Show-and-tell:
Browser test automation

Frank Lange
Leibniz-Institut für Pflanzenbiochemie

Agenda

1. Unit tests – why we should write tests
2. Architecture of CRIMSy
3. What we test in CRIMSy & why we need frontend tests
4. Browser tests with Selenium & Co.
5. Dockerize browser tests

Leibniz-Institut für Pflanzenbiochemie

Where is the code?

https://github.com/ipb-halle/Show-And-Tell_Browser-Test-Automation

Leibniz-Institut für Pflanzenbiochemie

Unit tests – why we should write tests

See

code/1_unit_tests/

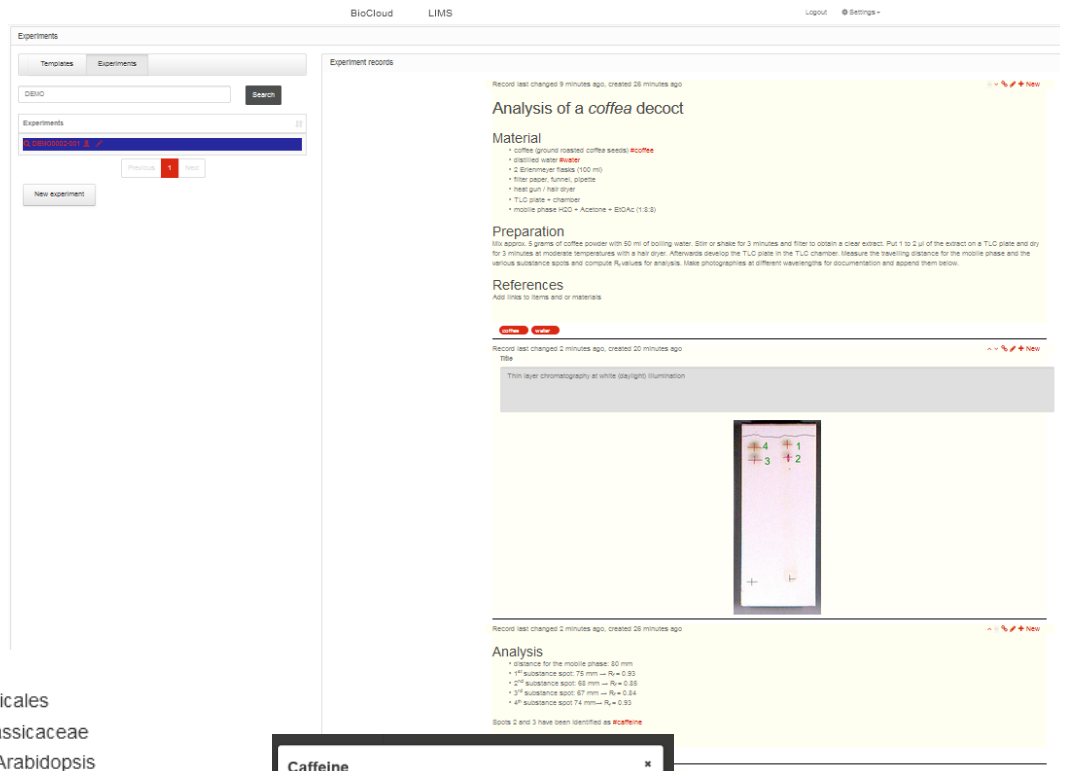
Leibniz-Institut für Pflanzenbiochemie

CRIMSy (**C**loud **R**esource & **I**nformation **M**anagement **Sy**stem)

https://crimsy.org

GitHub:
https://github.com/ipb-halle/CRIMSy

Test setup:
https://github.com/ipb-halle/CRIMSy/wiki/Test-setup



Record last changed 9 minutes ago, created 26 minutes ago

Analysis of a *coffea* decoct

Material

- coffee (ground roasted coffea seeds) #coffee
- distilled water #water
- 2 Erlenmeyer flasks (100 ml)
- filter paper, funnel, pipette
- heat gun / hair dryer
- TLC plate + chamber
- mobile phase H2O + Acetone + EtOAc (1:8:8)

Preparation

Mix approx. 5 grams of coffee powder with 50 ml of boiling water. Stir or shake for 3 minutes and filter to obtain a clear extract. Put 1 to 2 µl of the extract on a TLC plate and dry for 3 minutes at moderate temperatures with a hair dryer. Afterwards develop the TLC plate in the TLC chamber. Measure the travelling distance for the mobile phase and the various substance spots and compute R_f values for analysis. Make photographies at different wavelengths for documentation and append them below.

