Class 14: Sessions and Auth

Resources

- [Bcrypt]

Learning Objectives

- Acess and modify session data
- Model User information to store in a database
- Use Bcrpyt to hash passwords
- Store hashed passwords securely in a database

Big Idea

- Set `loggedin` session attribute to `true` when someone logs in
- Set `loggedin` session attribute to `false` when someone logs in
- Check `loggedin` session attribute when someone is trying to access a protected resource.
    - If `loggedin` is `true` render the secret resource
    - If `loggedin` is `false` return a "denied" page template

Accessing Session Data

```
HttpSession session = request.getSession();
session.setAttribute("username", username);
```

```
@GetMapping
public String homepage(HttpServletRequest request) {
  HttpSession session = request.getSession();
  String username = (String) session.getAttribute("username");
  System.out.println(username);
    return "index";
}
```

Connecting to a Database

```
try {
    Class.forName("org.postgresql.Driver");
    String url = "jdbc:postgresql://localhost:5432/javaauth";

    try {
        Connection mConn = DriverManager.getConnection(url);
        ResultSet results = mConn.createStatement().executeQuery("SELECT * FROM us
        while (results.next()) {
            int id = results.getInt("id");
            String name = results.getString("username");
        }
    } catch (SQLException e) {
        e.printStackTrace();
        System.out.println("DB error.");
    }
} catch (ClassNotFoundException e) {
    e.printStackTrace();
    System.out.println("Postgres driver not configured correctly.");
}
```

Seeding a Database

```
try {
    Class.forName("org.postgresql.Driver");
    String url = "jdbc:postgresql://localhost:5432/javaauth";

    try {
        mConn = DriverManager.getConnection(url);
        ResultSet results = mConn.createStatement().executeQuery("SELECT * FROM us
        while (results.next()) {
            int id = results.getInt("id");
            String name = results.getString("username");
        }
    } catch (SQLException e) {
        e.printStackTrace();
        System.out.println("DB error.");
    }
} catch (ClassNotFoundException e) {
    e.printStackTrace();
    System.out.println("Postgres driver not configured correctly.");
}
```

Hashing passwords

```
BCrypt.
String password = "password";
String hashed = BCrypt.hashpw(password, BCrypt.gensalt(12));
```

```
String attempt = "password";
String hash = ;
boolean result = BCrypt.checkpw(attempt, hash);
```

Protecting Routes

```java
@RequestMapping("/*")
public ModelAndView handlePrivateRequests(HttpServletRequest request) {
    String servlet = request.getServletPath();
    ModelAndView mv = new ModelAndView();

    HttpSession session = request.getSession();
    String username = (String) session.getAttribute("username");
    if (username != null) {
        mv.setViewName("secret");
    } else {
        mv.setViewName("accessdenied");
    }

    return mv;
}
```