Day 5
Spring framework

Lecturer: Msc. Minh Tan Le

Agenda

I. Dependency inversion using Spring Framework

II. Sessions & cookies

Demo: Use Spring framework

6. Convert project to Gradle.
7. Install Spring Core.
8. Create context XML configurations.

XML Bean Context format

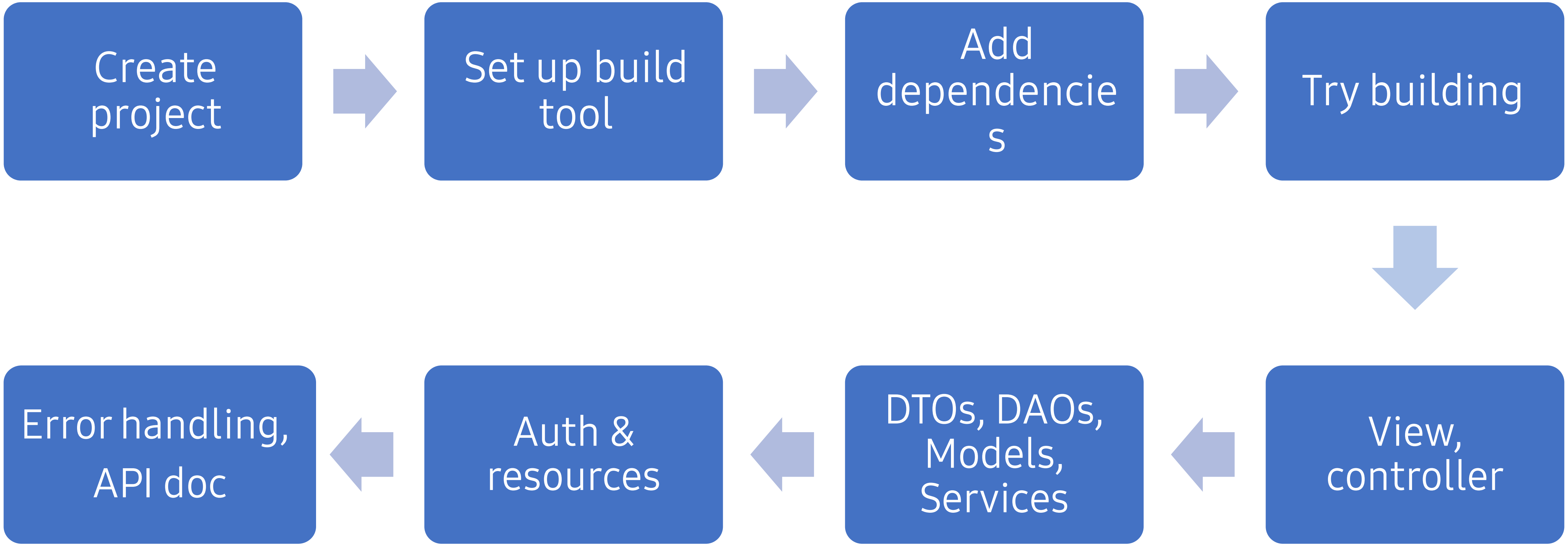
```
<beans xmlns = "http://www.springframework.org/schema/beans"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context = "http://www.springframework.org/schema/context"
  xsi:schemaLocation = "http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans.xsd
  http://www.springframework.org/schema/context
  http://www.springframework.org/schema/context/spring-context.xsd">

  <context:annotation-config/>

   ...

</beans>
```

Conclusions

- Spring Framework can be used in many type of apps.
- Use of XML to define context (@Bean is an alternative).
- Use of annotations: @Component, @Autowired
- Gradle configs for any apps.

