## ApolloServer constructor



```
options object to configure the Apollo server instance being created

const server = new ApolloServer ({typeDefs, resolvers});

promise that contains instance of the http server created

string representation of the GraphQL API schema in the Schema Definition Language created with the gal tag
```

## Create GraphQL API with Apollo Server

```
import ApolloServer constructor and gal tag from 'apollo-server'
import { ApolloServer, gql } from "apollo-server";
interface Superhero {
  id: string;
  title: string;
const superheroes: Superhero[] = [
  { id: "001", title: "Spider-Man" },
  { id: "002", title: "Iron Man" },
  { id: "003", title: "Black Panther" }
const typeDefs = gql`
  type Superhero {
    id: String!
    title: String!
                                                                               create the GraphQL schema
                                                                              document node with the gal tag
  type Query {
    superheroes: [Superhero!]!
const resolvers = {
  Query: {
                                                                              declare resolvers map for how
    superheroes: (): Superhero[] => superheroes,
                                                                              fields in the GraphQL API are to
                                                                                     be resolved
const server = new ApolloServer({ typeDefs, resolvers });
                                                                              create the ApolloServer instance
server.listen().then(({ url }) => {
  console.log(`Server is running at ${url}`);
                                                                                   start the server
});
```