The year of GraphQL

Andrei Calazans

@Andrei_Calazans

@AndreiCalazans

ENTRIA

THE TECH TEAM

GraphQL ?

Steven Blum Aug 8, 2012

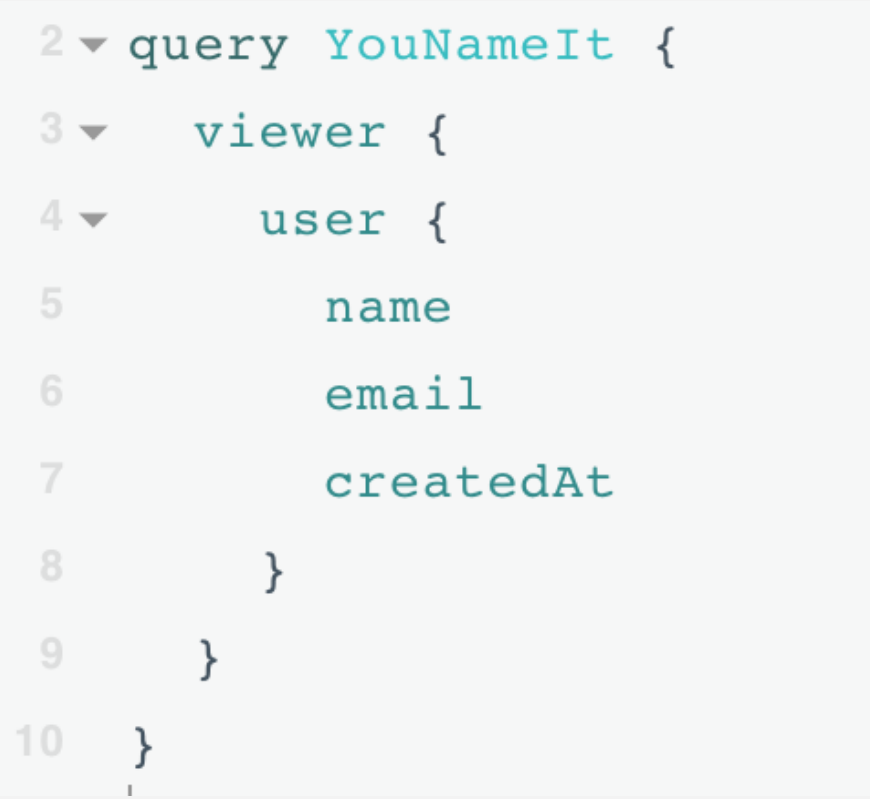
Why Is Facebook's App Such a Piece of Crap?

Desenvolvida em 2012

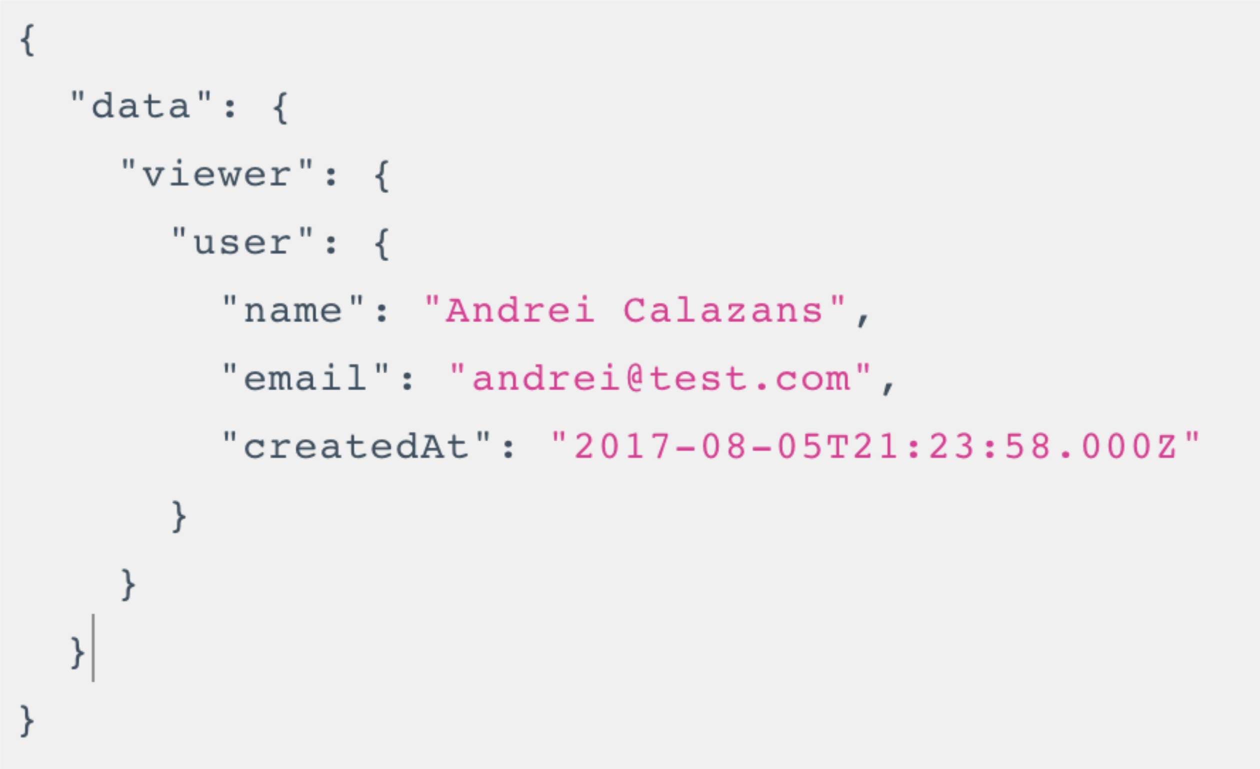
Open source version published in july 2015

Query

Mutations

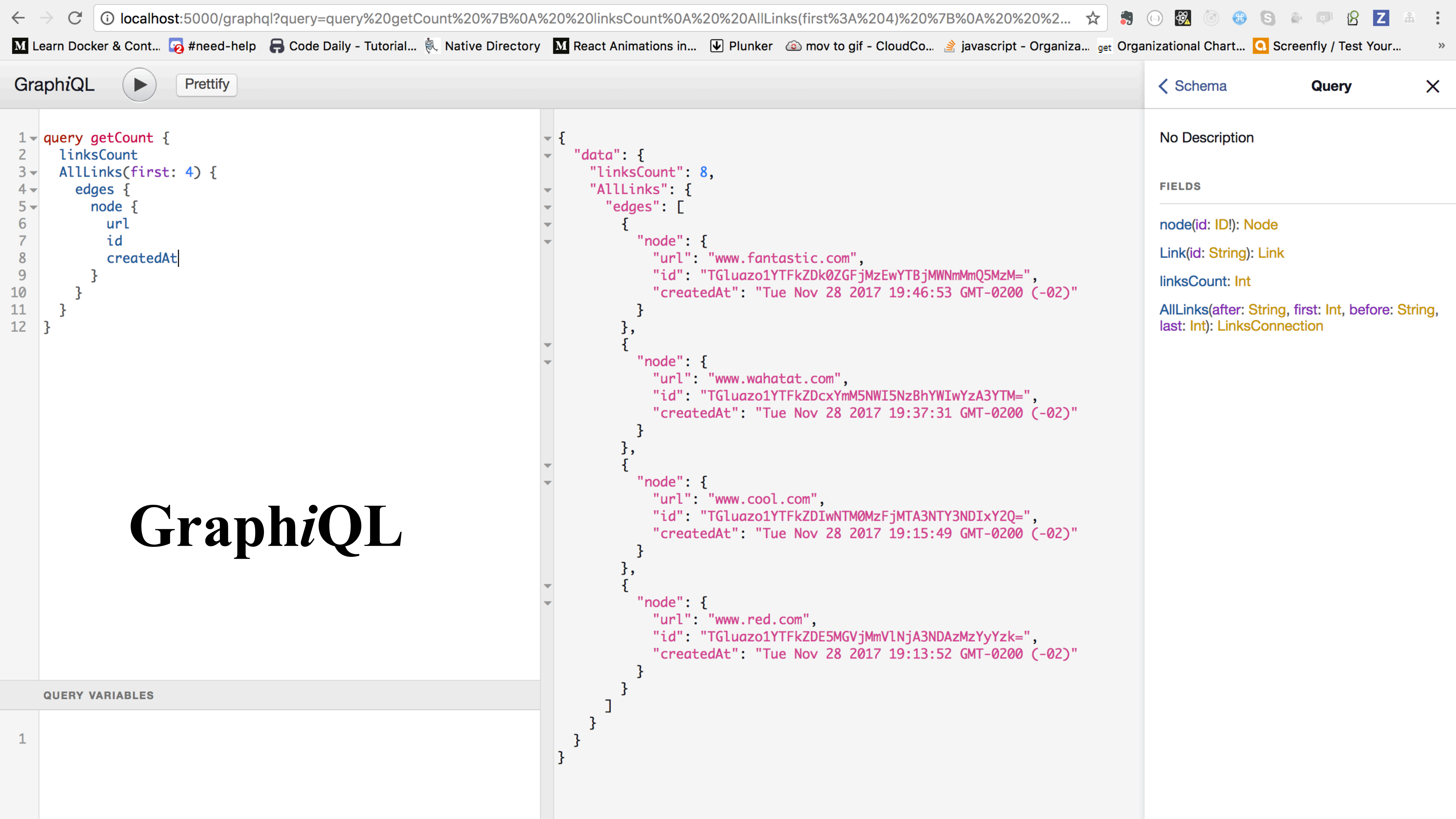


```graphql
query YouNameIt {
  viewer {
    user {
      name
      email
      createdAt
    }
  }
}
```



```json
{
  "data": {
    "viewer": {
      "user": {
        "name": "Andrei Calazans",
        "email": "andrei@test.com",
        "createdAt": "2017-08-05T21:23:58.000Z"
      }
    }
  }
}
```

GraphiQL



```
query getCount {
  linksCount
  AllLinks(first: 4) {
    edges {
      node {
        url
        id
        createdAt
      }
    }
  }
}
```

```
{
  "data": {
    "linksCount": 8,
    "AllLinks": {
      "edges": [
        {
          "node": {
            "url": "www.fantastic.com",
            "id": "TGluazo1YTFkZDk0ZGFjMzEwYTBjMWNmMmQ5MzM=",
            "createdAt": "Tue Nov 28 2017 19:46:53 GMT-0200 (-02)"
          }
        },
        {
          "node": {
            "url": "www.wahatat.com",
            "id": "TGluazo1YTFkZDcxYmM5NWI5NzBhYWIwYzA3YTM=",
            "createdAt": "Tue Nov 28 2017 19:37:31 GMT-0200 (-02)"
          }
        },
        {
          "node": {
            "url": "www.cool.com",
            "id": "TGluazo1YTFkZDIwNTM0MzFjMTA3NTY3NDIxY2Q=",
            "createdAt": "Tue Nov 28 2017 19:15:49 GMT-0200 (-02)"
          }
        },
        {
          "node": {
            "url": "www.red.com",
            "id": "TGluazo1YTFkZDE5MGVjMmVlNjA3NDAzMzYyYzk=",
            "createdAt": "Tue Nov 28 2017 19:13:52 GMT-0200 (-02)"
          }
        }
      ]
    }
  }
}
```

