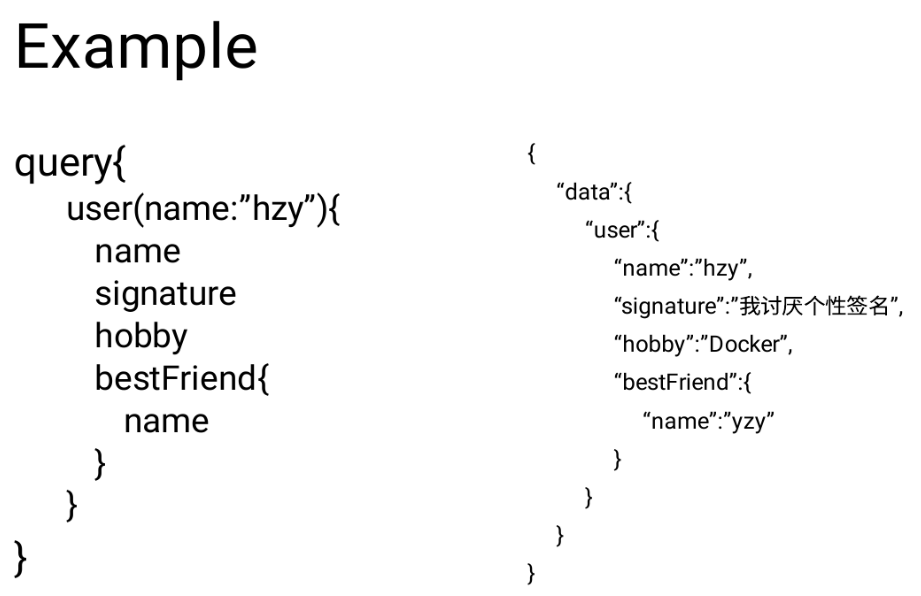
# 浅谈Apollo

## 前言：什么是GraphQL

对于前端来说，GraphQL是一个很好的减轻工作量的工具。

在规定的schema下，一次查询所需的所有的内容。



Example

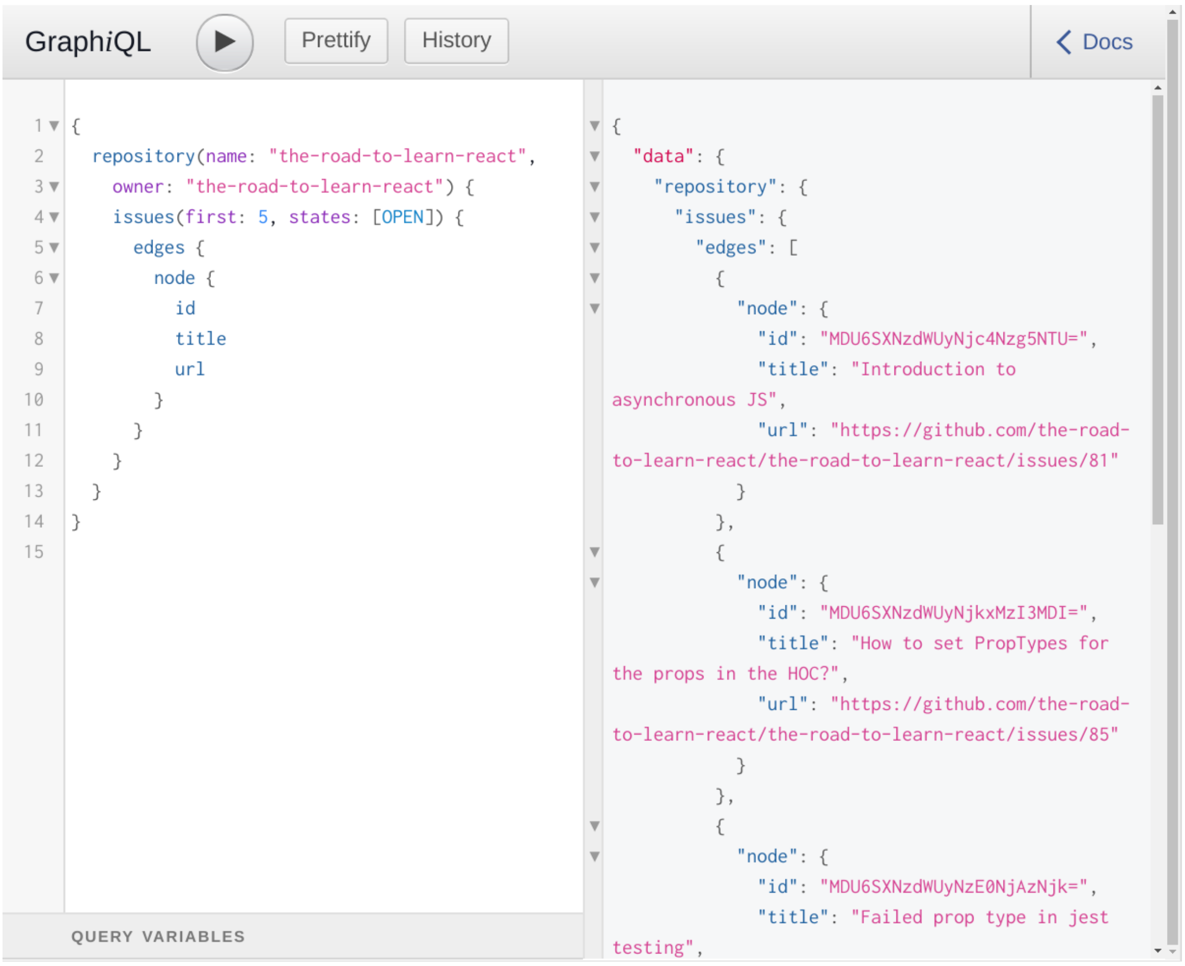
```
query{
    user(name:”hzy”){
      name
      signature
      hobby
      bestFriend{
        name
      }
    }
}
```

```
{
    “data”:{
        “user”:{
            “name”:”hzy”,
            “signature”:”我讨厌个性签名”,
            “hobby”:”Docker”,
            “bestFriend”:{
                “name”:”yzy”
            }
        }
    }
}
```

