

```
1  import { makeExecutableSchema } from 'graphql-tools';
2
3  import resolvers from './resolvers';
4
5  const typeDefs = `
6    type Author {
7      id: Int!
8      firstName: String!
9      lastName: String!
10     posts: [Post!]
11     stars: Int!
12   }
13
14   type Post {
15     id: Int!
16     title: String!
17     text: String!
18     views: Int!
19     author: Author!
20     genre: String!
21   }
22
23   type Query {
24     author(firstName: String!, lastName: String!, id: Int!): Author!
25     authors: [Author!]
26     posts: [Post!]
27     post(id: Int!): Post!
```

BREAKING CHANGES AND SCHEMA VALIDATION

- ▶ CI tooling is available to automatically identify breaking changes in a pull request
- ▶ Because the tooling is validating static queries against the schema, you can ensure that there will be no client-side runtime errors due to mismatched expectations
- ▶ CI integrations allow for the entire team to be notified when a breaking change is merged



IMPLEMENTATION

**SPENDING TIME TO
SAVE TIME**