

# Session 2: Building a GraphQL Backend

## Tech Stack

- Node.js + Apollo Server (JS)
- OR Python + Graphene (alt. track)

We'll build a simple blog API:

- Posts
- Authors



## Schema Definition (SDL)

```
type Post {  
  id: ID!  
  title: String!  
  content: String  
}  
  
type Query {  
  posts: [Post]  
  post(id: ID!): Post  
}  
  
type Mutation {  
  createPost(title: String!, content: String): Post  
}
```

## ⚙️ Apollo Server (Node.js Example)

```
const { ApolloServer, gql } = require('apollo-server')

const typeDefs = gql`...`
const resolvers = { ... }

const server = new ApolloServer({ typeDefs, resolvers })
server.listen().then(({ url }) => {
  console.log(`🚀 Server ready at ${url}`)
})
```

[🔗 https://www.apollographql.com/docs/apollo-server](https://www.apollographql.com/docs/apollo-server)

## Try it Out in GraphQL Playground

Use sample queries and mutations:

```
query {  
  posts {  
    id  
    title  
  }  
}
```

```
mutation {  
  createPost(title: "Hello", content: "World") {  
    id  
    title  
  }  
}
```