关于GraphQL的详情以及在后端的应用可以查看兔兔的《从GraphQL到Prisma》。

## 使用GraphQL与Github GQLAPI进行交互

1. 查询数据，接收返回值



```
{
  repository(name: "the-road-to-learn-react",
    owner: "the-road-to-learn-react") {
    issues(first: 5, states: [OPEN]) {
      edges {
        node {
          id
          title
          url
        }
      }
    }
  }
}
```

```
{
  "data": {
    "repository": {
      "issues": {
        "edges": [
          {
            "node": {
              "id": "MDU6SXNzdWUyNjc4Nzg5NTU=",
              "title": "Introduction to
asynchronous JS",
              "url": "https://github.com/the-road-
to-learn-react/the-road-to-learn-react/issues/81"
            }
          },
          {
            "node": {
              "id": "MDU6SXNzdWUyNjkxMzI3MDI=",
              "title": "How to set PropTypes for
the props in the HOC?",
              "url": "https://github.com/the-road-
to-learn-react/the-road-to-learn-react/issues/85"
            }
          },
          {
            "node": {
              "id": "MDU6SXNzdWUyNzE0NjAzNjk=",
              "title": "Failed prop type in jest
testing",
```

对应的代码

```
const GET_ORGANIZATION = `
  {
    organization(login: "the-road-to-learn-react") {
      name
      url
      ...
    }
  }
`;
const axiosGitHubGraphQL = axios.create({
  baseURL: 'https://api.github.com/graphql',
  ...
});
```

