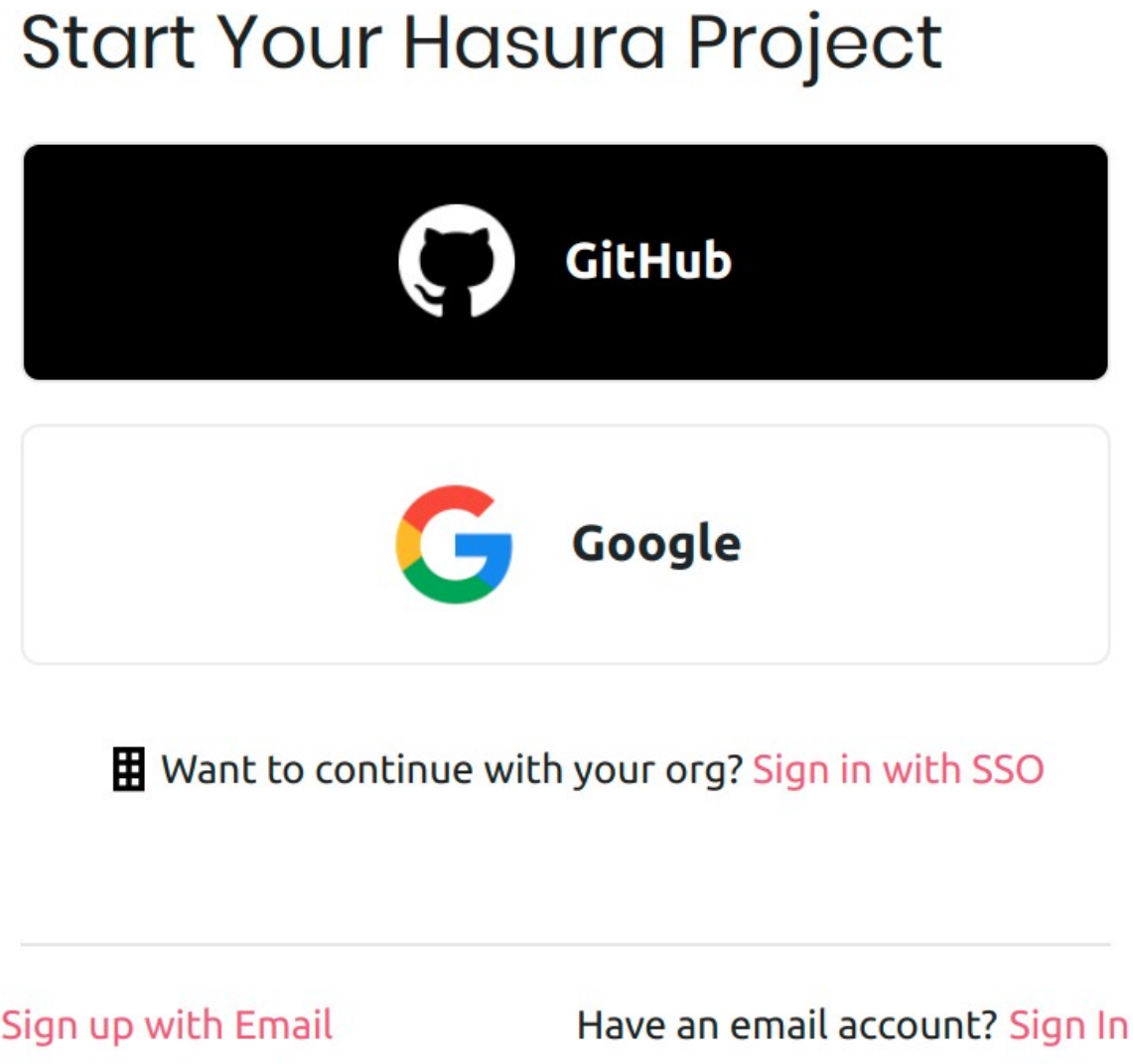