References

Add links to items and or materials

coffee water

Record last changed 2 minutes ago, created 20 minutes ago

Title

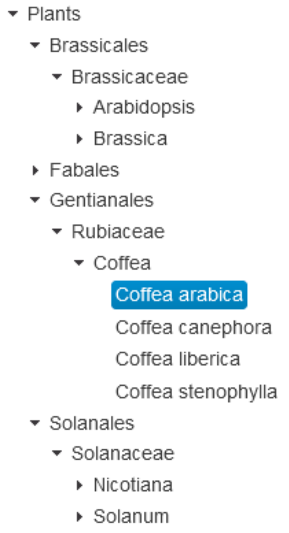
Thin layer chromatography at white (daylight) illumination

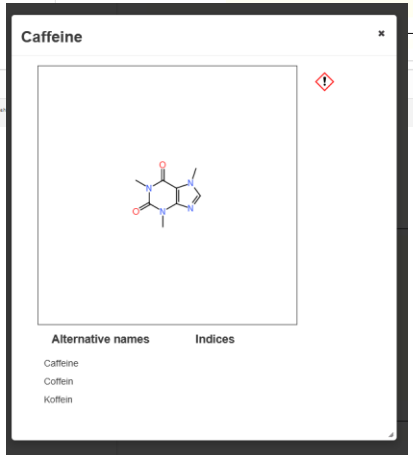
Record last changed 2 minutes ago, created 26 minutes ago

Analysis

- distance for the mobile phase: 80 mm
- 1st substance spot: 75 mm → R_f = 0.93
- 2nd substance spot: 68 mm → R_f = 0.85
- 3rd substance spot: 67 mm → R_f = 0.84
- 4th substance spot 74 mm→ R_f = 0.93

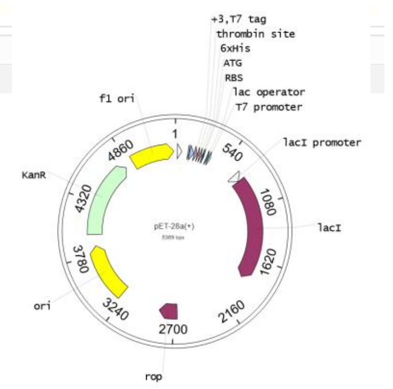
Spots 2 and 3 have been identified as #caffeine





