

Security

To define which security mechanisms should be used for a specific operation, use the `@ApiSecurity()` decorator.

```
@ApiSecurity('basic')
@Controller('cats')
export class CatsController {}
```

Before you run your application, remember to add the security definition to your base document using `DocumentBuilder`:

```
const options = new DocumentBuilder().addSecurity('basic', {
  type: 'http',
  scheme: 'basic',
});
```

Some of the most popular authentication techniques are built-in (e.g., `basic` and `bearer`) and therefore you don't have to define security mechanisms manually as shown above.

Basic authentication

To enable basic authentication, use `@ApiBasicAuth()`.

```
@ApiBasicAuth()
@Controller('cats')
export class CatsController {}
```

Before you run your application, remember to add the security definition to your base document using `DocumentBuilder`:

```
const options = new DocumentBuilder().addBasicAuth();
```

Bearer authentication

To enable bearer authentication, use `@ApiBearerAuth()`.

```
@ApiBearerAuth()
@Controller('cats')
export class CatsController {}
```

Before you run your application, remember to add the security definition to your base document using `DocumentBuilder`:

```
const options = new DocumentBuilder().addBearerAuth();
```

OAuth2 authentication

To enable OAuth2, use `@ApiOAuth2()`.

```
@ApiOAuth2(['pets:write'])  
@Controller('cats')  
export class CatsController {}
```

Before you run your application, remember to add the security definition to your base document using `DocumentBuilder`:

```
const options = new DocumentBuilder().addOAuth2();
```

Cookie authentication

To enable cookie authentication, use `@ApiCookieAuth()`.

```
@ApiCookieAuth()  
@Controller('cats')  
export class CatsController {}
```

Before you run your application, remember to add the security definition to your base document using `DocumentBuilder`:

```
const options = new DocumentBuilder().addCookieAuth('optional-session-id');
```