

55. Resolver - post.ts / FieldResolver - textSnippet()

#graphql

#resolver

#fieldresolver

#query

#backend

#frontend

#graphql-codegen

Implement textSnippet FieldResolver

- Assuming the `text` field of a `post` is very large, we wouldn't want to download and display all of the texts on the `homepage`
- We could display only a `snippet` of it on the `frontend` like this: `<Text>{p.text.slice(0,50)}...`
`</Text>` but still, all of the `data` would be downloaded to the `client` needlessly
- Instead we define a `FieldResolver` that acts on `Post` objects and can be accessed with `post.textSnippet` from the `frontend`
- Note that** we changed `@Resolver()` to `@Resolver(Post)` to let GraphQL know that this `Resolver` is for `Post` objects. Only then we can define a `FieldResolver` that will act on `Post` objects

/resolvers/post.ts

```
@Resolver(Post)
export class PostResolver {
  @FieldResolver(() => String)
  textSnippet(
    @Root() root: Post // get called for Post objects
  ) {
    return root.text.slice(0, 150);
  }
}
```

Update GraphQL and generate TypeScript code

- Remove `text` and add `textSnippet` to the returned values of the `posts` query
- Even though we do not have a `textSnippet` column in the database, the `FieldResolver` will provide this field as a return value, so only the snippets will be sent to the `client` from the `server`

/graphql/posts.graphql

```
query Posts($limit: Int!, $cursor: String) {
  posts(limit: $limit, cursor: $cursor) {
    hasMore
    posts {
      id
      createdAt
      textSnippet
    }
  }
}
```

```

      updatedAt
      title
      textSnippet
    }
  }
}

```

- Run codegen to generate the **TypeScript** code for **graphql**

```
yarn gen
```

- and now we have the updated **usePostsQuery()** hook in **/generated/graphql.tsx**

Update homepage to use **post.textSnippet**

- We update the homepage to use this **textSnippet**

web/src//index.tsx

```

{data!.posts.posts.map((p) => (
  <Box key={p.id} p={5} shadow="md" borderWidth="1px">
    <Flex>
      <Heading fontSize="xl">{p.title}</Heading>
      <Flex ml="auto">
        <Text>posted by:</Text>
        <Text ml={2} fontWeight="bold">
          p.creator.username}
        </Text>
      </Flex>
    </Flex>
    <Text>{p.textSnippet}...</Text>
  </Box>
))}

```