Learn more at: https://docs.spring.io/spring-framework/reference/core/beans/java/bean-annotation.html

III. Spring MVC for web



Create project

Set up build tool

Add dependencies

Try building

View, controller

DTOs, DAOs, Models, Services

Auth & resources

Error handling, API doc

Instruction

1. Same as Servlet
   - File -> New -> Other -> Dynamic Web Project
2. Generate Deployment Descriptor Stub (web.xml)
3. Convert to Maven project
4. Init gradle

Instruction

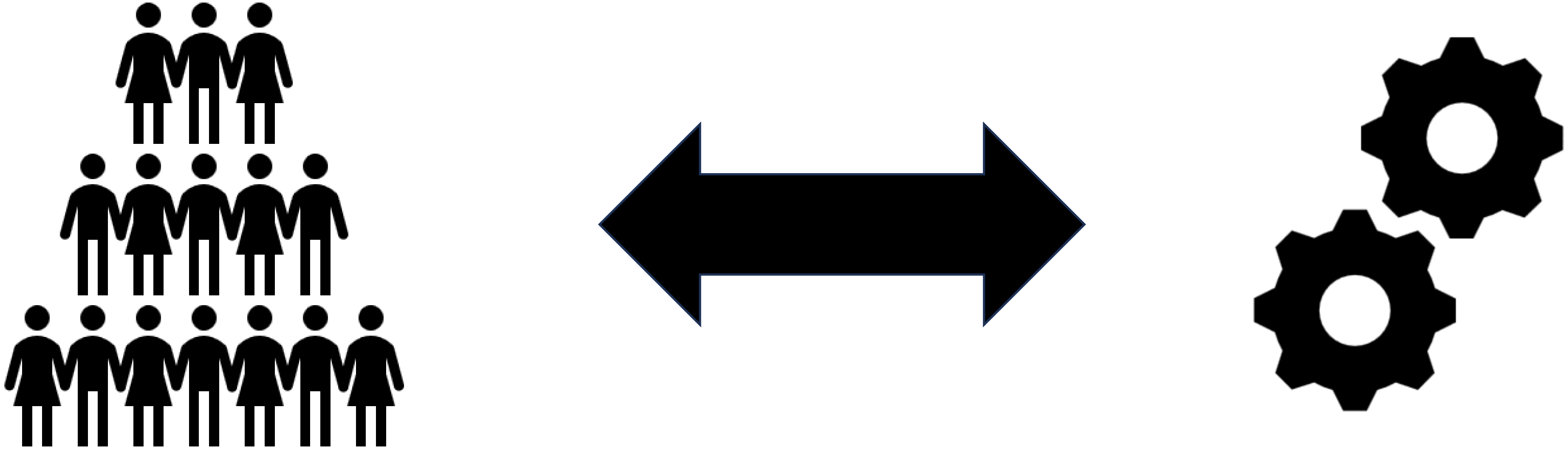
5. Install packages
   - javax.servlet:javax.servlet-api
   - javax.servlet:jstl
   - org.springframework:spring-webmvc
6. Build using gradle (gradle refresh)

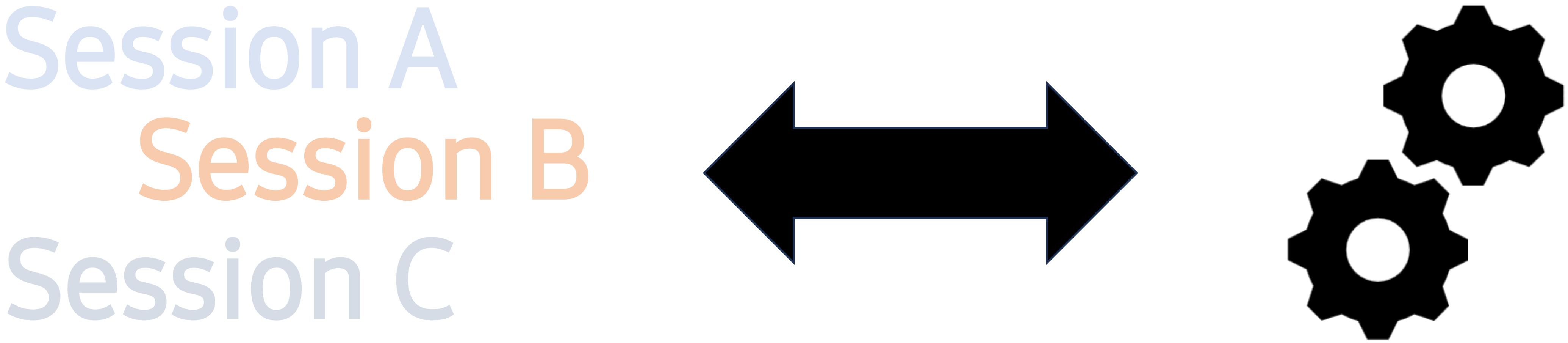
Built-in beans

No	Class	Properties	Description
1	org.springframework.web.servlet.view.InternalResourceViewResolver	prefix suffix	Auto map view to name as String returned by controller methods.

