CLOUD COMPUTING SYSTEMS

Lab 6

João Resende, Nuno Preguiça

(jresende_at_fct.un.pt, nuno.preguica_at_fct.unl.pt)

GOAL

In the end of this lab you should be able to:

- Know how to address access control.

Approach to implement access control

Step 1: user must log in to the system;

Step 2: when executing operations, check if the user is allowed to execute the operation.

AUTHENTICATION ENDPOINT

Step 1: user must log in in the system;

- Endpoint **/user/auth** receives a JSON object with properties **user** and **pwd;**
- Method checks if authentication is correct. If it is, create a cookie with a unique identifier, and store on Redis a session object with the **user** under a key with the generated unique identifier.

AUTHENTICATION ENDPOINT (2)

Note: This code is given as an example. You would have to create classes **Login** and **Session**. **RedisLayer** would be a class that uses RedisCache to store objetcs (User, Auction, …, Session).

```
@POST
@Path("/auth")
@Consumes(MediaType.APPLICATION_JSON)
public Response auth(Login user) {
  boolean pwdOk = false;

  // Check pwd

  if( pwdOk) {
    String uid = UUID.randomUUID().toString();
    NewCookie cookie = new NewCookie.Builder("scc:session")
                                    .value(uid)
                                    .path("/")
                                    .comment("sessionid")
                                    .maxAge(3600)
                                    .secure(false)
                                    .httpOnly(true)
                                    .build();
    RedisLayer.getInstance().putSession( new Session( uid, user.getUser()));
    return Response.ok().cookie(cookie).build();
  } else
    throw new NotAuthorizedException("Incorrect login");
}
```

ACCESS CONTROL

Step 2: when executing operations, check if the user is allowed to execute the operation.

- In methods that require access control, use the cookie to know which user is calling the method.

ACCESS CONTROL (2)

```
@POST
@Path("/")
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public Auction postAuction(@CookieParam("scc:session") Cookie session,
                           Auction auction) {
  try {
    // Check that auction is correct

    checkCookieUser(session, auction.getOwner());

    // Code to create auction

  } catch( WebApplicationException e) {
    throw e;
  } catch( Exception e) {
    throw new InternalServerErrorException( e);
  }
}
```

ACCESS CONTROL (3)

```
/**
 * Throws exception if not appropriate user for operation on Auction
 */
public Session checkCookieUser(Cookie session, String id)
                        throws NotAuthorizedException {
  if (session == null || session.getValue() == null)
    throw new NotAuthorizedException("No session initialized");
  Session s;
  try {
    s = RedisLayer.getInstance().getSession(session.getValue());
  } catch (CacheException e) {
    throw new NotAuthorizedException("No valid session initialized");
  }
  if (s == null || s.getUser() == null || s.getUser().length() == 0)
    throw new NotAuthorizedException("No valid session initialized");
  if (!s.getUser().equals(id) && !s.getUser().equals("admin"))
    throw new NotAuthorizedException("Invalid user : " + s.getUser());
  return s;
}
```

TODO

Add access control to your code.