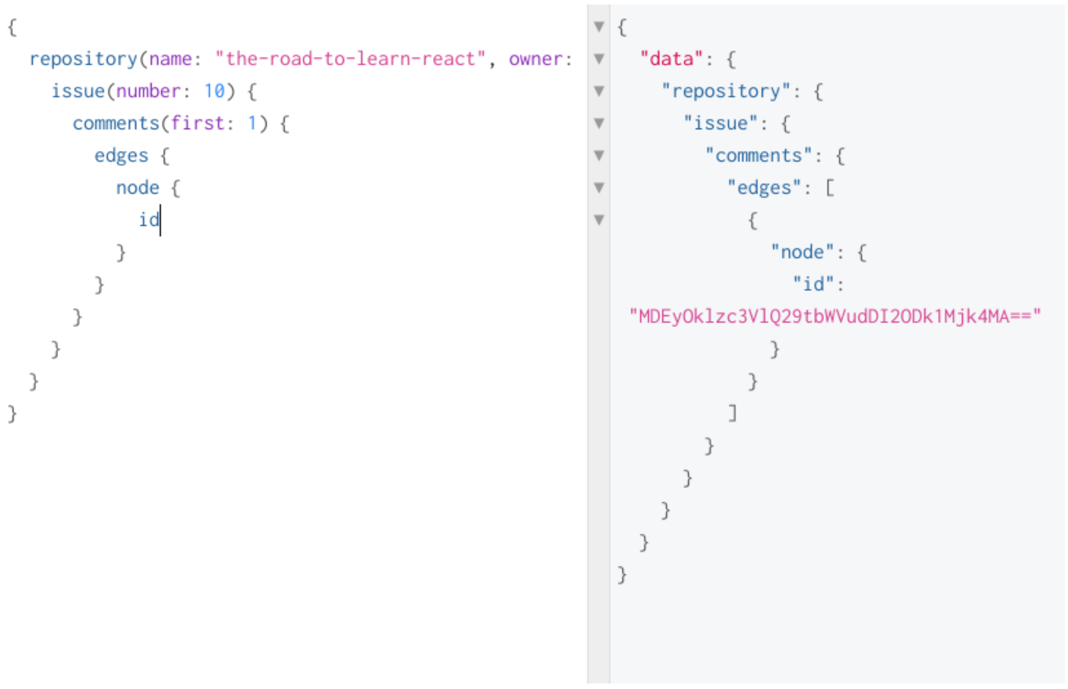
2. 处理数据，修改状态

```
class App extends Component {
     state = {
       organization: null,
       errors: null,
     };
     ...
     onFetchFromGitHub = () => {
       axiosGitHubGraphQL
         .post('', { query: GET_ORGANIZATION })
         .then(result =>
            this.setState(() => ({
              organization: result.data.data.organization,
              errors: result.data.errors,
            })),
          );
      }
      ...
    }
```

Or use Redux

```
function* handleFetch() { //use redux-saga
  const res = yield call(axiosGitHubGraphQL.post, '',
                         { query: GET_ORGANIZATION })
  yield put(action("FETCH_SUCCESS", data))
}
const reducer(state, action) => {
  switch (action.type) {
    case "FETCH_REQUEST": {
      return { ...state, loading: true }
    }
    case "FETCH_SUCCESS": {
      return { ...state, loading: false, data: action.payload }
     ...
  }
}
```

3. 从state中读取数据，施工我们的视图层
4. 一些恼人的细节
    1. state高度嵌套，查询麻烦



```
{
  repository(name: "the-road-to-learn-react", owner:
    issue(number: 10) {
      comments(first: 1) {
        edges {
          node {
            id
          }
        }
      }
    }
  }
}
```

```
{
  "data": {
    "repository": {
      "issue": {
        "comments": {
          "edges": [
            {
              "node": {
                "id":
"MDEyOklzc3VlQ29tbWVudDI2ODk1Mjk4MA=="
              }
            }
          ]
        }
      }
    }
  }
}
```