Built-in annotations

No	Class	Suggested parameters	Description
1	org.springframework.stereotype.Controller		Determine Spring controller class.
2	org.springframework.web.bind.annotation.RequestMapping	value, method	Map controller method to path (value) & method. Example: value="/", method=RequestMethod.GET





Session A

Session B

Session C

Session A

Session B

Session C

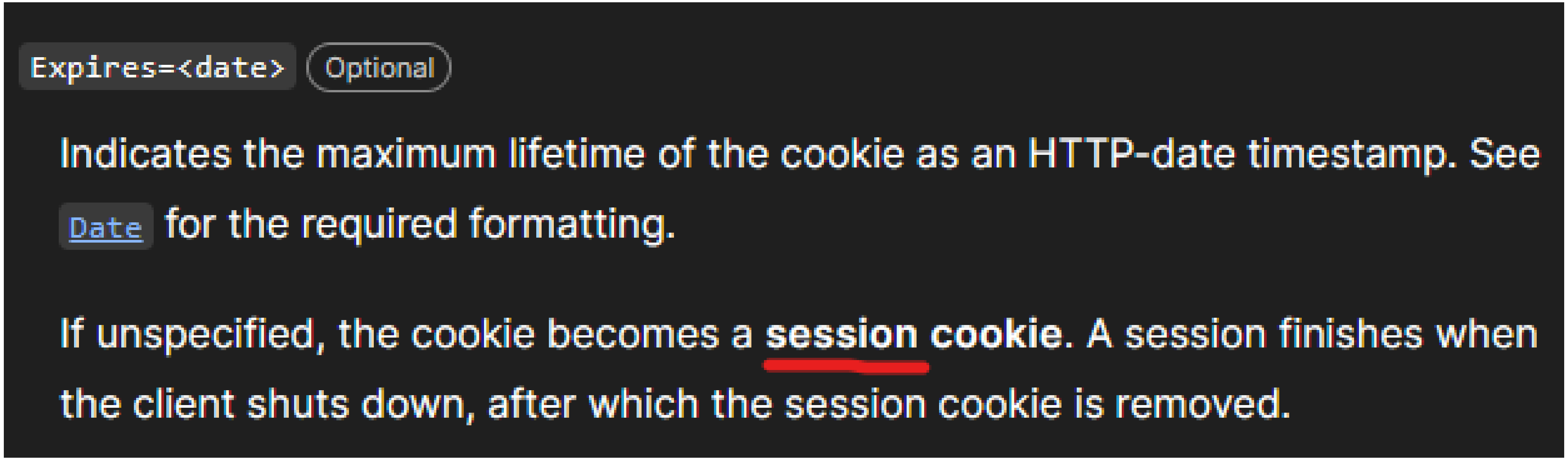
=

Client-side

A period of time after accessing to website and before closing the process.

Server-side

Server-defined session, including expires/max-age



`Expires=<date>` Optional

Indicates the maximum lifetime of the cookie as an HTTP-date timestamp. See `Date` for the required formatting.

If unspecified, the cookie becomes a **session cookie**. A session finishes when the client shuts down, after which the session cookie is removed.

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie#expiresdate

Determine sessions. Can we use...

Connection?

Determine sessions. Can we use…

IP?

Determine sessions. Can we use...

ID (token)?

But what are tokens?

- Values being used to:
  - Identify sessions
  - Match one or more use cases
- Tokens are:
  - Temporary
  - Random
  - Unique

We often use with cookies

- Cookies are temporary data received from specific website.
- Each website can link to many cookies.
- A pack of cookies is sometimes called a cookie jar.

