

Challenge: Implement `removeStar` Mutation

Test your skills and implement another mutation using Apollo Client!

WE'LL COVER THE FOLLOWING ^

- Problem Statement
- Solution

Problem Statement

Implement the `removeStar` mutation next to the `addStar` mutation and then execute it for the same repository that you are starring. Your console should first log the `addStar` mutation and then the `removeStar` mutation.

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!, $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);

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

client
  .mutate({
    mutation: ADD_STAR,
    variables: {
      repositoryId: 'MDEwO1JlcG9zaXRvcnk2MzM1MjkwNw==',
    },
  })
  .then(console.log);

```



Solution

Let's have a look at the solution:

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!, $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);

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

const REMOVE_STAR = gql`
  mutation RemoveStar($repositoryId: ID!) {
    removeStar(input: { starrableId: $repositoryId }) {
      starrable {
        id
        viewerHasStarred
      }
    }
  }
`;

client
  .mutate({
    mutation: ADD_STAR,
    variables: {
      repositoryId: 'MDEwO1JlcG9zaXRvcnk2MzM1MjkwNw==',
    },
  })
  .then(console.log);

client
  .mutate({
    mutation: REMOVE_STAR,
    variables: {
      repositoryId: 'MDEwO1JlcG9zaXRvcnk2MzM1MjkwNw==',
    },
  })
  .then(console.log);

```