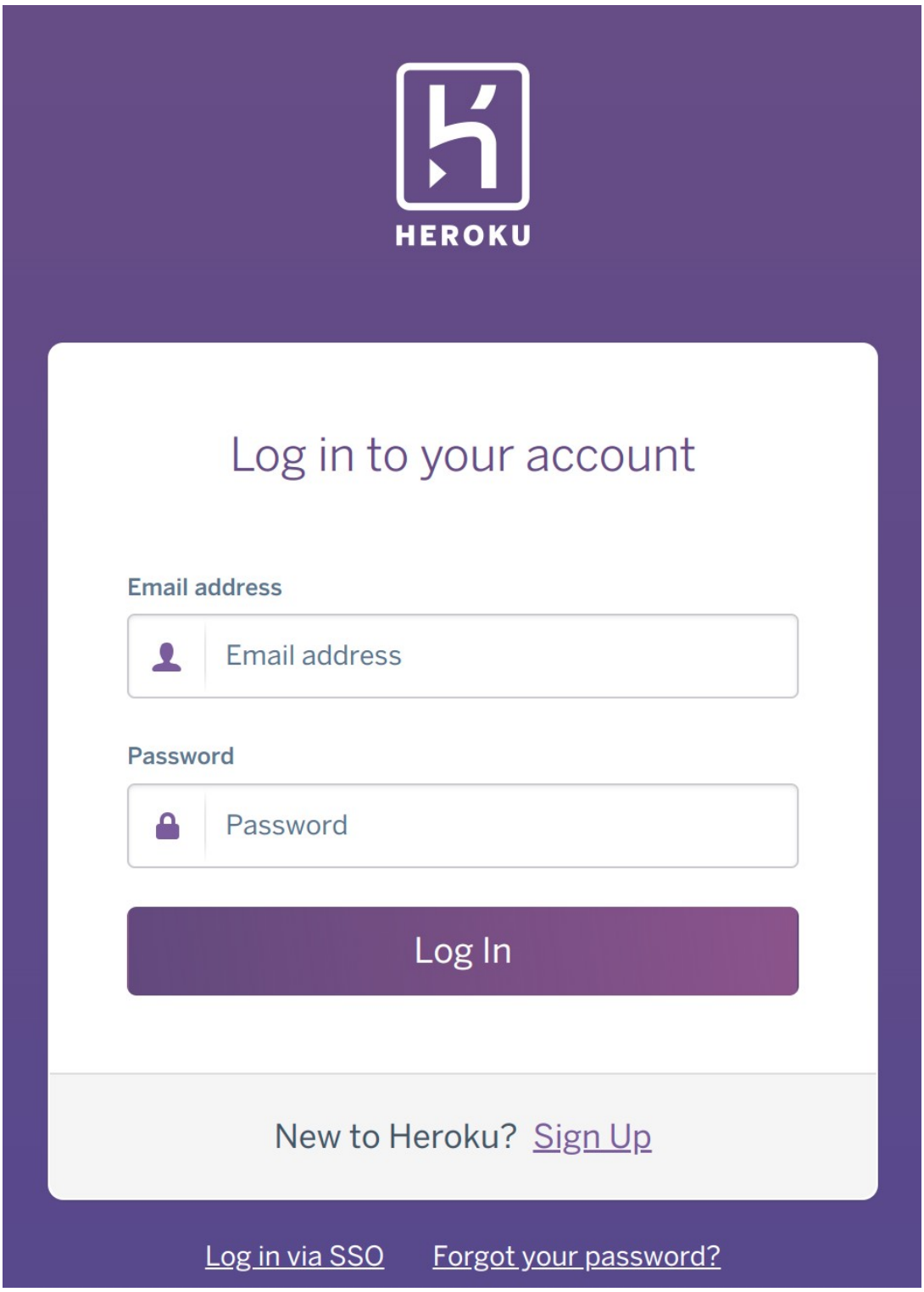
Hasura Training

