useMutation

A React Apollo's useMutation Hook



```
(optional) options to
                                                                                                                                                        configure the
                                                                                                                                                       behaviour of the
                tuple that contains mutate function needed to execute the
                                                                                                                    type of variables
                                                                                                                                                       component making
               mutation AND object containing details of mutation result + additional functions (e.g. refetch)
                                                                                                                  the mutation is to be
                                                                                                                                                       the mutation (e.g.
                                                                                                                                                      passing in variables)
                                                                                                                     executed with
const [mutate, { data, loading, error }] = useMutation<TData, TVariables>(mutation, options);
                                                                                                    type of data
                                                                                                                                          graphal AST
                                             boolean
           function used
                                                         runtime error
                                object
                                                                                                  expected to be
                                            indicating
                                                          of mutation
                                                                                                                                           mutation
           to execute the
                              containing
                                                                                                  returned from
                                            whether
                                                                                                                                           document
                                                           request
              mutation
                              results of
                                          mutation is in
                                                                                                     mutation
                                                                                                                                          constructed
                               mutation
                                              flight
                                                                                                                                        with the agl tag
```

Execute a mutation in a component

```
import React, { useState } from "react";
                                                                                  import agl tag from apollo-boost
import gql from "apollo-boost";
                                                                                    import useMutation from react apollo
import { useMutation } from "@apollo/react-hooks";
import
  UpdateSuperhero as UpdateSuperheroData,
                                                                                   import autogenerated type
definitions (data + variables)
  UpdateSuperheroVariables
                                                                                       for the mutation
} from "./__generated__/UpdateSuperhero";
const UPDATE_SUPERHER0 = gql`
  mutation UpdateSuperhero($id: ID!) {
     updateSuperhero(id: $id) {
                                                                                    create GraphQL mutation
                                                                                    document with the agl tag
        name
export const FunctionComponent = () => {
  const [superhero, updateSuperhero] = useState({
     id: "001",
     name: "Spider-Man"
  });
  const
     updateSuperheroMutation,
                                                                                     pass the relevant type definitions and mutation
     { loading, error }
                                                                                   document to the useMutation
   = useMutation<
                                                                                           hook
     UpdateSuperheroData,
                                                                                   return the mutation function
     UpdateSuperheroVariables
                                                                                    and the loading, and error
                                                                                     results of the mutation
  >(UPDATE_SUPERHERO,
     variables: {
                                                                                   pass the variables needed to
       id: "002"
                                                                                      execute the mutation
     onCompleted: (data) => {
                                                                                   use the onCompleted callback
                                                                                   to run functionality when the
       updateSuperhero(data);
                                                                                      mutation is complete
                                                                                        successfully
  const mutationLoadingMessage = loading ? (
     <h2>Loading!</h2>
   null;
  const mutationLoadingError = error ? (
    <h2>Error!</h2>
  null;
  return
     <div>
       <h2>{superhero.name}</h2>
       <button onClick={() => updateSuperheroMutation()}>
                                                                                     triager mutation to run
          Change Superhero!
       </br>
                                                                                 when mutation is in the loading or
       {mutationLoadingMessage}
                                                                                    error state, component
                                                                                   presents loading or error messages respectively
       {mutationLoadingError}
     </div>
```