No Description

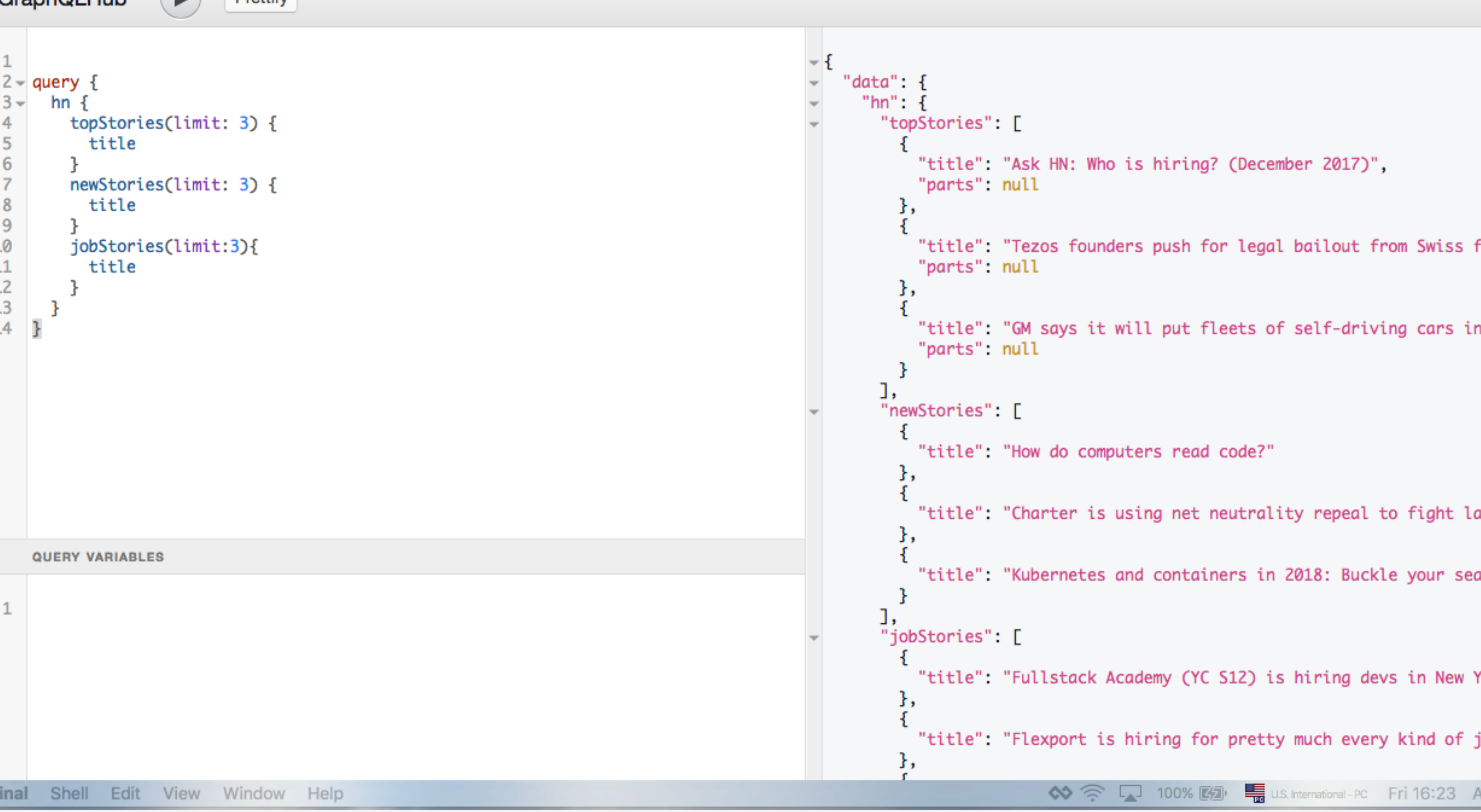
FIELDS

node(id: ID!): Node

Link(id: String): Link

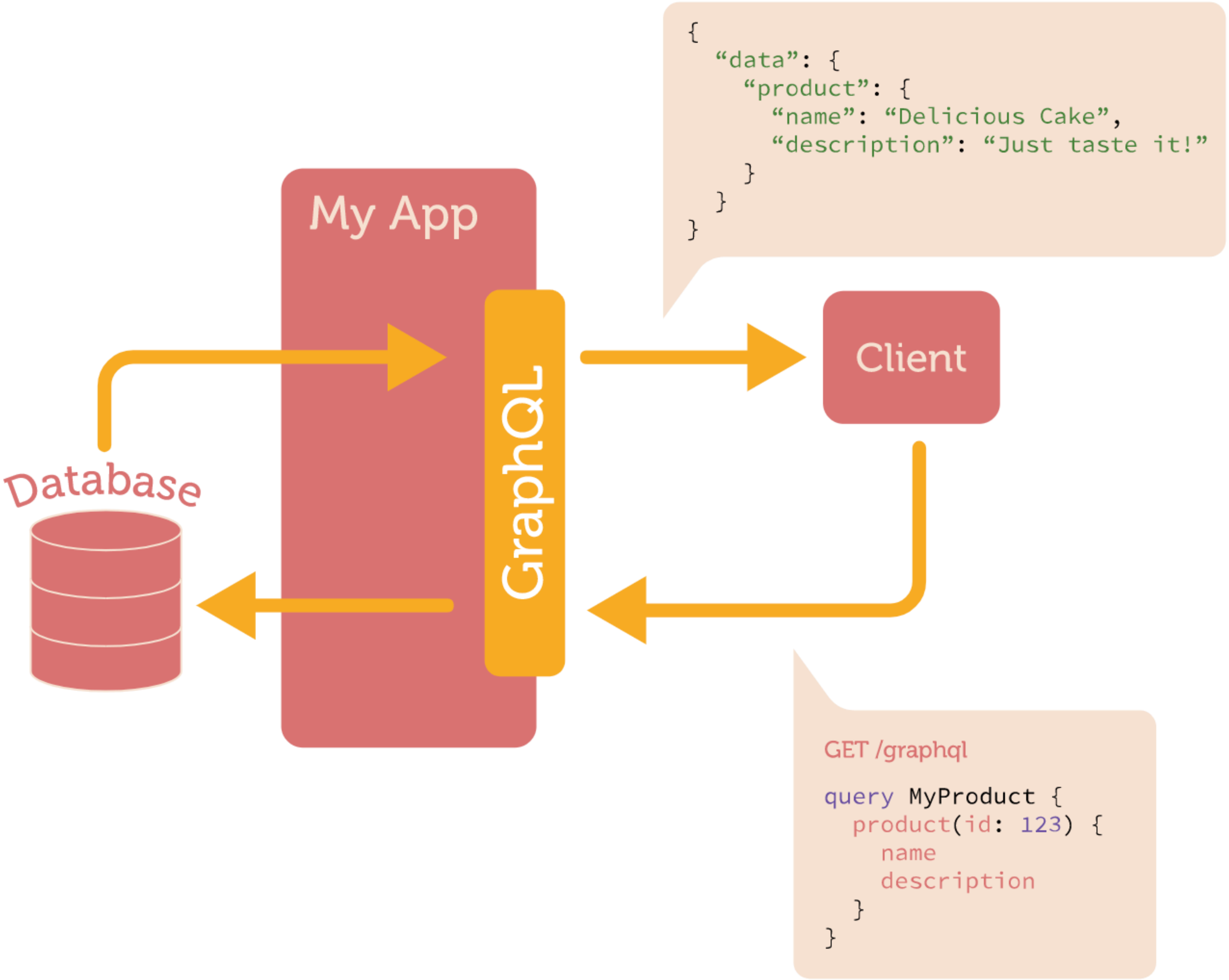
linksCount: Int

AllLinks(after: String, first: Int, before: String, last: Int): LinksConnection



```

query {
  hn {
    topStories(limit: 3) {
      title
    }
    newStories(limit: 3) {
      title
    }
    jobStories(limit:3){
      title
    }
  }
}
```

```
{
  "data": {
    "hn": {
      "topStories": [
        {
          "title": "Ask HN: Who is hiring? (December 2017)",
          "parts": null
        },
        {
          "title": "Tezos founders push for legal bailout from Swiss f
          "parts": null
        },
        {
          "title": "GM says it will put fleets of self-driving cars in
          "parts": null
        }
      ],
      "newStories": [
        {
          "title": "How do computers read code?"
        },
        {
          "title": "Charter is using net neutrality repeal to fight la
        },
        {
          "title": "Kubernetes and containers in 2018: Buckle your sea
        }
      ],
      "jobStories": [
        {
          "title": "Fullstack Academy (YC S12) is hiring devs in New Y
        },
        {
          "title": "Flexport is hiring for pretty much every kind of j
        },
```



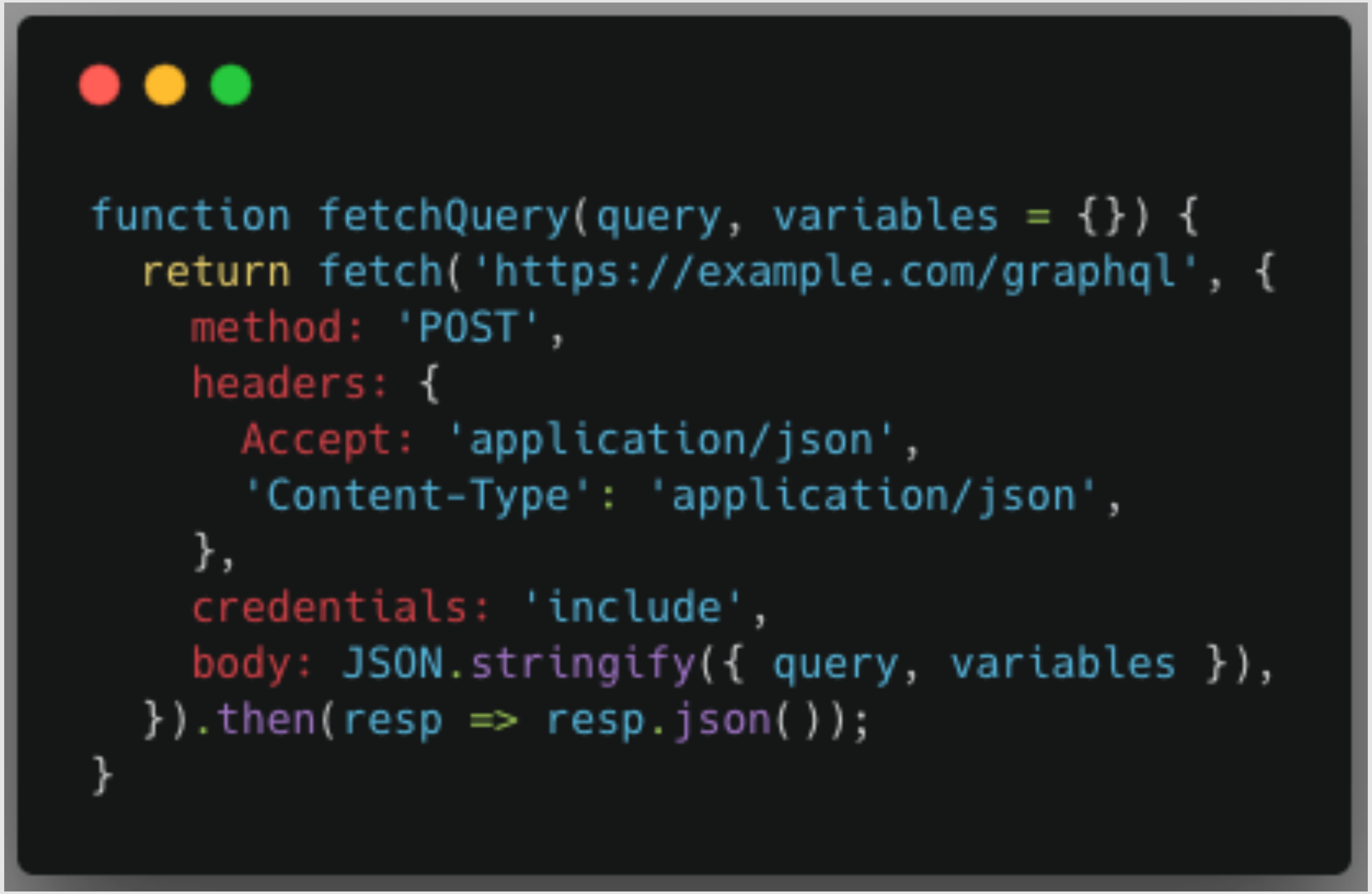
```
{
  “data”: {
    “product”: {
      “name”: “Delicious Cake”,
      “description”: “Just taste it!”
    }
  }
}
```

