Sharing models

> warning **Warning** This chapter applies only to the code first approach.

One of the biggest advantages of using Typescript for the backend of your project is the ability to reuse the same models in a Typescript-based frontend application, by using a common Typescript package.

But there's a problem: the models created using the code first approach are heavily decorated with GraphQL related decorators. Those decorators are irrelevant in the frontend, negatively impacting performance.

Using the model shim

To solve this issue, NestJS provides a "shim" which allows you to replace the original decorators with inert code by using a `webpack` (or similar) configuration. To use this shim, configure an alias between the `@nestjs/graphql` package and the shim.

For example, for webpack this is resolved this way:

```
resolve: { // see: https://webpack.js.org/configuration/resolve/
  alias: {
      "@nestjs/graphql": path.resolve(__dirname,
"../node_modules/@nestjs/graphql/dist/extra/graphql-model-shim")
  }
}
```

> info **Hint** The TypeORM package has a similar shim that can be found here.