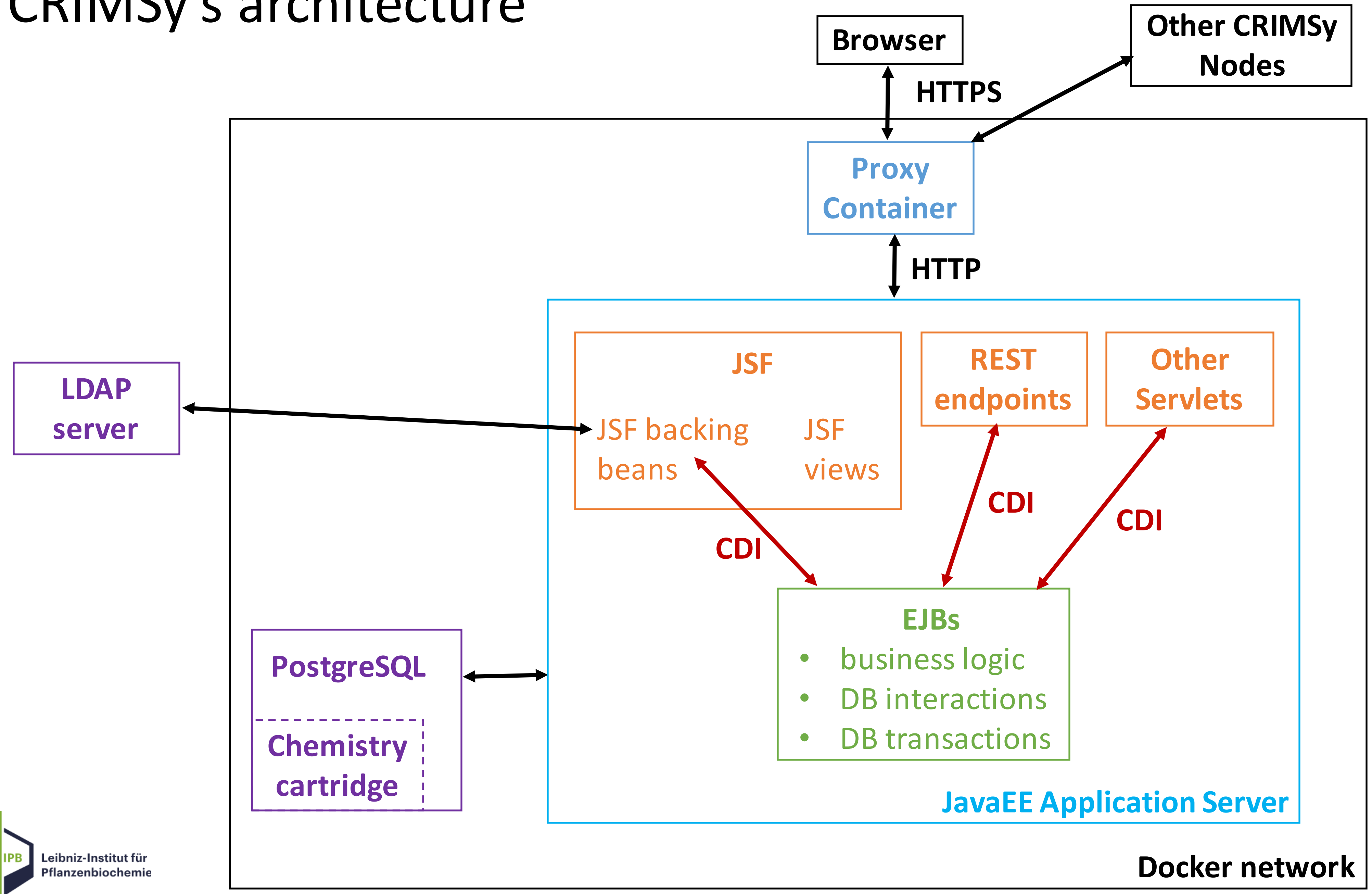
Caffeine

Alternative names	Indices
Caffeine	
Coffein	
Koffein	



Leibniz-Institut für Pflanzenbiochemie