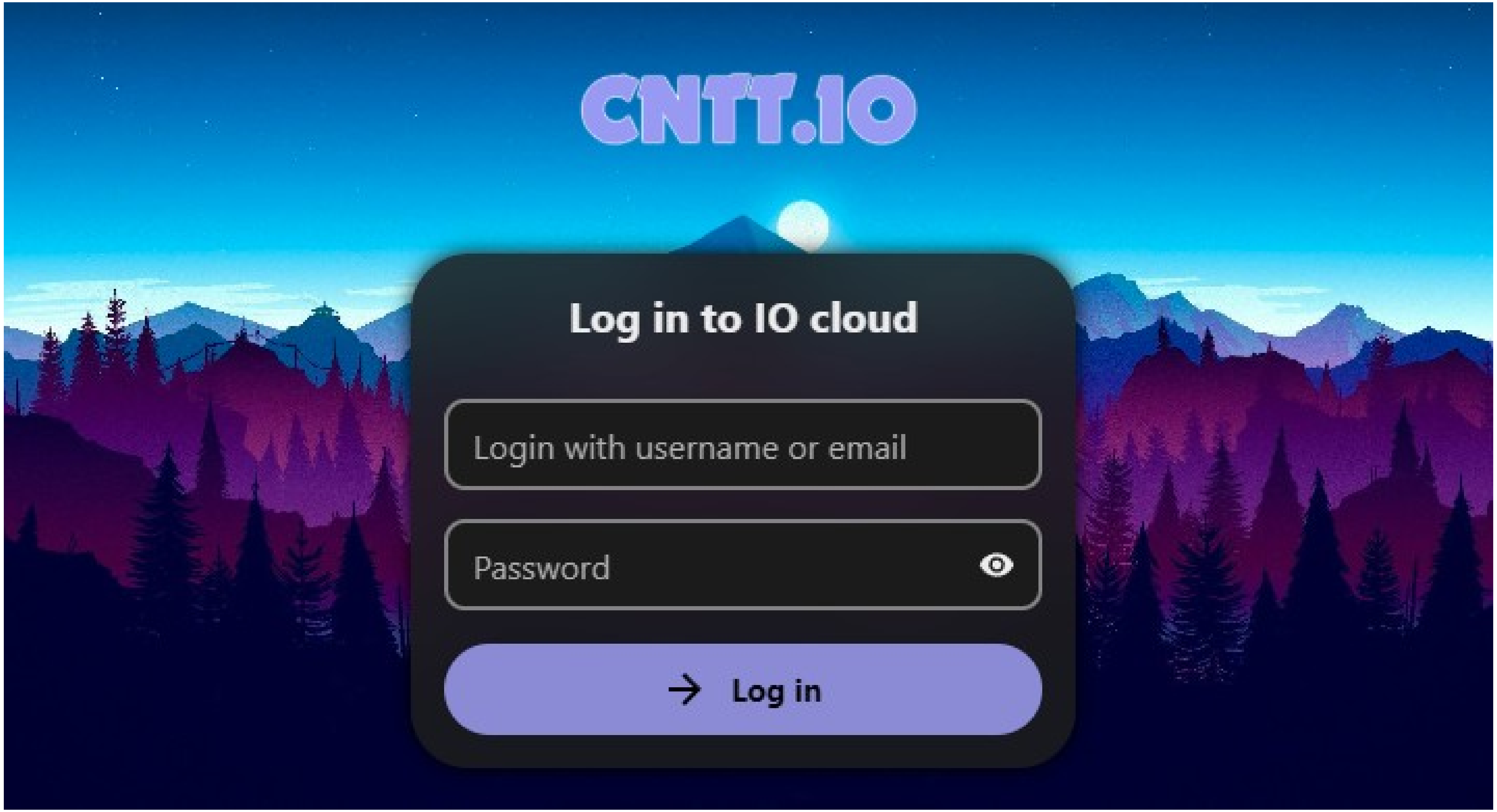
What’s inside a cookie?

- Name
- Value
- Domain
- Path
- (Expires/Max-age)
- (HttpOnly)
- (Secure)

…

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie

Discussion
Should token be saved in DB?


