Challenge: Apollo Client and GraphQL Pagination

Test your understanding of GraphQL and Apollo Client and implement pagination!

WE'LL COVER THE FOLLOWING

Problem Statement

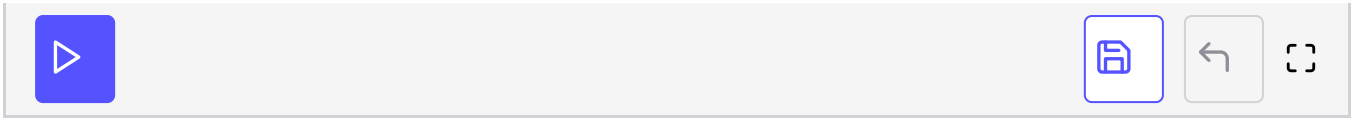
1. Extend the `repositories` list field by querying an ordered list of repositories which is ordered by the number of stargazers
2. Extract the content of a repository `node` to a GraphQL a reusable fragment
3. Add the pagination feature for list of repositories
    - Add the `pageInfo` field with its `endCursor` and `hasNextPage` fields in the query
    - Add the `after` argument and introduce a new `$cursor` variable for it
    - Perform the first query without a `cursor` argument
    - Perform a second query with the `endCursor` of the previous query result as `cursor` argument.

Environment Variables

Key:	Value:
REACT_APP_GITHUB...	Not Specified...
GITHUB_PERSONAL...	Not Specified...

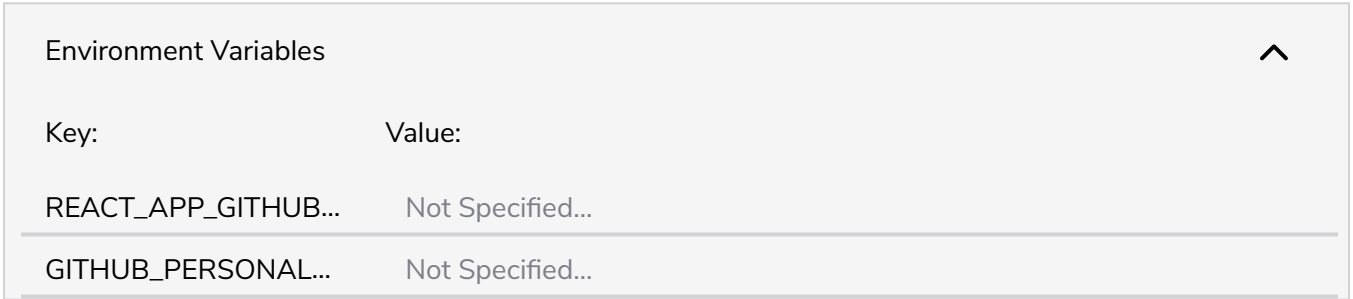
```
import 'dotenv/config';
import 'cross-fetch/polyfill';
import ApolloClient, { gql } from 'apollo-boost';

const client = new ApolloClient({
  uri: 'https://api.github.com/graphql',
```

```
  request: operation => {
    operation.setContext({
      headers: {
        authorization: `Bearer ${process.env.GITHUB_PERSONAL_ACCESS_TOKEN}`,
      },
    });
  },
});

const GET_REPOSITORIES_OF_ORGANIZATION = gql`
  query($organization: String!) {
    organization(login: $organization) {
      name
      url
      repositories(first: 5) {
        edges {
          node {
            name
            url
          }
        }
      }
    }
  }
`;

client
  .query({
    query: GET_REPOSITORIES_OF_ORGANIZATION,
    variables: {
      organization: 'the-road-to-learn-react',
    },
  })
  .then(console.log);
```



Solution

Let's have a look at the solution:




```
import 'dotenv/config';
import 'cross-fetch/polyfill';
import ApolloClient, { gql } from 'apollo-boost';

const client = new ApolloClient({
  uri: 'https://api.github.com/graphql',
  request: operation => {
    operation.setContext({
```

```
      headers: {
        authorization: `Bearer ${process.env.GITHUB_PERSONAL_ACCESS_TOKEN}`,
      },
    });
  },
});

const GET_REPOSITORIES_OF_ORGANIZATION = gql`
  query($organization: String!, $cursor: String) {
    organization(login: $organization) {
      name
      url
      repositories(
        first: 5
        orderBy: { direction: DESC, field: STARGAZERS }
        after: $cursor
      ) {
        edges {
          node {
            ...repository
          }
        }
        pageInfo {
          endCursor
          hasNextPage
        }
      }
    }
  }
  fragment repository on Repository {
    name
    url
  }
`;

client
  .query({
    query: GET_REPOSITORIES_OF_ORGANIZATION,
    variables: {
      organization: 'the-road-to-learn-react',
      cursor: undefined,
    },
  })
  // resolve first page
  .then(result => {
    const { pageInfo, edges } = result.data.organization.repositories;
    const { endCursor, hasNextPage } = pageInfo;

    console.log('second page', edges.length);
    console.log('endCursor', endCursor);

    return pageInfo;
  })
  // query second page
  .then(({ endCursor, hasNextPage }) => {
    if (!hasNextPage) {
      throw Error('no next page');
    }

    return client.query({
      query: GET_REPOSITORIES_OF_ORGANIZATION,
      variables: {
```

```
      organization: 'the-road-to-learn-react',
      cursor: endCursor,
    },
  });
})
// resolve second page
.then(result => {
  const { pageInfo, edges } = result.data.organization.repositories;
  const { endCursor, hasNextPage } = pageInfo;

  console.log('second page', edges.length);
  console.log('endCursor', endCursor);

  return pageInfo;
})
// log error when there is no next page
.catch(console.log);
```