GET /graphql

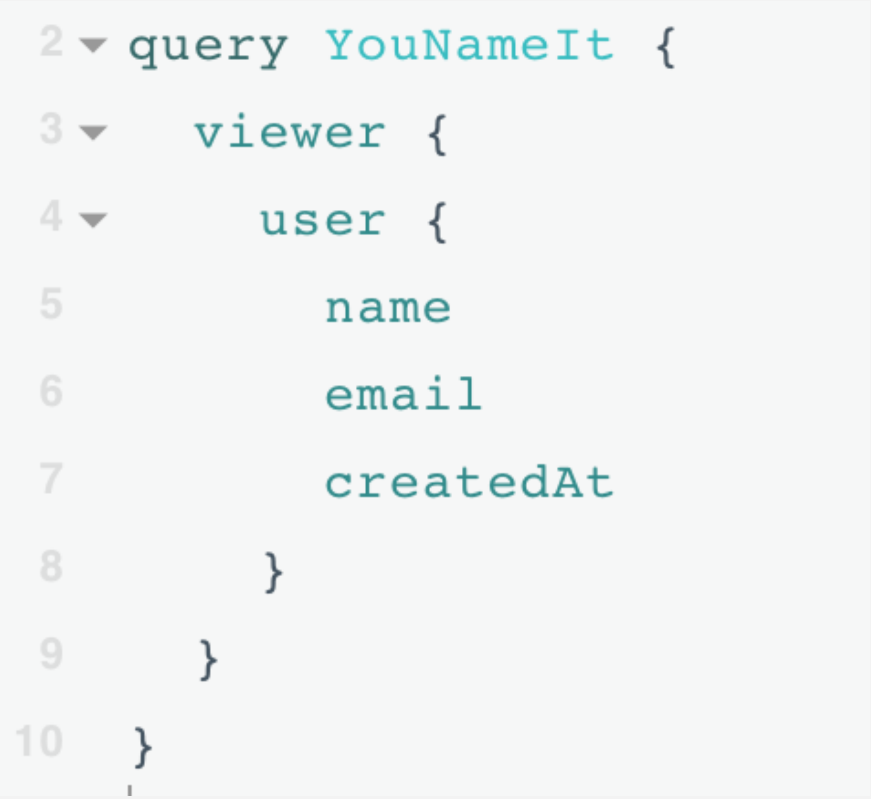
```
query MyProduct {
  product(id: 123) {
    name
    description
  }
}
```

Source: site

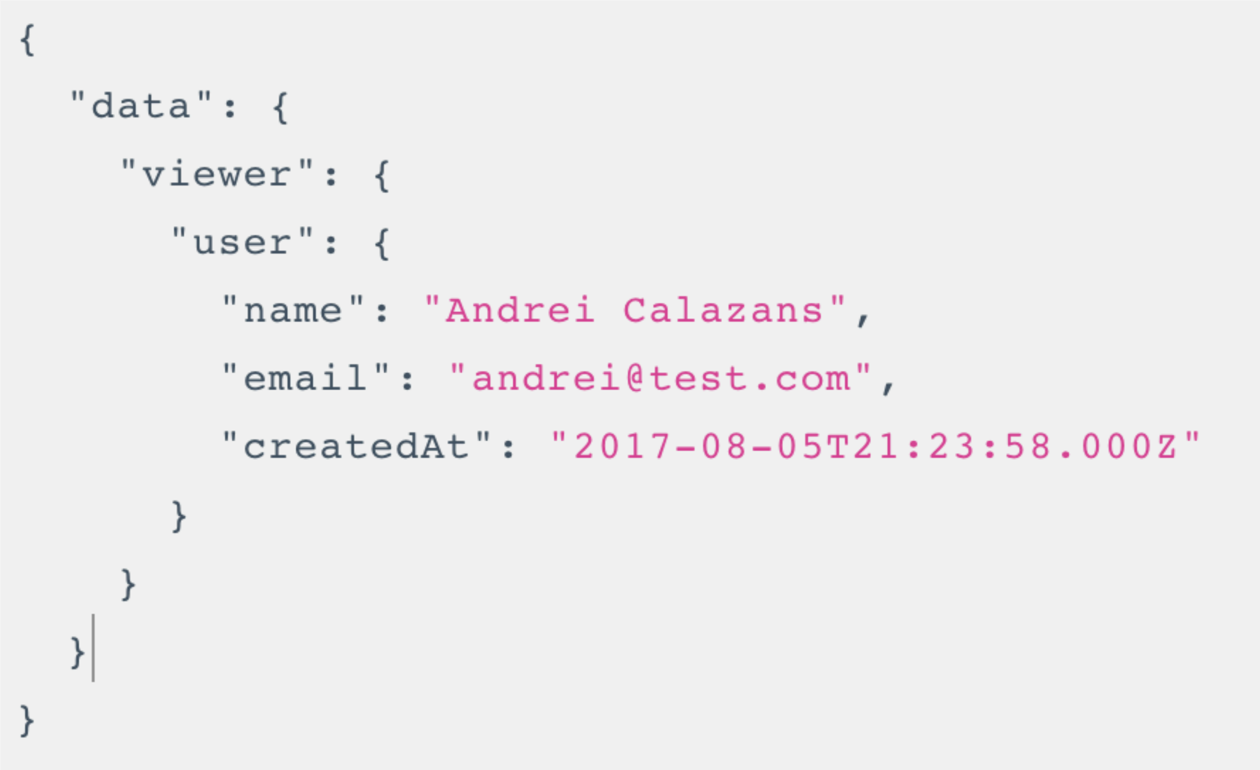
How to use it



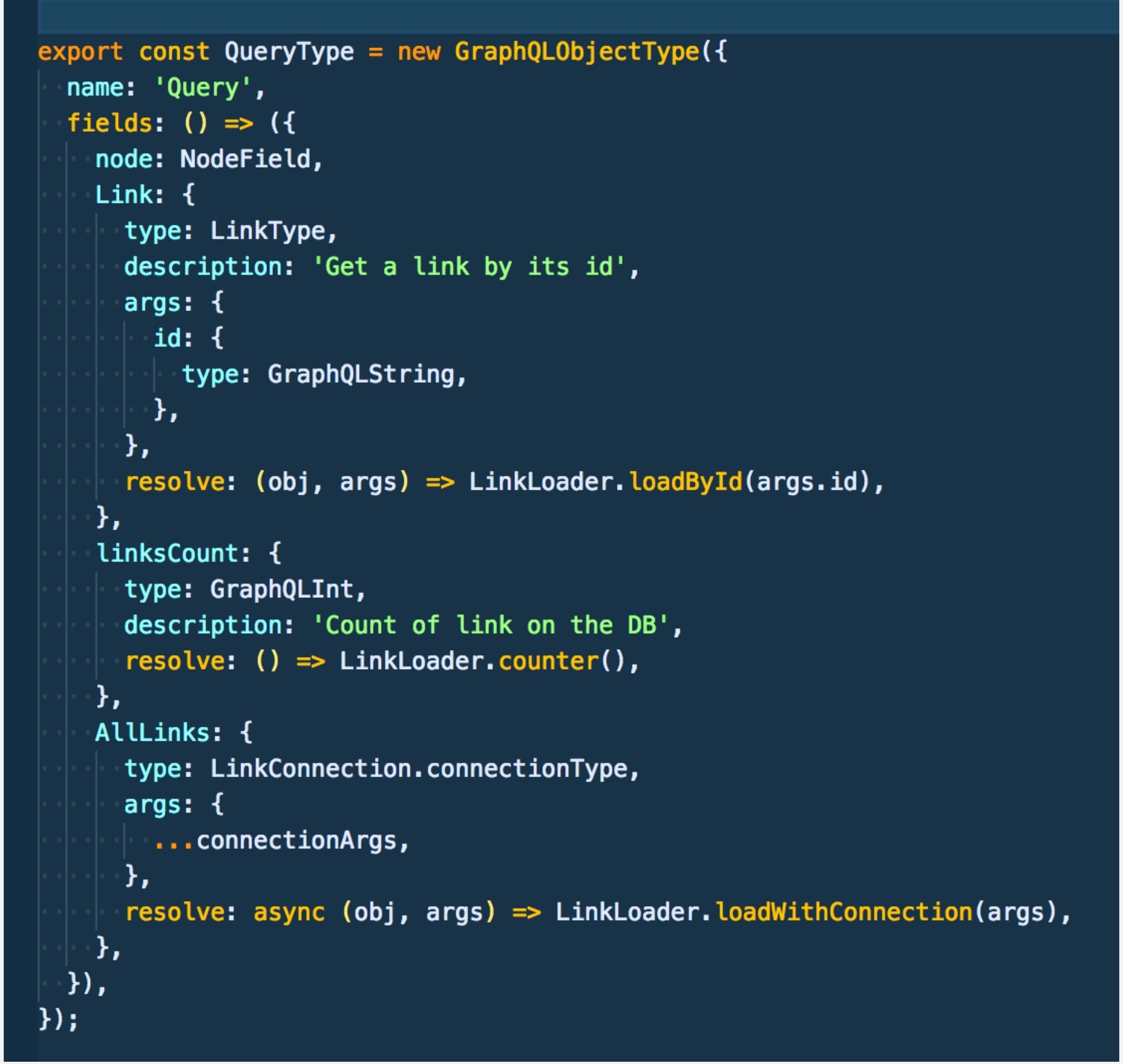
```
function fetchQuery(query, variables = {}) {
  return fetch('https://example.com/graphql', {
    method: 'POST',
    headers: {
      Accept: 'application/json',
      'Content-Type': 'application/json',
    },
    credentials: 'include',
    body: JSON.stringify({ query, variables }),
  }).then(resp => resp.json());
}
```



