

```
const MutationType = new GraphQLObjectType({ // ajout d'un utilisateur
  name : 'Mutation',
  fields : {
    addUser : { //nom de la fonction
      type : UserType, //elle retourne un UserType
      args : {
        firstName : { type: new GraphQLNonNull(GraphQLString) }, // GraphQLNonNull permet de verifier que le champs est bien renseigner
        age : { type : new GraphQLNonNull(GraphQLInt)},
        companyId : { type: GraphQLString }
      },
      resolve(parentValue, args){
        return axios.post(`http://localhost:3000/users/`, {firstName: args.firstName, age: args.age, companyId: args.companyId}).then((response)=>{
          return response.data; // la fonction retourne donc un UserType
        })
      }
    },
    deleteUser: { // le nom de la fonction
      type : UserType, //le type de retour de la requete
      args : [{ // les arguments de la fonction
        id : { type: new GraphQLNonNull(GraphQLString)}
      ]},
      resolve(parentValue, args){ // fonction qui se passe quand on valide la fonction deleteUser
        return axios.delete(`http://localhost:3000/users/${args.id}`).then((response)=>{
          return response.data; // la fonction retourne donc un UserType
        })
      }
    }
  }
});
```

```
1 mutation{
2   deleteUser(id:"2"){
3     id,
4   }
5 }
6 }
```

```
{
  "data": {
    "deleteUser": {
      "id": null
    }
  }
}
```

Q Search Mutation...

No Description

FIELDS

```
addUser(
  firstName: String!
  age: Int!
  companyId: String
): User

deleteUser(id: String!): User
```