Figure 1: https://tinyurl.com/mvf2ecee

What We Will Need

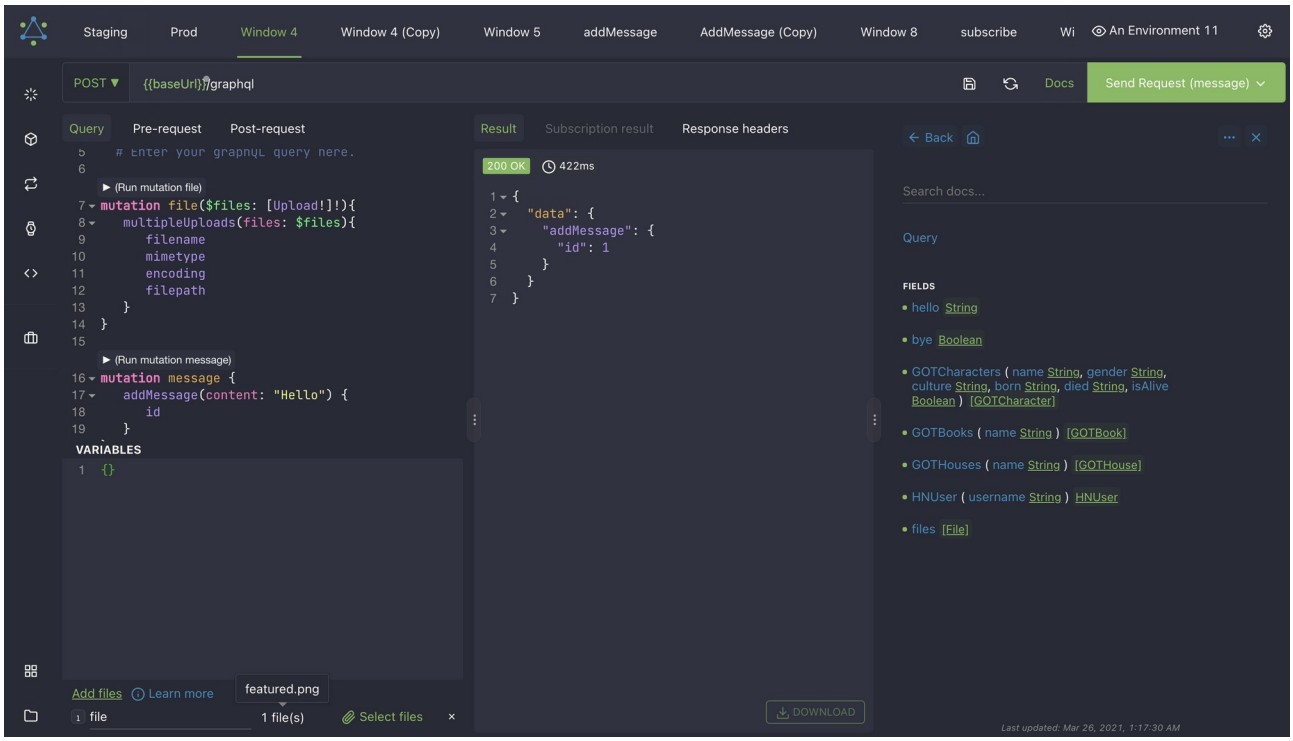


http://cloud.hasura.io



https://www.heroku.com/

What We Will Need (Optional)



https://altairgraphql.dev/

What We Will Cover

- What is GraphQL?
- What is Hasura?
- How do we use them?
- How do we use them well?
- What are the general principles?
- What are the particular details?
- What is the proper mental model?
- How do we get there?

What is GraphQL?

- Query Language
- Schema
  - Types
    - Fields
- Types relate to Types
- Sound familiar?

```
query MyQuery {
  user {
    id
    name
    email
    addresses {
      id
      name
      address_line_one
      address_line_two
      city
    }
  }
}
```


```
{
  "data": {
    "user": [
      {
        "id": 4,
        "name": "Dickie",
        "email": "dmayell2@yellowbook.com",
        "addresses": [
          {
            "id": 1,
            "name": null,
            "address_line_one": "119 Victoria Trail",
            "address_line_two": null,
            "city": "Springfield"
          }
        ]
      },
      {
        "id": 5,
        "name": "Korella",
        "email": "kfriedlos3@comsenz.com",
        "addresses": [
          {
            "id": 2,
            "name": null,
            "address_line_one": "06 Holy Cross Lane",
            "address_line_two": null,
            "city": "Indianapolis"
          }
        ]
      },
```


What is Hasura?

- API Server
- Development Environment

How do we use them?

- Build network applications ***faster***, more ***reliably***, and ***cheaper*** with GraphQL.
- Build GraphQL applications ***faster***, more ***reliably***, and ***cheaper*** with Hasura.
- Apply the general principles with training from ***Hasura Developer Relations***.
- Apply the particular details with support from ***Hasura Developer Success***.

How do we use them well?

- Adopt a principled approach.
- Develop a proper mental model.

What are the general principles?

1. Build for *humans*.
2. Build with *personas*.
3. Build *in* the database.
4. Build a *data model*.
5. *Remember* OLTP vs OLAP.

What is the proper mental model?

- Hasura `graphql-engine` is the API server, is on the data plane, and is on the critical path.
- Hasura `console` is part of the development environment, is on the control plane, and is off the critical path.
- Hasura `CLI` is part of the development environment, is on the control plane, and is far off the critical path.

What is the proper mental model?

- The key *functional* concerns are: data modeling, API development, security, and integrations.
- The key *non-functional* concerns are: deployment, caching, monitoring.

How do we get there?

- Track 1: Development
- Track 2: Deployment

Track 1: Development

- Setup
- Modeling
- Security
- Actions
- Events
- APIs
- Remote Joins
- Remote Schema

Track 2: Deployment

- Self-Hosted vs Cloud
- Development Environment
- CI/CD