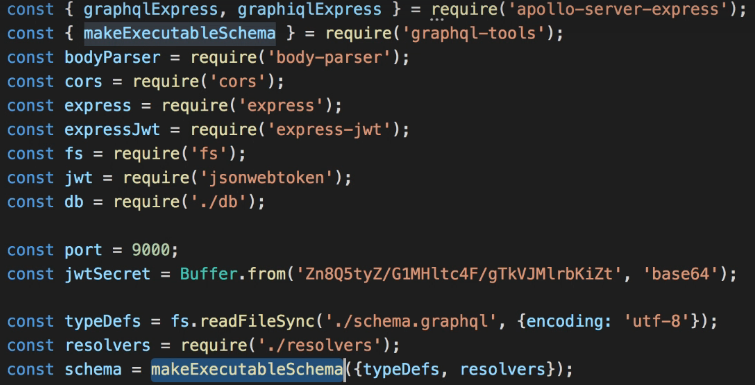
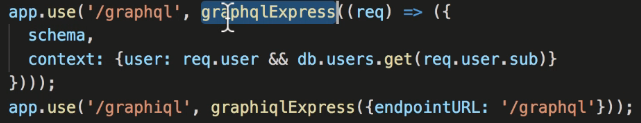
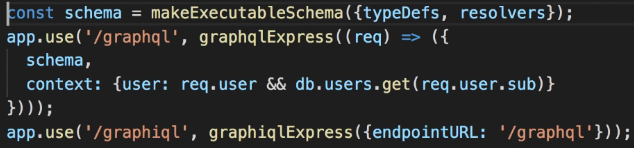
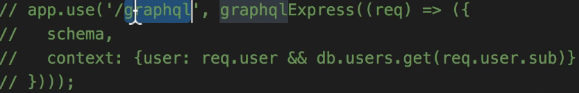
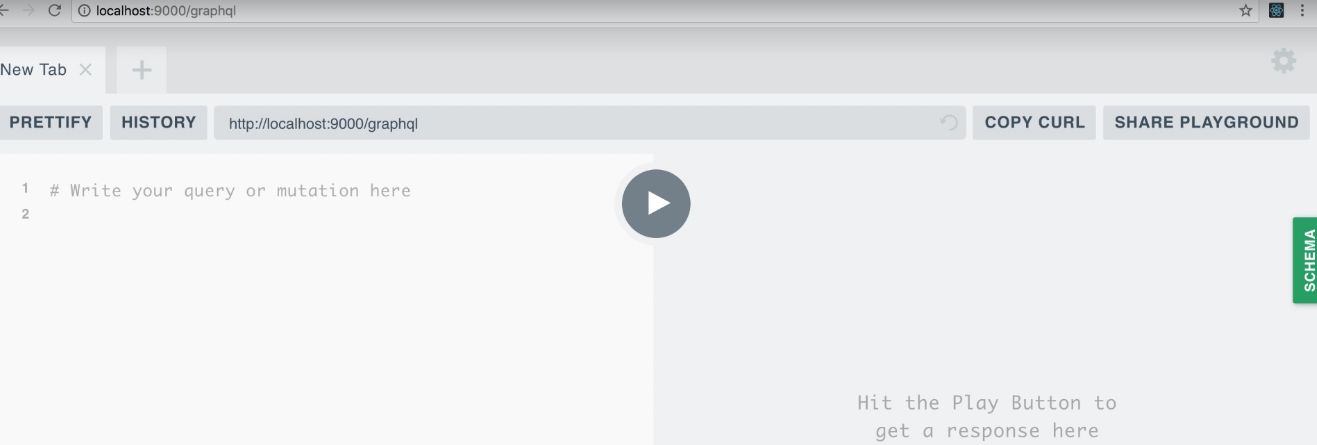
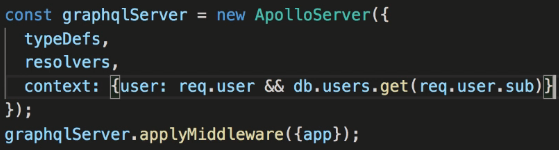
**Upgrading to Apollo Server 2.0**  
\* In July 2018 a new major version of Apollo Server was released.  
\* I’ll show you how to upgrade.  
\* There are a few changes required to the server code but they’re not fundamental differences.  
\* Remember that GraphQL is an **open specification** and Apollo is just 1 implementation of the standard so the changes in Apollo Server 2.0 don’t affect any of the core GraphQL concepts we learned about.  
**npm install apollo-server-express@latest**  
**npm start**  
=> we get an error.  
  
=> Since Apollo Server 2.0 is a major new version, it’s not backwards compatible with v1.  
=> So we’ll need to make some changes to our code.  
  
\* Let’s remind ourselves how our code works:  
  
  
\* What do we need to change to make it work with v2?  
\* **Basically we need to replace this part**:  
  
\* It’s just these 3 statements that no longer work with Apollo Server 2.0.  
\* **Let’s change the code**.  
  
\* This class now provides all the GraphQL middleware functionality.  
\* At the moment, the **typeDefs** variable is a string.  
=> With Apollo Server 2.0, we’ll need to import the gql function and use it to transform the string into a parsed object.  
  
  
\* So now we’re using that function in our server as well.  
\* The only difference is that here I’m calling it as a regular function instead of a Tag Function with a Template Literal.  
  
\* That’s pretty much all we need to get a basic GraphQL server running.  
\* Calling the applyMiddleware will effectively do the same thing as:  
  
\* We don’t need to specify the path anymore because it now uses “/graphql” by default.  
**npm start**  
\* Now it works all fine.  
\* If we got to localhost:9000/graphql, we see the GraphQL playground which is a bit like the GraphiQL tool we used before.  
  
=> This one has more features - supports multiple tabs, configuring HTTP Headers and so on.  
\* So Apollo Server 2.0 includes GraphQL Playground by default instead of GraphiQL.  
\* **You can disable the Playground**:  
  
\* **You should disable the Playground in production**.  
\* **We still need to set the context**.  
**context**  
 **=> Here we’re using req variable which is currenyl not defined anywhere.**  
=> With Apollo Server 2.0, we can pass a function as the “context” value.  
  
\* **Now everything works again including our authentication**.