60. Resolver - post.ts / Mutation - vote()

#graphql #resolver #authentication #mutation #backend #typeorm

Implement the mutation for voting

- It will just be an upvote or downvote no matter how many points user gives

/resolvers/post.ts

```
@Mutation(() => Boolean)
@UseMiddleware(isAuth)
async vote(
  @Arg("value", () => Int) value: number,
  @Arg("postId", () => Int) postId: number,
  @Ctx() { req }: MyContext
) {
  const isUpdoot = value !== -1;
  const realValue = isUpdoot ? 1 : -1;
  const userId = req.session.userId;

  // This also works but we can also do it in the sql query as below
  // await Updoot.insert({
  //   userId,
  //   postId,
  //   value: realValue,
  // });

  await getConnection().query(
    `
    START TRANSACTION;

    insert into updoot("userId", "postId", "value") //quotes are needed to preserve capital letters
    values (${userId}, ${postId}, ${realValue});

    update post
    set points = points + ${realValue}
    where id = ${postId};
    COMMIT;
    `
  );

  return true;
}
```

- Now we can vote up or down on posts, however we can’t cancel or change our vote afterwards.