Module 2-15

Authentication

Client: Review of RestTemplate methods

We have previously seen various ways of consuming an API using a RestTemplate object:

getForObject, postForObject, put, and delete

Let’s introduce one more, **exchange**, which can do all of the above request methods as well as bake in the token into the request.

Client: The exchange method

Here is the syntax of the exchange:

```
restTemplate.exchange(
URL of the API endpoint,
Request Method,
A HttpEntity object,
Reflective Call).getBody()
```

We chain the **.getBody()** method to properly deserialize into the class specified by the Reflective Call.

Client: The exchange method example

Here is an example call:

```
HttpHeaders headers = new HttpHeaders();
headers.setBearerAuth(AUTH_TOKEN);
HttpEntity entity = new HttpEntity<>(headers);
restTemplate.exchange(BASE_URL, HttpMethod.GET, entity, Location.class)
```

- AUTH_TOKEN is a string containing the JWT token.
- Note how it's added to a HttpHeaders object.
- … the rest should look familiar.

Client: The exchange method

Reasons to use .exchange:

- More generic, works for all request types
- Allows us to easily include the entity data for all requests

Let’s practice using the exchange method

Authorization vs Authentication

Authorization: Give a user permission to access the system.

Authentication: Verify that a user is able to access the system (a.k.a. - they are who they claim to be)

Authentication Methods

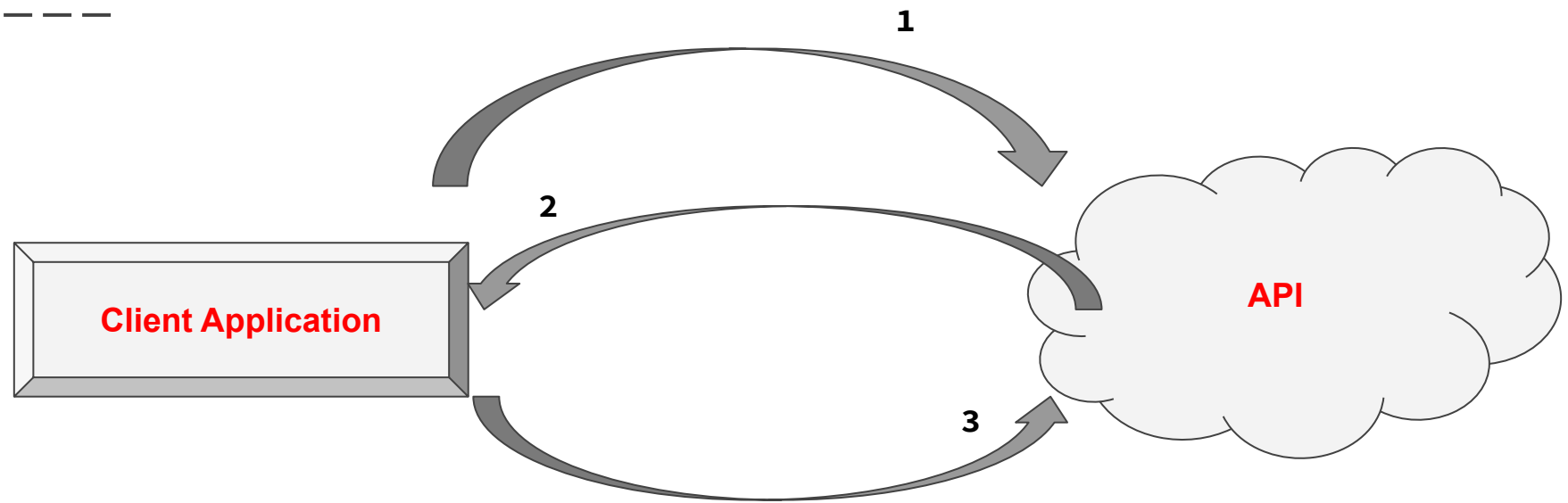
Various methods are used to authenticate a user, some of these are used together:

- Username and password
- RSA Token
- Biometrics (Facial Recognition, Fingerprint Scanner)
- Validation codes sent through SMS or to the user’s emails.

JWT Tokens

- A token is used to identify a user in the system. A valid token indicates that the user is authenticated.
- Tokens contain a claim in its payload (token body), which provides information about the user - i.e. username and role.

Token Workflow



1. A post request is made to the API containing credentials.
2. If the credentials are correct, a token is sent back to the client.
3. The client includes the token in any subsequent request.

JWT Token Demo

Generating Tokens on the Server

Setting up the infrastructure to generate tokens is beyond the scope of this course,the code to do this will be provided for you.

@PreAuthorize annotation examples

SPEL (Spring Expression Language) is used to fine tune how endpoints are secured. Here are some examples:

- **@PreAuthorize("isAuthenticated()")**: When defined at the class level, indicates that only authenticated users may access the endpoint.
  - **@PreAuthorize("permitAll")**: When defined at the method level, allows access to the endpoint to any user (even un-authenticated ones, and overrides the class level @PreAuthorize.
  - **@PreAuthorize("hasRole('ROLE')")**: When defined at the method level, specifies that only users with a certain role can access the endpoint.

Possible 4XX errors Thrown

- **401 Unauthorized**: You are trying to consume the API without a valid token.
- **403 Forbidden**: You were authenticated, but still not allowed to view the resource (i.e. the resource requires an admin vs a regular user).

Server Code Walkthrough