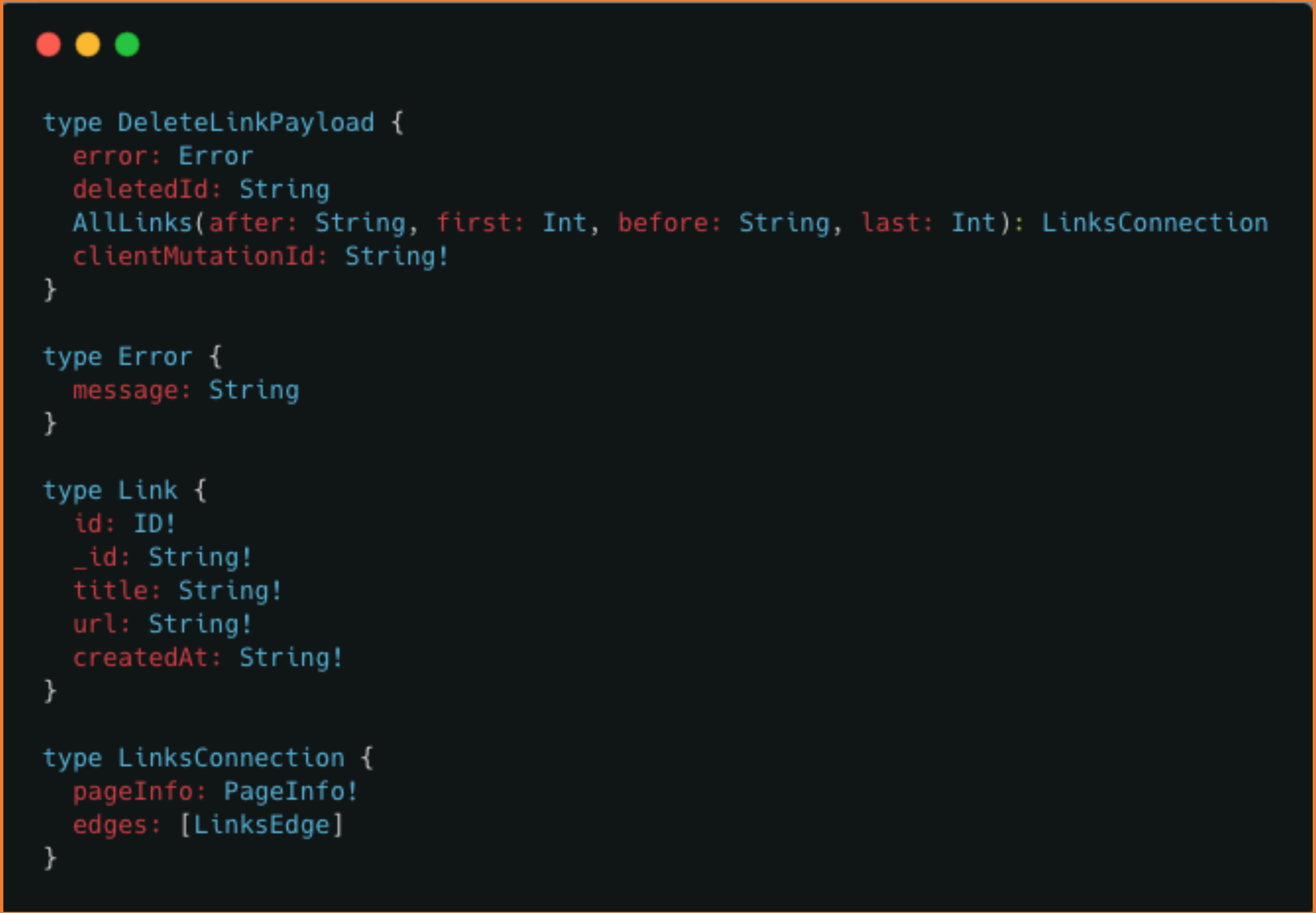
```graphql
query YouNameIt {
  viewer {
    user {
      name
      email
      createdAt
    }
  }
}
```



```json
{
  "data": {
    "viewer": {
      "user": {
        "name": "Andrei Calazans",
        "email": "andrei@test.com",
        "createdAt": "2017-08-05T21:23:58.000Z"
      }
    }
  }
}
```

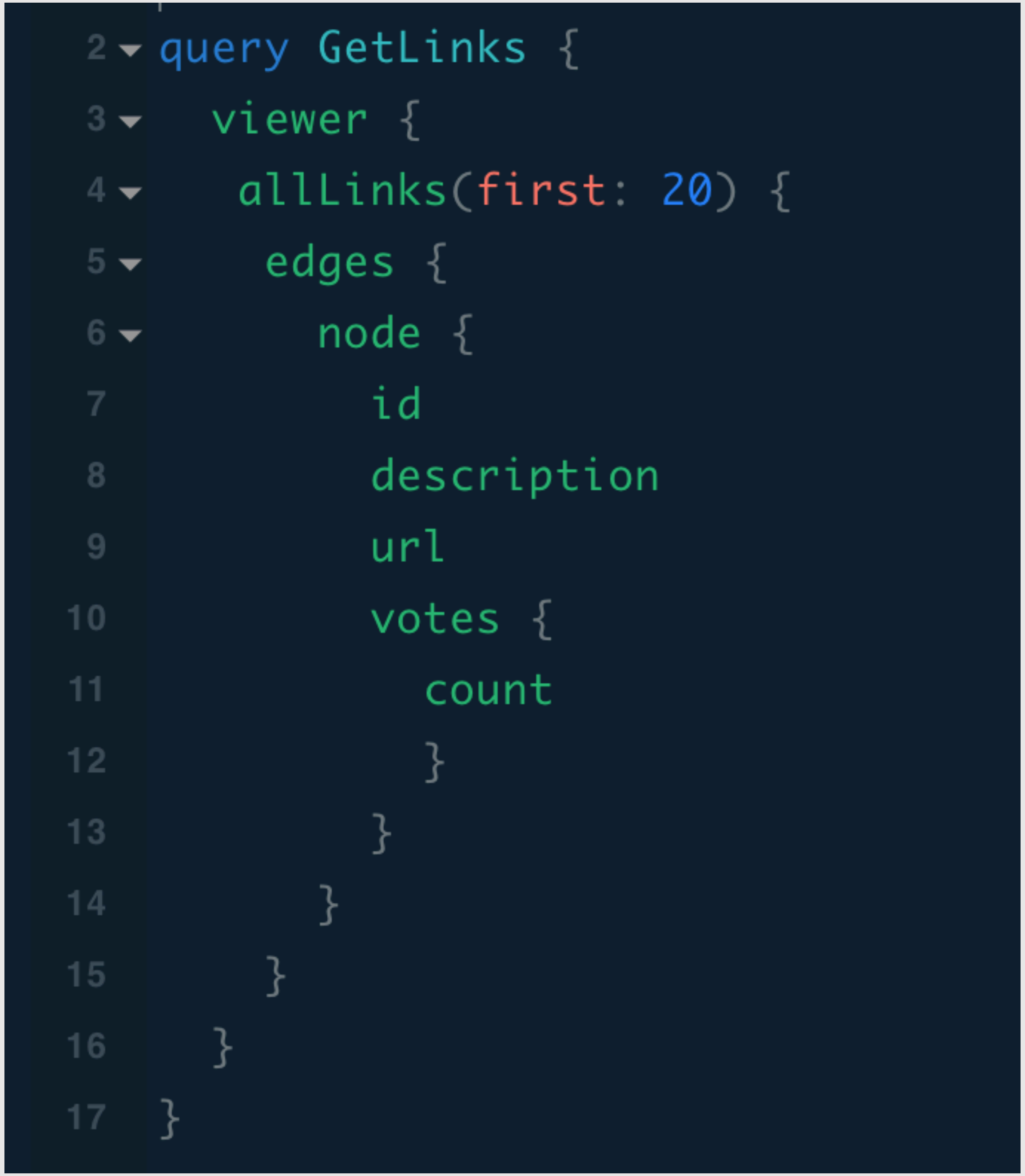


```
export const QueryType = new GraphQLObjectType({
  name: 'Query',
  fields: () => ({
    node: NodeField,
    Link: {
      type: LinkType,
      description: 'Get a link by its id',
      args: {
        id: {
          type: GraphQLString,
        },
      },
      resolve: (obj, args) => LinkLoader.loadById(args.id),
    },
    linksCount: {
      type: GraphQLInt,
      description: 'Count of link on the DB',
      resolve: () => LinkLoader.counter(),
    },
    AllLinks: {
      type: LinkConnection.connectionType,
      args: {
        ...connectionArgs,
      },
      resolve: async (obj, args) => LinkLoader.loadWithConnection(args),
    },
  }),
});
```

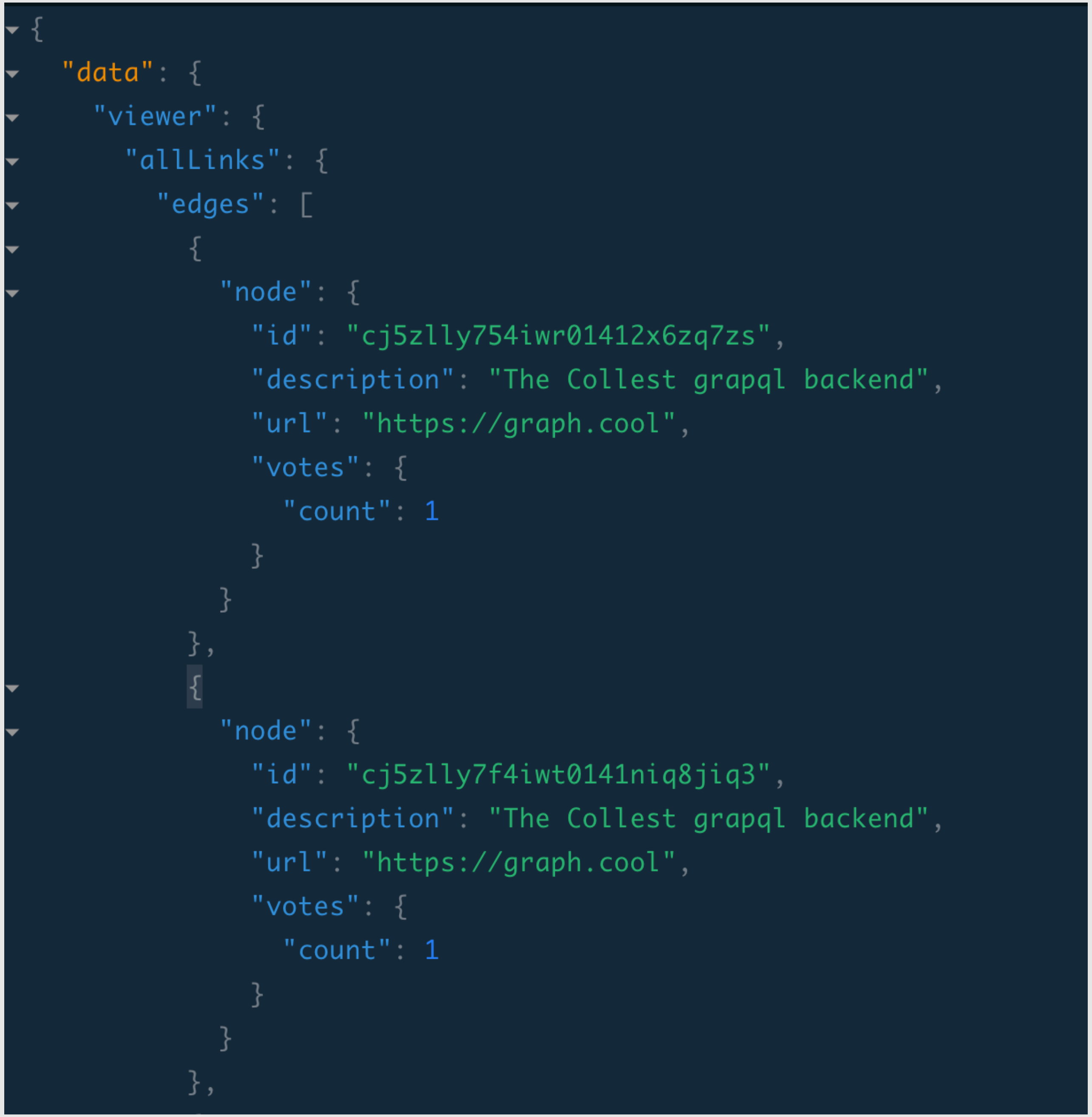


```
type DeleteLinkPayload {
  error: Error
  deletedId: String
  AllLinks(after: String, first: Int, before: String, last: Int): LinksConnection
  clientMutationId: String!
}

type Error {
  message: String
}

type Link {
  id: ID!
  _id: String!
  title: String!
  url: String!
  createdAt: String!
}

type LinksConnection {
  pageInfo: PageInfo!
  edges: [LinksEdge]
}
```