CNTT.IO
Log in to IO cloud
Login with username or email
Password
→ Log in

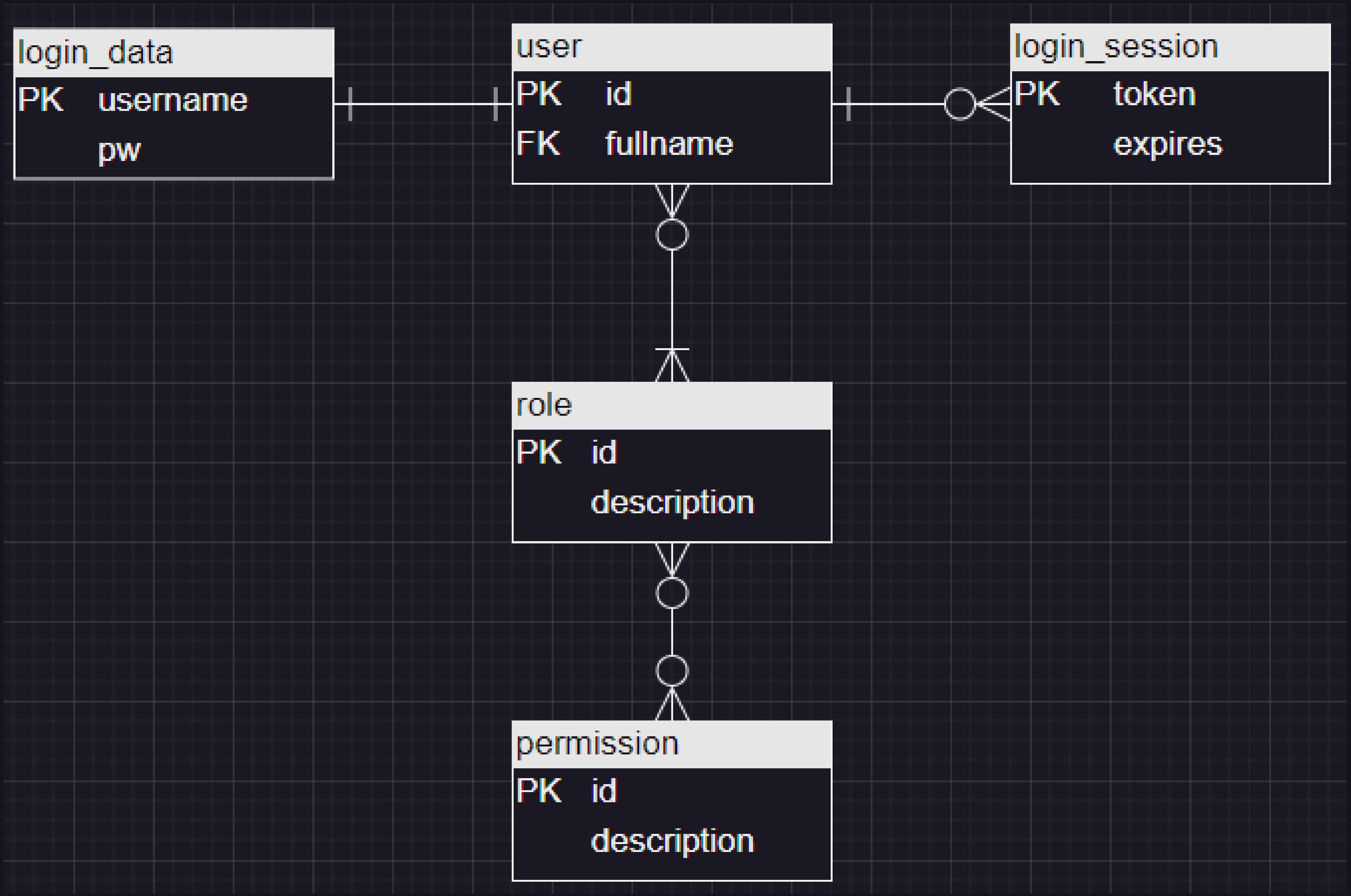
Demo: Build auth functions

1. A login form with username & password.
2. After a form is submitted, show a page with result (successfully login or failed login).
3. Save a cookie to client to remember session.
4. Add logout feature.

Homework

1. Install DBeaver.
2. Install PostgreSQL.
3. Create schema: db<studentID> (example: db15110123)
4. Create DB table (in schema other than public!)
   - user
   - login_data
   - login_session
     - expires (timestamp)
   - roles
   - permission
   - granted_permission
   - assigned_role

Best practice: https://vertabelo.com/blog/user-authentication-module/



login_data

PK username

pw

user

PK id

FK fullname

login_session

PK token

expires

role

PK id

description

permission

PK id

description