    2. 缓存信息需要重复搬砖
    3. 不能抛开视图层专注处理数据
    4. 本地UI进行积极更新

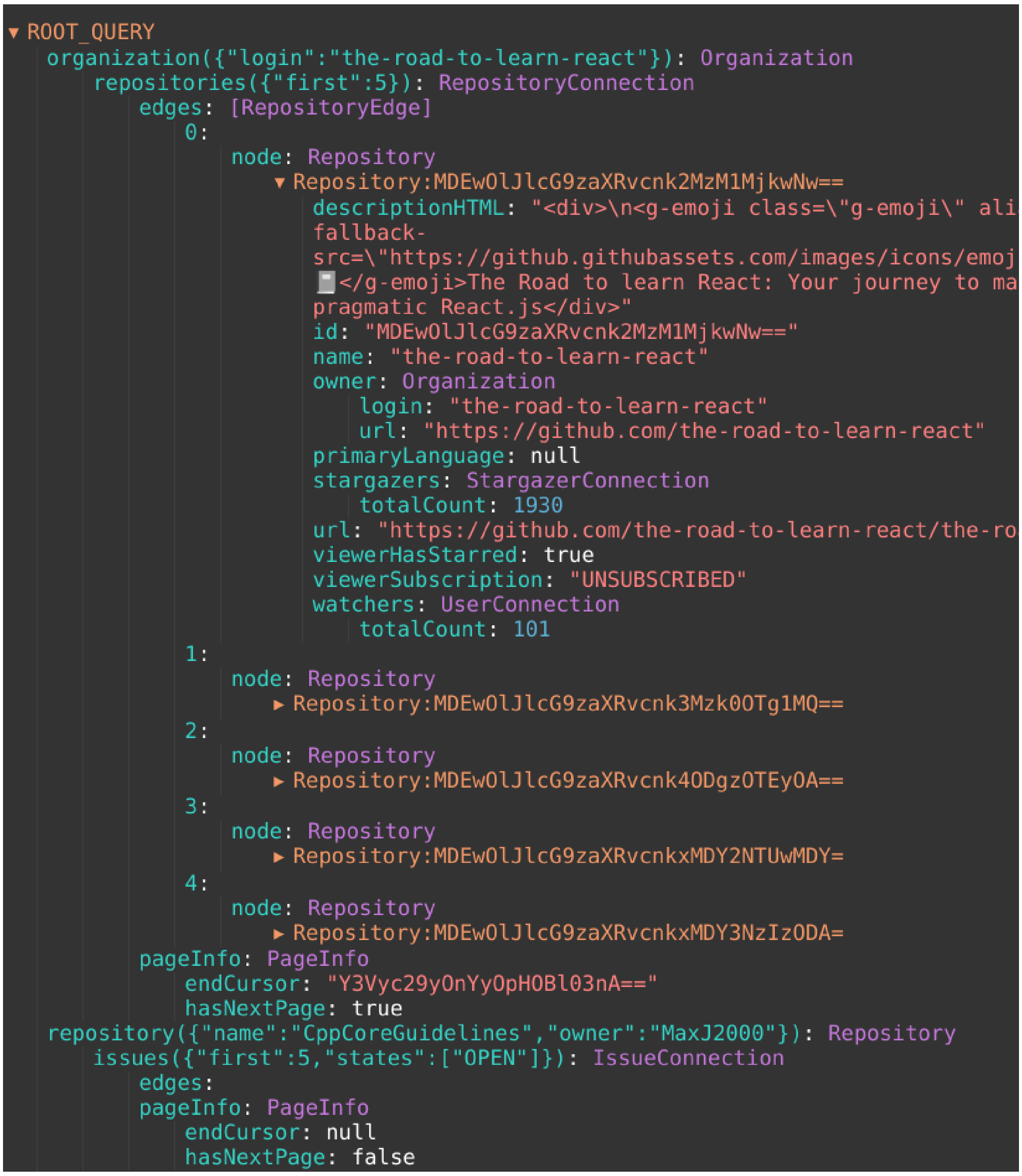
1 Unwatch  1 Unstar

## Apollo一把梭

### Apollo帮我们做了什么

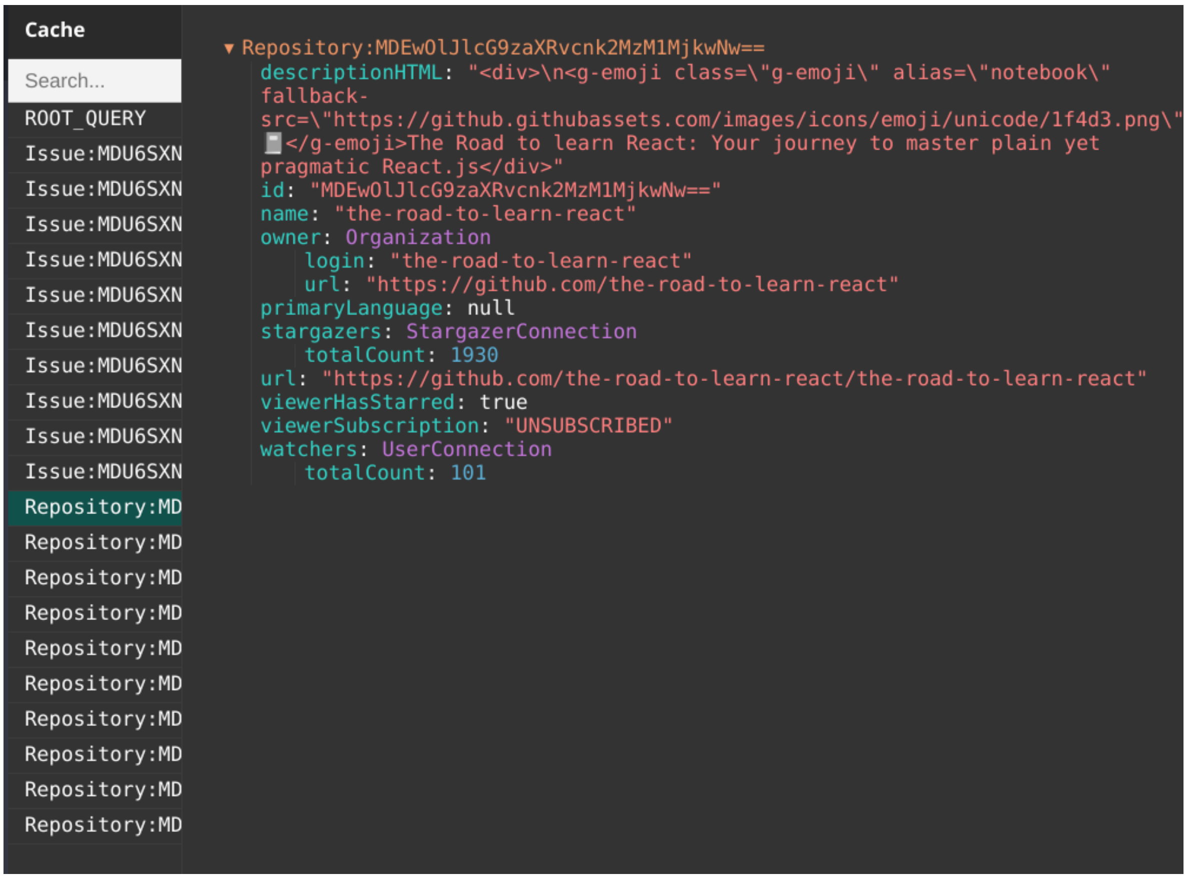
1. 缓存数据，提升性能
2. normalize

From



```
▼ ROOT_QUERY
    organization({"login":"the-road-to-learn-react"}): Organization
        repositories({"first":5}): RepositoryConnection
            edges: [RepositoryEdge]
                0:
                    node: Repository
                        ▼ Repository:MDEwOlJlcG9zaXRvcnk2MzM1MjkwNw==
                            descriptionHTML: "<div>\n<g-emoji class=\"g-emoji\" ali
                            fallback-
                            src=\"https://github.githubassets.com/images/icons/emoj
                            </g-emoji>The Road to learn React: Your journey to ma
                            pragmatic React.js</div>"
                            id: "MDEwOlJlcG9zaXRvcnk2MzM1MjkwNw=="
                            name: "the-road-to-learn-react"
                            owner: Organization
                                login: "the-road-to-learn-react"
                                url: "https://github.com/the-road-to-learn-react"
                            primaryLanguage: null
                            stargazers: StargazerConnection
                                totalCount: 1930
                            url: "https://github.com/the-road-to-learn-react/the-ro
                            viewerHasStarred: true
                            viewerSubscription: "UNSUBSCRIBED"
                            watchers: UserConnection
                                totalCount: 101
                1:
                    node: Repository
                        ▶ Repository:MDEwOlJlcG9zaXRvcnk3Mzk0OTg1MQ==
                2:
                    node: Repository
                        ▶ Repository:MDEwOlJlcG9zaXRvcnk4ODgzOTEyOA==
                3:
                    node: Repository
                        ▶ Repository:MDEwOlJlcG9zaXRvcnkxMDY2NTUwMDY=
                4:
                    node: Repository
                        ▶ Repository:MDEwOlJlcG9zaXRvcnkxMDY3NzIzODA=
            pageInfo: PageInfo
                endCursor: "Y3Vyc29yOnYyOpHOBl03nA=="
                hasNextPage: true
    repository({"name":"CppCoreGuidelines","owner":"MaxJ2000"}): Repository
        issues({"first":5,"states":["OPEN"]}): IssueConnection
            edges:
            pageInfo: PageInfo
                endCursor: null
                hasNextPage: false
```