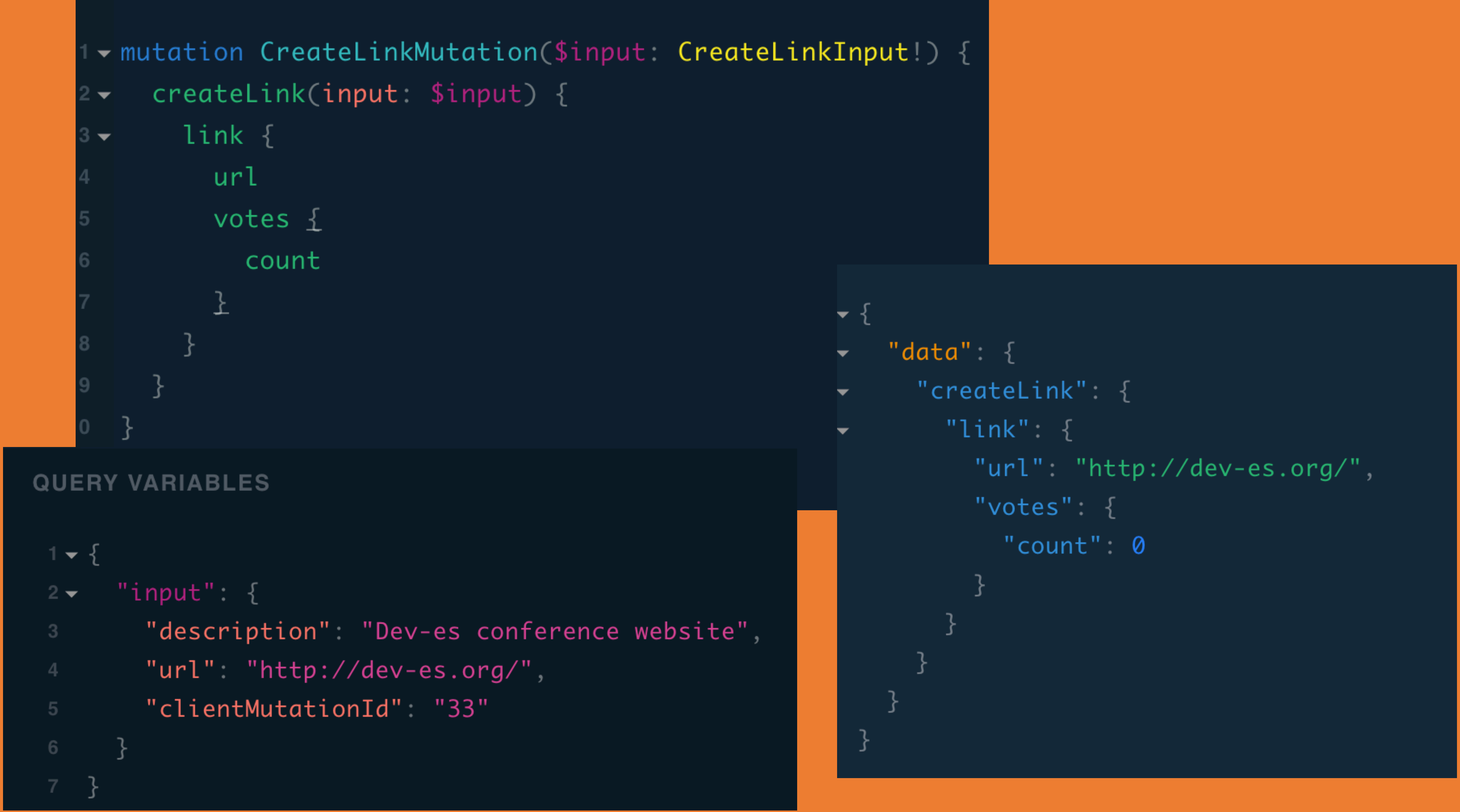
QUERY/MUTATION



```
query GetLinks {
  viewer {
   allLinks(first: 20) {
    edges {
      node {
        id
        description
        url
        votes {
          count
        }
      }
    }
   }
  }
}
```



```
{
  "data": {
    "viewer": {
      "allLinks": {
        "edges": [
          {
            "node": {
              "id": "cj5zlly754iwr01412x6zq7zs",
              "description": "The Collest grapql backend",
              "url": "https://graph.cool",
              "votes": {
                "count": 1
              }
            }
          },
          {
            "node": {
              "id": "cj5zlly7f4iwt0141niq8jiq3",
              "description": "The Collest grapql backend",
              "url": "https://graph.cool",
              "votes": {
                "count": 1
              }
            }
          },
```

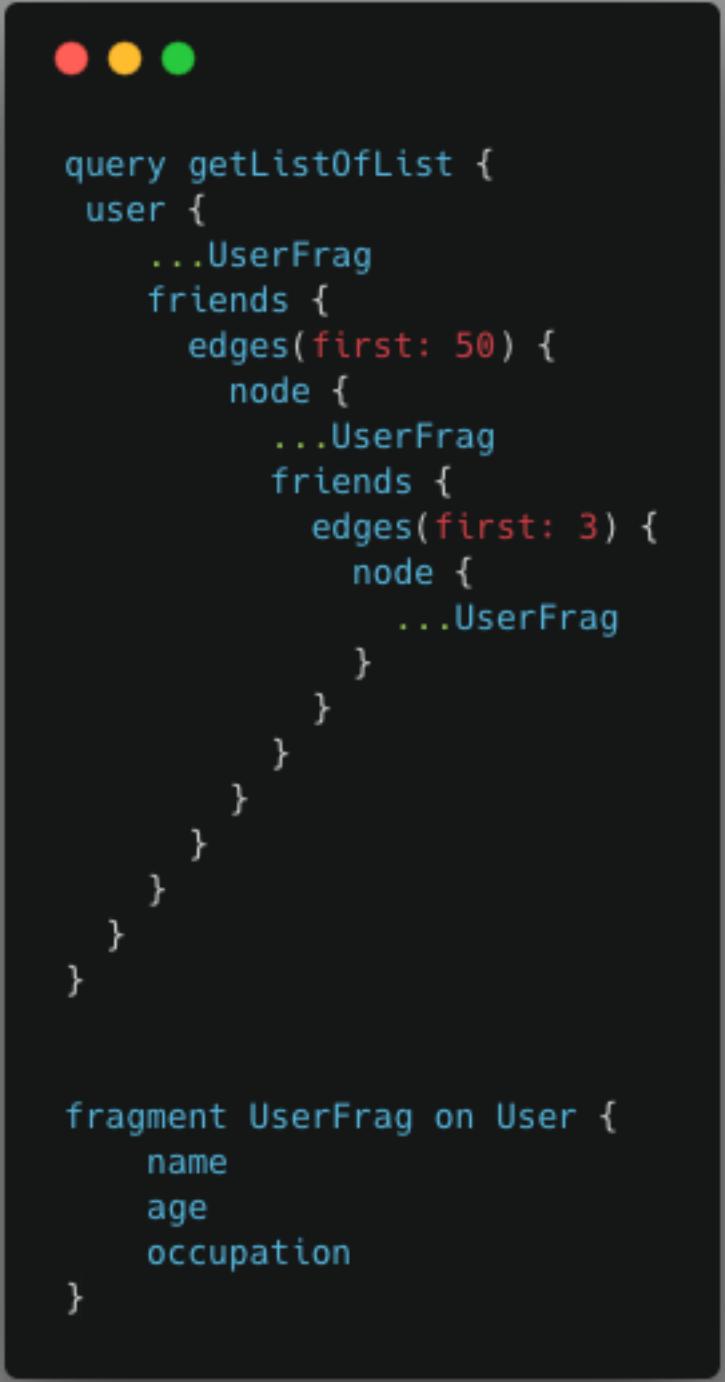


```
mutation CreateLinkMutation($input: CreateLinkInput!) {
  createLink(input: $input) {
    link {
      url
      votes {
        count
      }
    }
  }
}
```