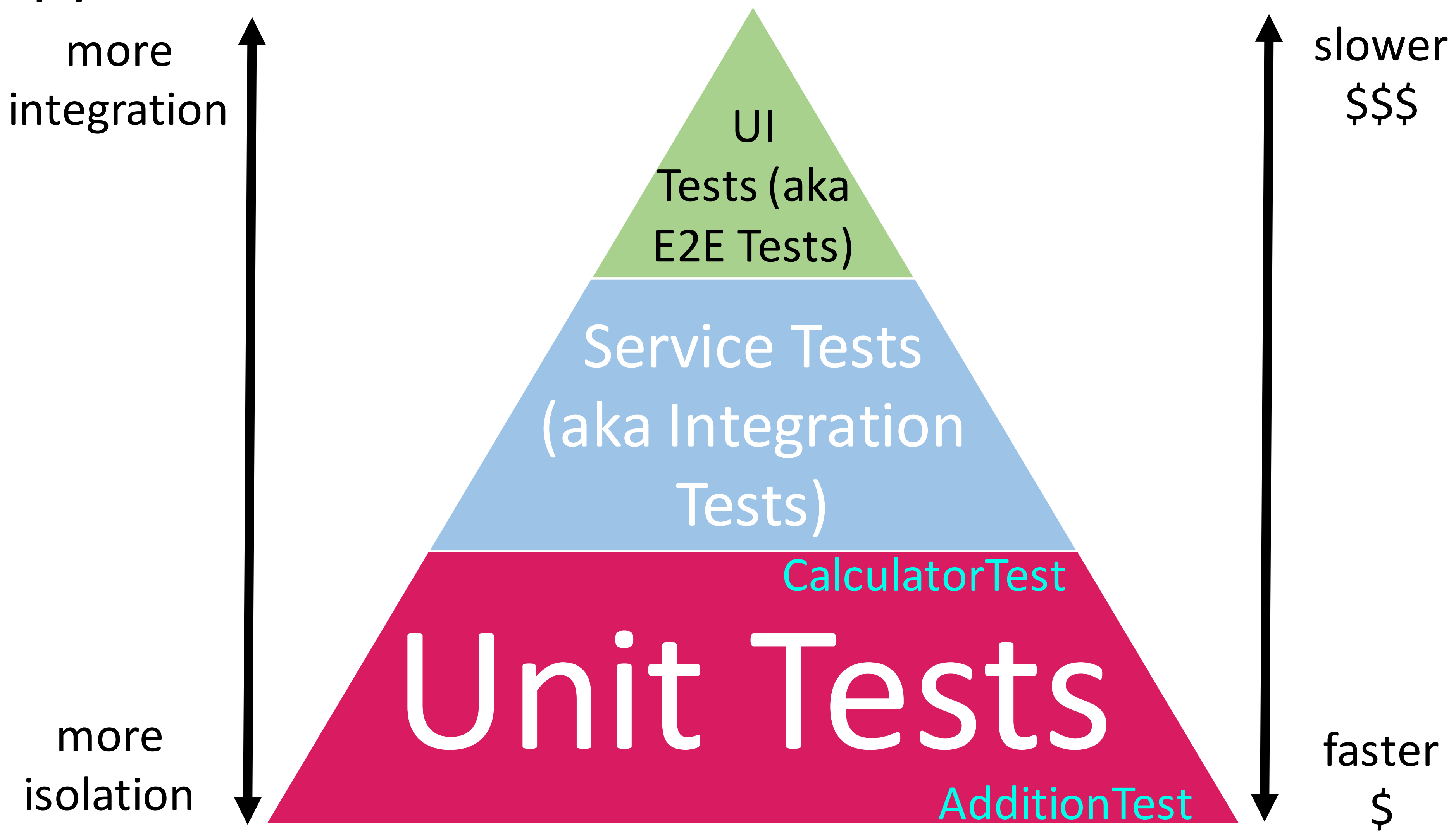
CRIMSy's architecture



JSF = Java Server Faces
EJB = Enterprise Java Beans
CDI = Context & Dependency Injection