To



```
ROOT_QUERY
Issue:MDU6SXN
Issue:MDU6SXN
Issue:MDU6SXN
Issue:MDU6SXN
Issue:MDU6SXN
Issue:MDU6SXN
Issue:MDU6SXN
Issue:MDU6SXN
Issue:MDU6SXN
Issue:MDU6SXN
Repository:MD
Repository:MD
Repository:MD
Repository:MD
Repository:MD
Repository:MD
Repository:MD
Repository:MD
Repository:MD
Repository:MD
```

```
▾ Repository:MDEwOlJlcG9zaXRvcnk2MzM1MjkwNw==
   descriptionHTML: "<div>\n<g-emoji class=\"g-emoji\" alias=\"notebook\"
   fallback-
   src=\"https://github.githubassets.com/images/icons/emoji/unicode/1f4d3.png\"
   📓</g-emoji>The Road to learn React: Your journey to master plain yet
   pragmatic React.js</div>"
   id: "MDEwOlJlcG9zaXRvcnk2MzM1MjkwNw=="
   name: "the-road-to-learn-react"
   owner: Organization
       login: "the-road-to-learn-react"
       url: "https://github.com/the-road-to-learn-react"
   primaryLanguage: null
   stargazers: StargazerConnection
       totalCount: 1930
   url: "https://github.com/the-road-to-learn-react/the-road-to-learn-react"
   viewerHasStarred: true
   viewerSubscription: "UNSUBSCRIBED"
   watchers: UserConnection
       totalCount: 101
```

3. 较为便捷的实现积极UI

## 代码实现

1. Apollo-Link

```
import { ApolloClient } from 'apollo-client';
import { HttpLink } from 'apollo-link-http';
import { InMemoryCache } from 'apollo-cache-inmemory';

const httpLink = new HttpLink({
  uri: https://api.github.com/graphql
});
const cache = new InMemoryCache();
const client = new ApolloClient({
  link: httpLink,
  cache,
});
```

2. Apollo-react

```
import { ApolloProvider } from 'react-apollo';

ReactDOM.render(
  <ApolloProvider client={client}>
    <App />
  </ApolloProvider>,
  document.getElementById('root')
);
```

3. Query

```
import gql from 'graphql-tag';
import { Query } from 'react-apollo';

const GET_CURRENT_USER = gql`
  {
    viewer {
      login
      name
    }
  }
`;

const Profile = () => (
  <Query query={GET_CURRENT_USER}>
    {({ data }) => {
      const { viewer } = data;
      return (
        <div>
          {viewer.name} {viewer.login}
        </div>
      );
    }}
  </Query>
);
```

4. Mutation

```
import { Mutation } from 'react-apollo';

const STAR_REPOSITORY = gql`
  mutation($id: ID!) {
    addStar(input: { starrableId: $id }) {
      starrable {
        id
        viewerHasStarred
      }
    }
  }
`;

<Mutation mutation={STAR_REPOSITORY} variables={{ id }}>
          {(addStar) => (
            <Button
              className={'RepositoryItem-title-action'}
              onClick={addStar}
            >
              {stargazers.totalCount} Star
            </Button>
          )}
</Mutation>
```

5. Local State

```
const updateAddStar = (client, mutationResult) => {
  ...
};

<Mutation mutation={STAR_REPOSITORY} variables={{ id }}
          update={updateAddStar}>
            ...
</Mutation>
```

实际上的updateAddStar

```
const REPOSITORY_FRAGMENT = gql`
  fragment repository on Repository {
    id
    viewerHasStarred
    stargazers {
      totalCount
    }
  }
`;

const updateAddStar = (
  client,
  { data: { addStar: { starrable: { id } } } },
) => {
  const repository = client.readFragment({
    id: `Repository:${id}`,
    fragment: REPOSITORY_FRAGMENT,
  });
  const totalCount = repository.stargazers.totalCount + 1;
  client.writeFragment({
    id: `Repository:${id}`,
    fragment: REPOSITORY_FRAGMENT,
    data: {
      ...repository,
      stargazers: {
        ...repository.stargazers,
        totalCount,
      },
    },
  });
};
```

6. Optimistic UI

```
<Mutation mutation={STAR_REPOSITORY} variables={{ id }}  update={updateAddStar}
    optimisticResponse={{
       updateSubscription: {
           __typename: 'Mutation',
           subscribable: {
             __typename: 'Repository',
             id,
             viewerHasStarred: !viewerHasStarred,
          },
       },
}}>
```

```
    {(addStar, { data, loading, error }) => (
              <Button
                className={'RepositoryItem-title-action'}
                onClick={addStar}
              >
                {stargazers.totalCount} Star
              </Button>
            )}
</Mutation>
```