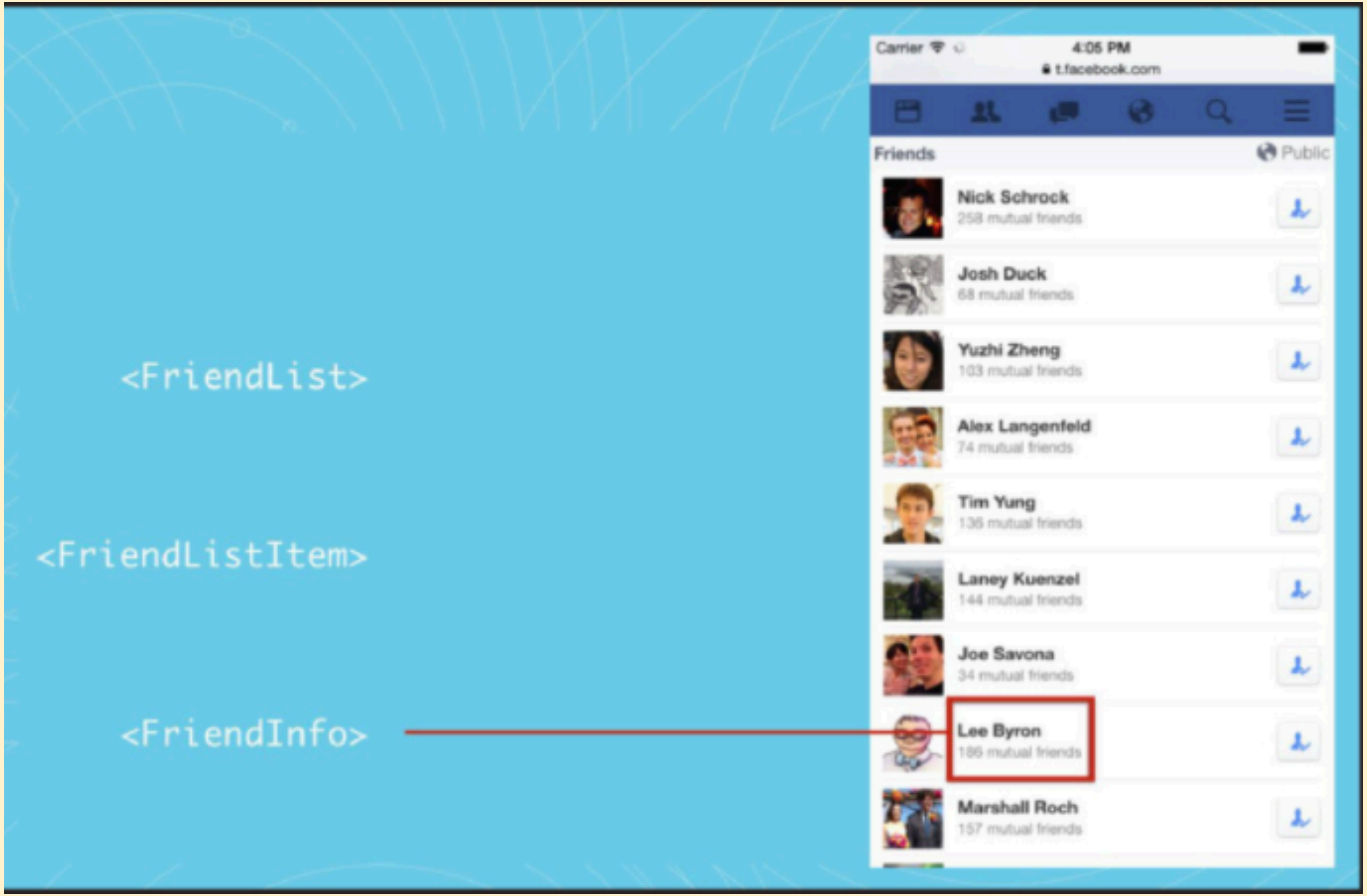
QUERY VARIABLES

```
{
  "input": {
    "description": "Dev-es conference website",
    "url": "http://dev-es.org/",
    "clientMutationId": "33"
  }
}
```

```
{
  "data": {
    "createLink": {
      "link": {
        "url": "http://dev-es.org/",
        "votes": {
          "count": 0
        }
      }
    }
  }
}
```



```
query getListOfList {
 user {
    ...UserFrag
    friends {
      edges(first: 50) {
        node {
          ...UserFrag
          friends {
            edges(first: 3) {
              node {
                ...UserFrag
              }
            }
          }
        }
      }
    }
  }
}


fragment UserFrag on User {
    name
    age
    occupation
}
```



