42. Resolver - post.ts / update Mutation - createPost()

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

- The createPost() mutation needs to be updated since we added a text field, a points field and a creatorId to it
- points field will default to zero so we don't need to set it at post creation
- Define an @InputType for passing into createPost() as an input parameter

post.ts

```
@InputType()
class PostInput {
  @Field()
  title: string;
  @Field()
  text: string;
}
```

- We can take the creatorId from our context

post.ts

```
@Mutation(() => Post)
  async createPost(
    @Arg("input") input: PostInput,
    @Ctx() { req }: MyContext
  ): Promise<Post> {
    return Post.create({
      ...input,
      creatorId: req.session.userId,
    }).save();
  }
```

Allow only logged in users to crete posts

- Simplest way to do this would be to check if a user is logged in. We will implement a better way in next section

post.ts

```
@Mutation(() => Post)
  async createPost(
```

```
    @Arg("input") input: PostInput,
    @Ctx() { req }: MyContext
  ): Promise<Post> {
    if (!req.session.userId) {
      // if user not logged in
      throw new Error("not authenticated");
    }

    return Post.create({
      ...input,
      creatorId: req.session.userId,
    }).save();
  }
```