Serve Static

In order to serve static content like a Single Page Application (SPA) we can use the `ServeStaticModule` from the `@nestjs/serve-static` package.

Installation

First we need to install the required package:

```
$ npm install --save @nestjs/serve-static
```

Bootstrap

Once the installation process is done, we can import the `ServeStaticModule` into the root `AppModule` and configure it by passing in a configuration object to the `forRoot()` method.

```
import { Module } from '@nestjs/common';
import { AppController } from './app.controller';
import { AppService } from './app.service';
import { ServeStaticModule } from '@nestjs/serve-static';
import { join } from 'path';

@Module({
  imports: [
    ServeStaticModule.forRoot({
      rootPath: join(__dirname, '..', 'client'),
    }),
  ],
  controllers: [AppController],
  providers: [AppService],
})
export class AppModule {}
```

With this in place, build the static website and place its content in the location specified by the `rootPath` property.

Configuration

ServeStaticModule can be configured with a variety of options to customize its behavior. You can set the path to render your static app, specify excluded paths, enable or disable setting Cache-Control response header, etc. See the full list of options here.

> warning **Notice** The default `renderPath` of the Static App is `*` (all paths), and the module will send "index.html" files in response. It lets you create Client-Side routing for your SPA. Paths, specified in your controllers will fallback to the server. You can change this behavior setting `serveRoot`, `renderPath` combining them with other options.

Example

A working example is available here.