<FriendList>

<FriendListItem>

<FriendInfo>

Nick Schrock
258 mutual friends

Josh Duck
68 mutual friends

Yuzhi Zheng
103 mutual friends

Alex Langenfeld
74 mutual friends

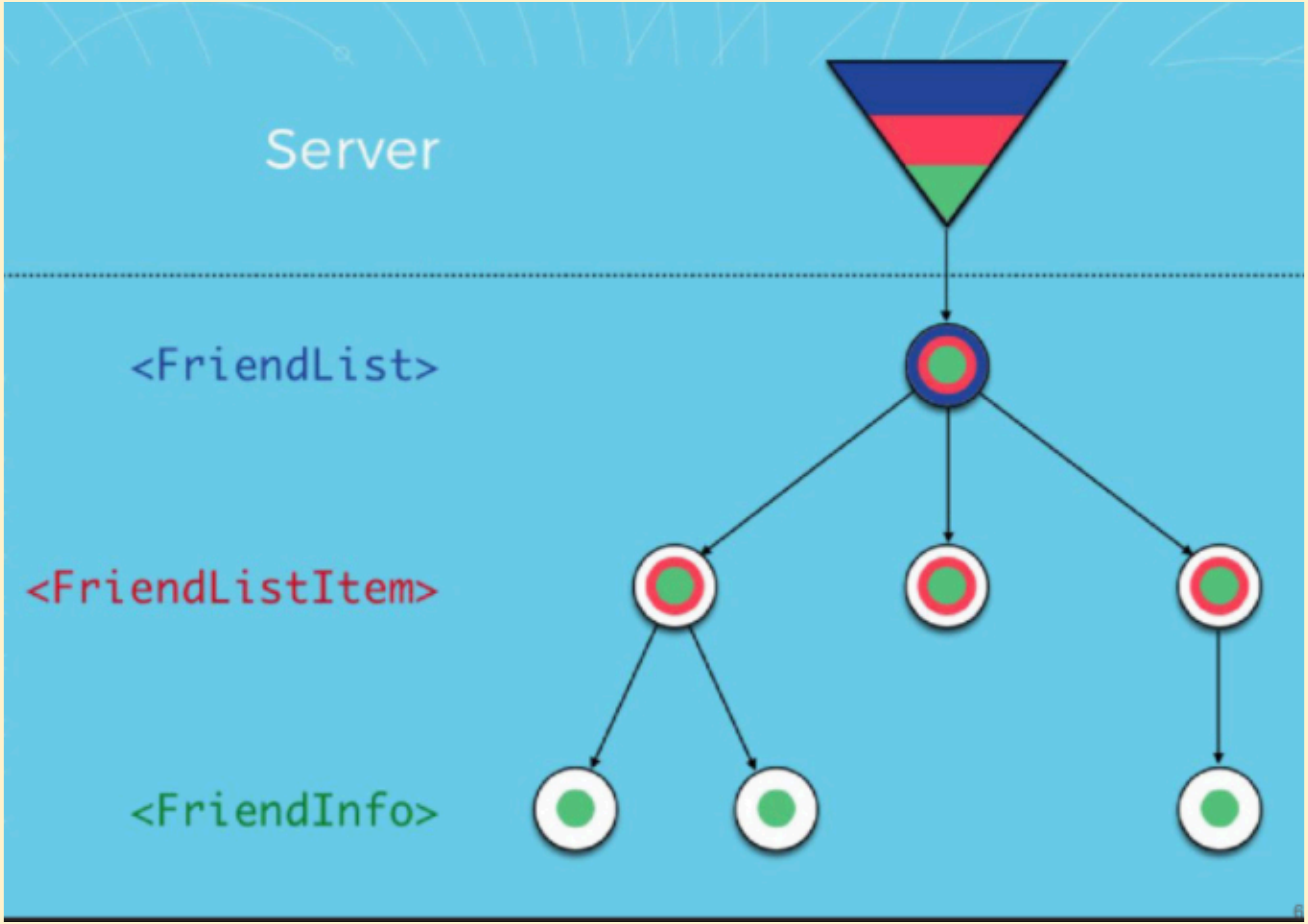
Tim Yung
136 mutual friends

Laney Kuenzel
144 mutual friends

Joe Savona
34 mutual friends

Lee Byron
186 mutual friends

Marshall Roch
157 mutual friends



Server

<FriendList>

<FriendListItem>

<FriendInfo>

GraphQL Clients

APOLLO

Relay

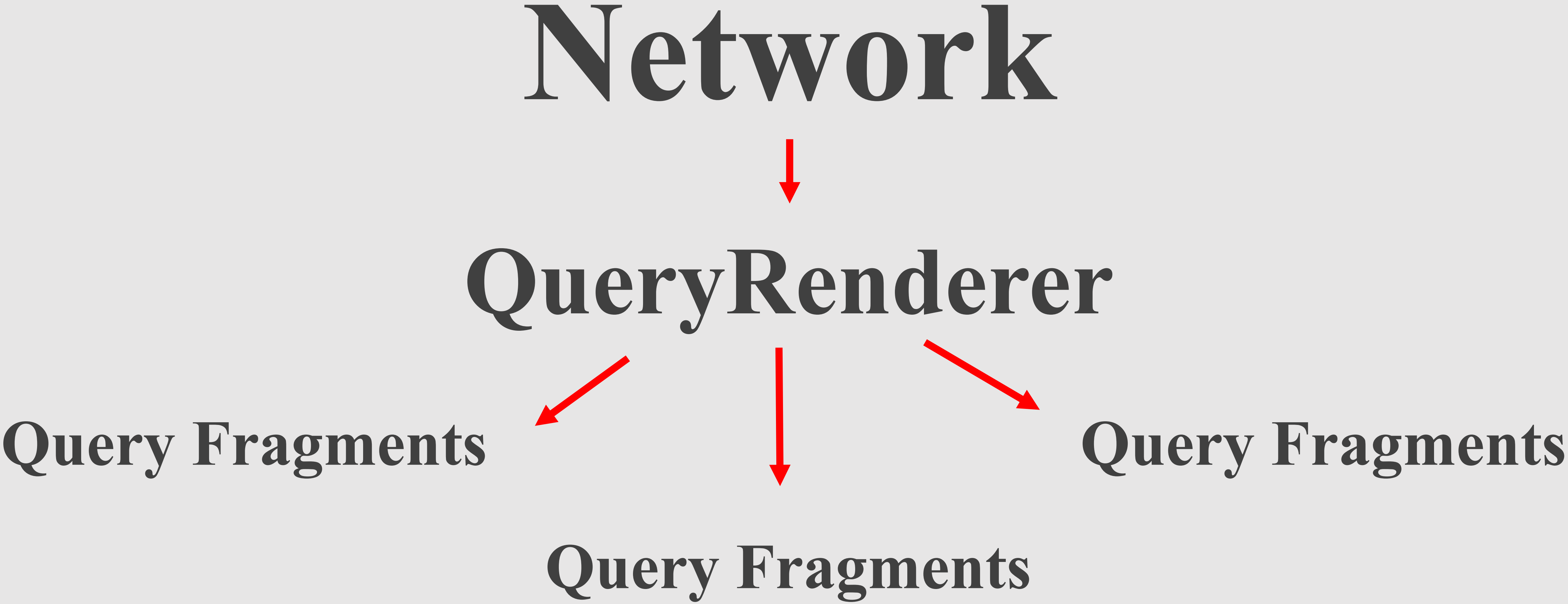
Relay-compiler

Creates _generated_ files

Type checks all your GraphQL

Optimizes it

Adds global keys



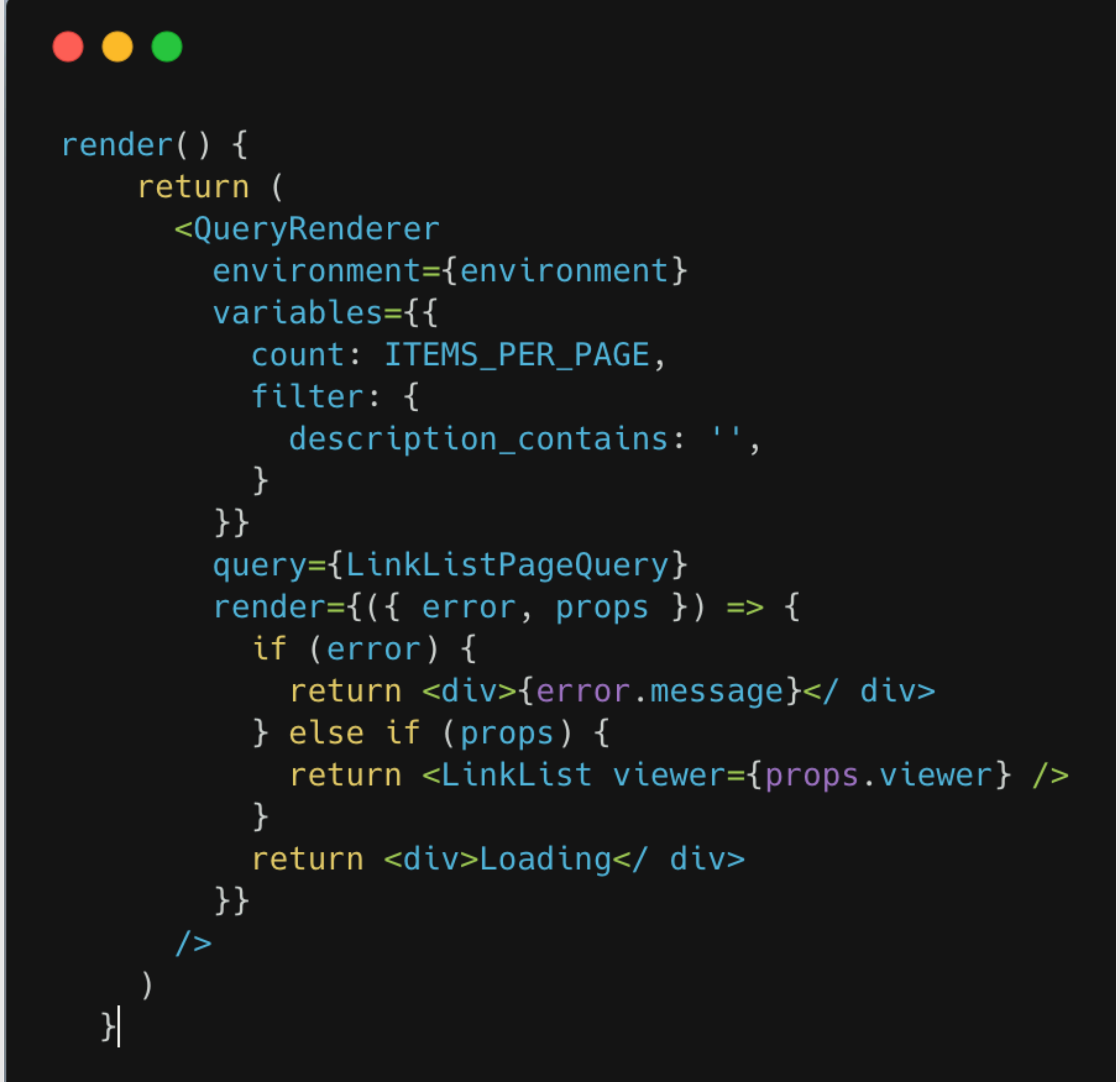
Network

QueryRenderer

Query Fragments

Query Fragments

Query Fragments

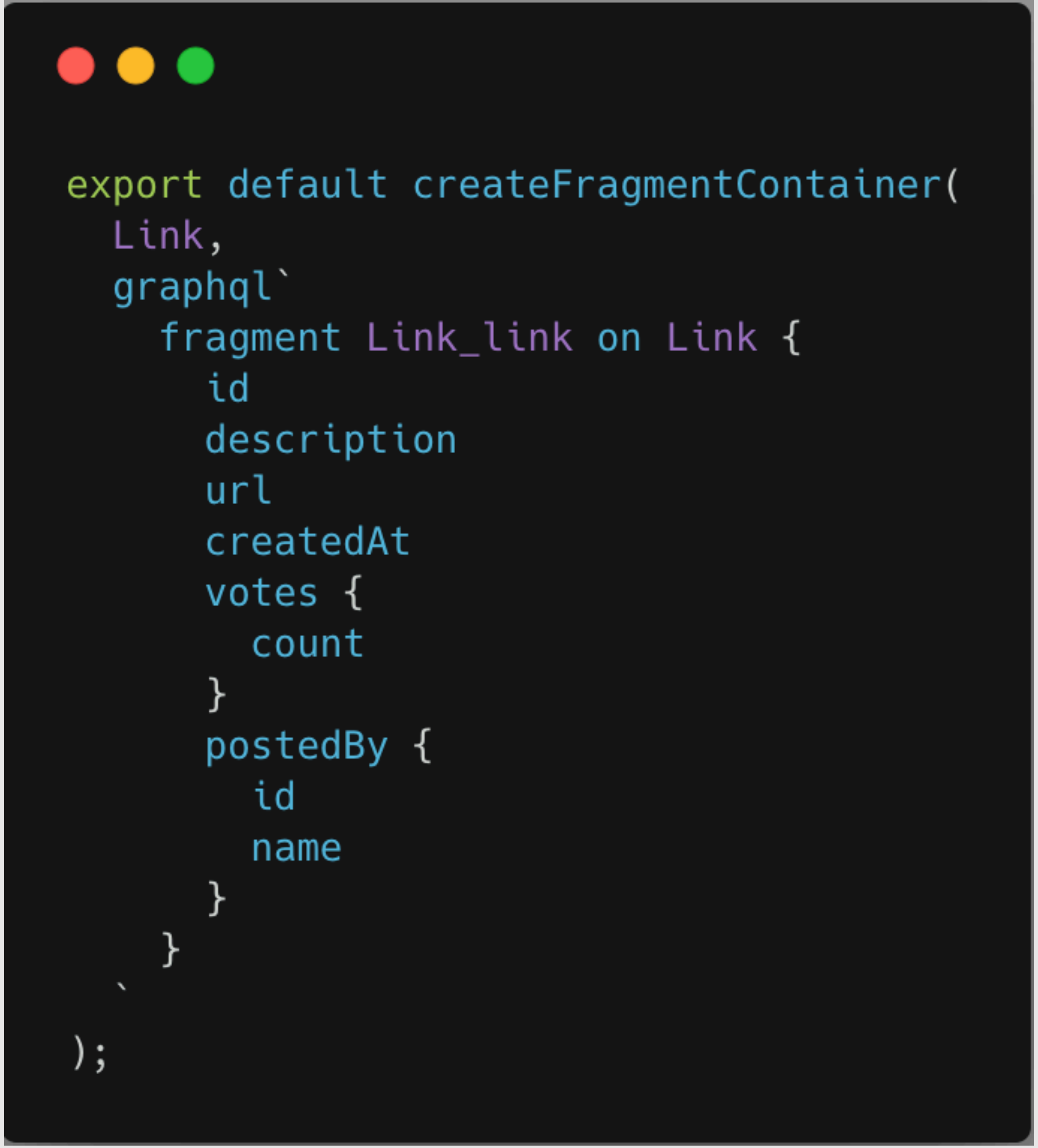


```
render() {
    return (
      <QueryRenderer
        environment={environment}
        variables={{
          count: ITEMS_PER_PAGE,
          filter: {
            description_contains: '',
          }
        }}
        query={LinkListPageQuery}
        render={({ error, props }) => {
          if (error) {
            return <div>{error.message}</ div>
          } else if (props) {
            return <LinkList viewer={props.viewer} />
          }
          return <div>Loading</ div>
        }}
      />
    )
  }
```

