51. Pagination for Posts - Resolver post.ts / Query posts()

#pagination #resolver #query #graphql #backend #typeorm

- Right now the posts query retrieves all of the posts from the server
- We have to add pagination to limit the number of posts retrieved at a time

---

Define a PaginatedPosts type

- This will hold the posts that are fetched as well as a boolean value that tells whether there are more posts to fetch or not

/resolvers/post.ts

```
@ObjectType()
class PaginatedPosts {
  @Field(() => [Post])
  posts: Post[];
  @Field()
  hasMore: boolean;
}
```

Implement the query

- We will use typeorm.io/#/select-query-builder
- There's offset based pagination and cursor based pagination. We will implement cursor based since offset based can cause performance issues as well as refresh issues when new posts are being added frequently,
- The limit determines how many posts should be fetched and put into the list (we set max. as 50)
- The cursor determines a specific location (in our case a date) that the list will start from
- We fetch realLimitPlusOne posts to see if there's more posts than realLimit

/resolvers/post.ts

```
import { getConnection } from "typeorm";

@Query(() => PaginatedPosts)
  async posts(
```

```
    @Arg("limit", () => Int) limit: number,
    // the first fetch will not have a cursor so cursor should be nullable
    @Arg("cursor", () => String, { nullable: true }) cursor: string | null
  ): Promise<PaginatedPosts> {

    const realLimit = Math.min(50, limit);
    const realLimitPlusOne = Math.min(50, limit) + 1;

    const qb = getConnection()
      .getRepository(Post)
      .createQueryBuilder("p")
      .orderBy('"createdAt"', "DESC") // mind the double quotes '" ... "'
      .take(realLimitPlusOne);

    if (cursor) {
      qb.where("createdAt < :cursor", { cursor: new Date(parseInt(cursor)) });
    }

    const posts = await qb.getMany();

    return {
      posts: posts.slice(0, realLimit),
      hasMore: posts.length === realLimitPlusOne,
    }; // see if there's more posts to retrieve
  }
```