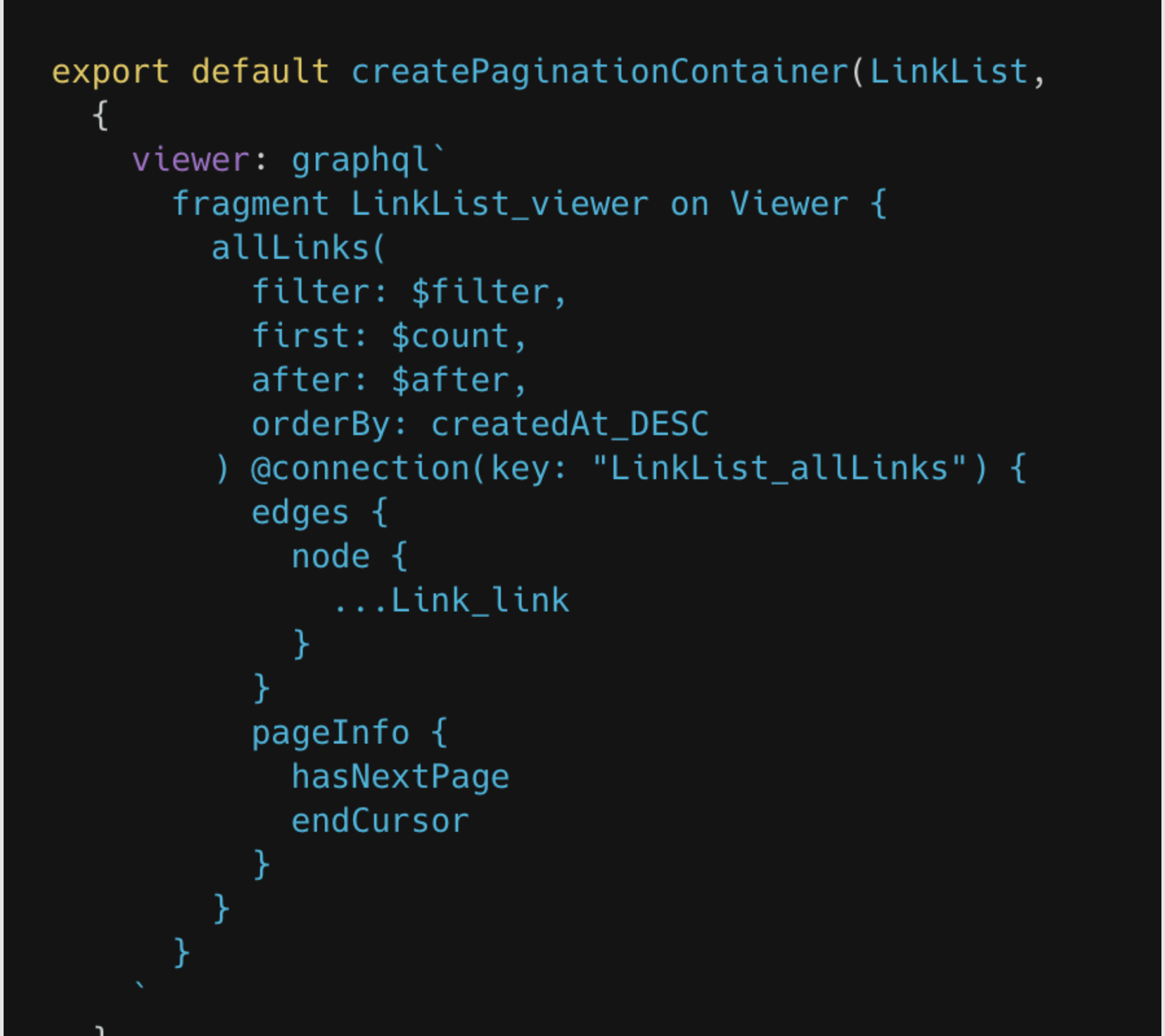
CreateFragmentContainer

CreatePaginationContainer

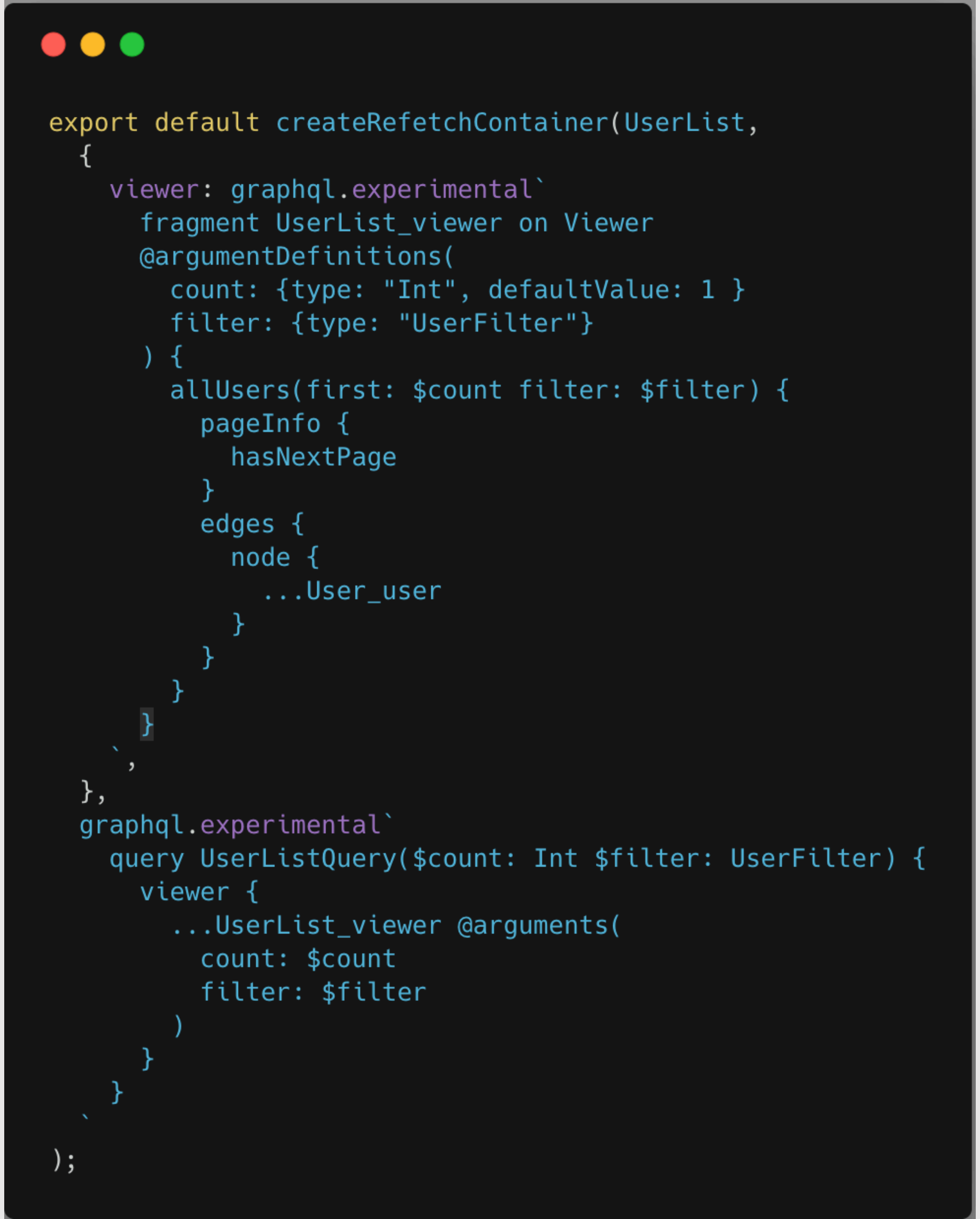
CreateRefetchContainer



```
export default createFragmentContainer(
  Link,
  graphql`
    fragment Link_link on Link {
      id
      description
      url
      createdAt
      votes {
        count
      }
      postedBy {
        id
        name
      }
    }
  `
);
```

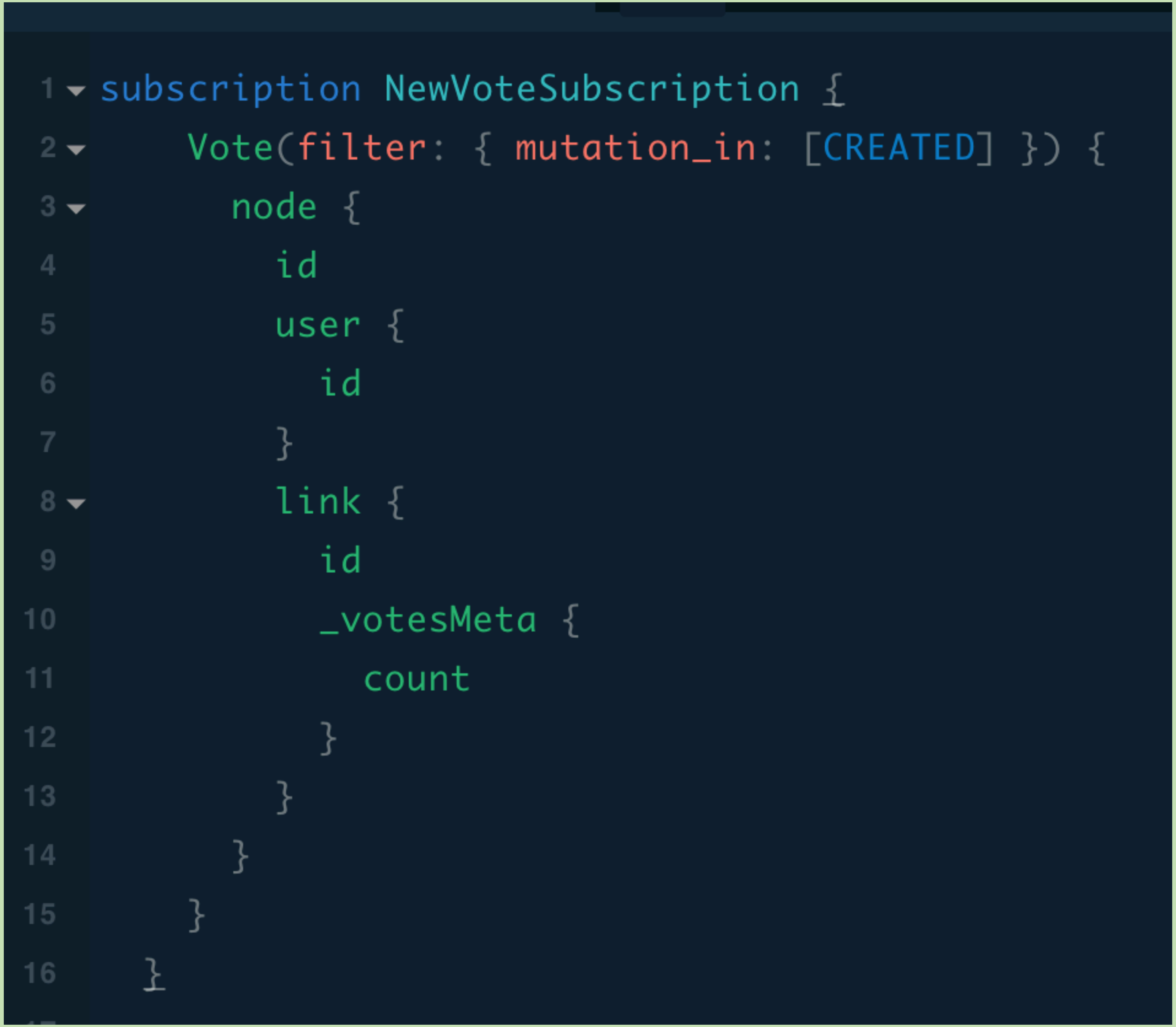


```
export default createPaginationContainer(LinkList,
  {
    viewer: graphql`
      fragment LinkList_viewer on Viewer {
        allLinks(
          filter: $filter,
          first: $count,
          after: $after,
          orderBy: createdAt_DESC
        ) @connection(key: "LinkList_allLinks") {
          edges {
            node {
              ...Link_link
            }
          }
          pageInfo {
            hasNextPage
            endCursor
          }
        }
      }
    `
  }
```



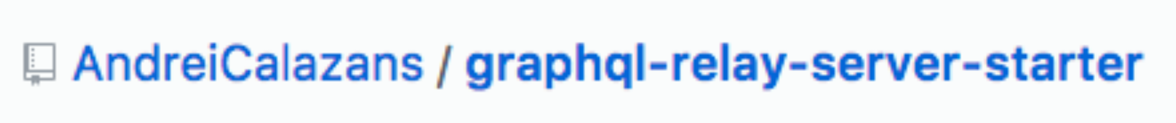
```
export default createRefetchContainer(UserList,
  {
    viewer: graphql.experimental`
      fragment UserList_viewer on Viewer
      @argumentDefinitions(
        count: {type: "Int", defaultValue: 1 }
        filter: {type: "UserFilter"}
      ) {
        allUsers(first: $count filter: $filter) {
          pageInfo {
            hasNextPage
          }
          edges {
            node {
              ...User_user
            }
          }
        }
      }
    `,
  },
  graphql.experimental`
    query UserListQuery($count: Int $filter: UserFilter) {
      viewer {
        ...UserList_viewer @arguments(
          count: $count
          filter: $filter
        )
      }
    }
  `
);
```

Subscriptions

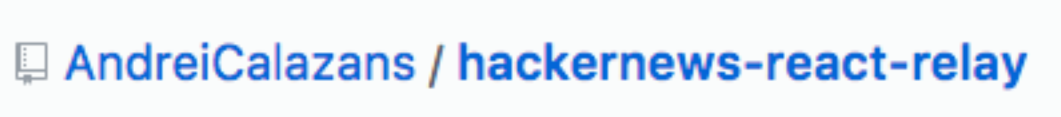


```
subscription NewVoteSubscription {
    Vote(filter: { mutation_in: [CREATED] }) {
      node {
        id
        user {
          id
        }
        link {
          id
          _votesMeta {
            count
          }
        }
      }
    }
  }
```

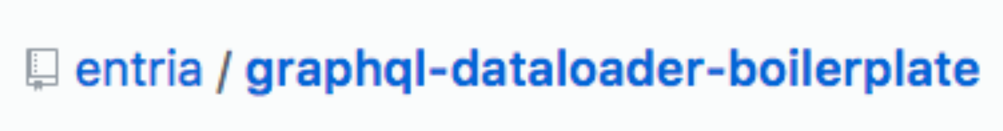
LINK



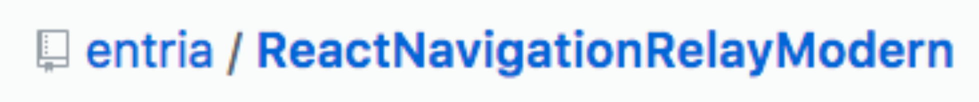
AndreiCalazans / **graphql-relay-server-starter**



AndreiCalazans / **hackernews-react-relay**



entria / **graphql-dataloader-boilerplate**



entria / **ReactNavigationRelayModern**

Muito Obrigado!