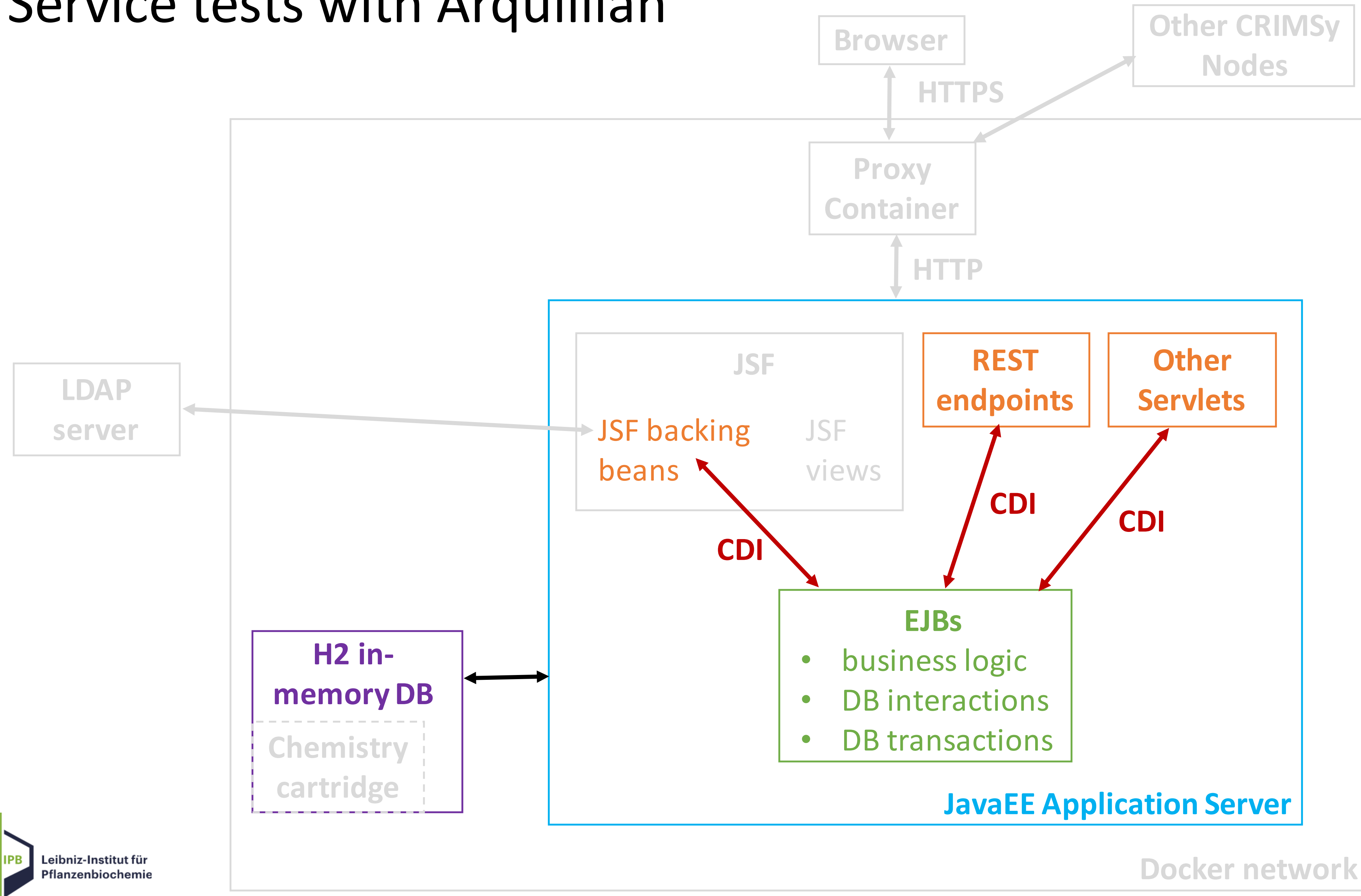
Leibniz-Institut für Pflanzenbiochemie

Test pyramid



https://martinfowler.com/articles/practical-test-pyramid.html

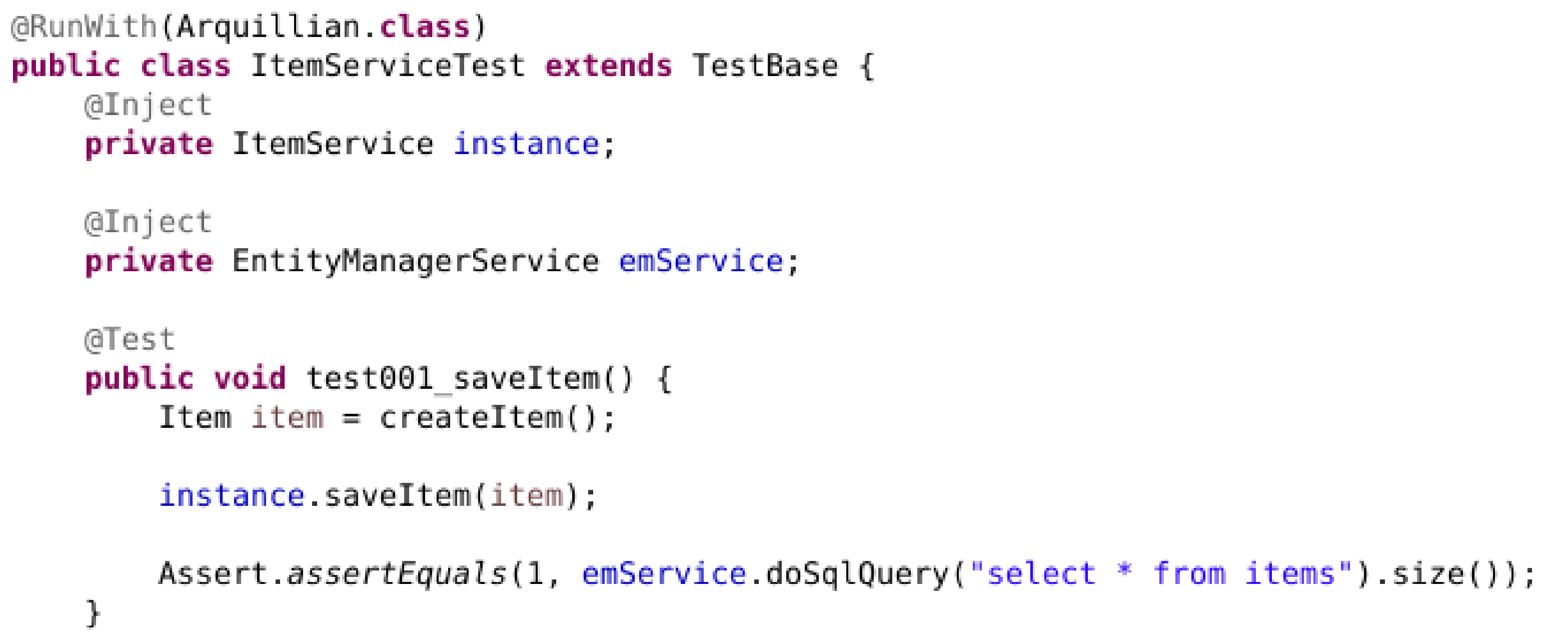
Service tests with Arquillian



JSF = Java Server Faces
EJB = Enterprise Java Beans
CDI = Context & Dependency Injection

Leibniz-Institut für Pflanzenbiochemie

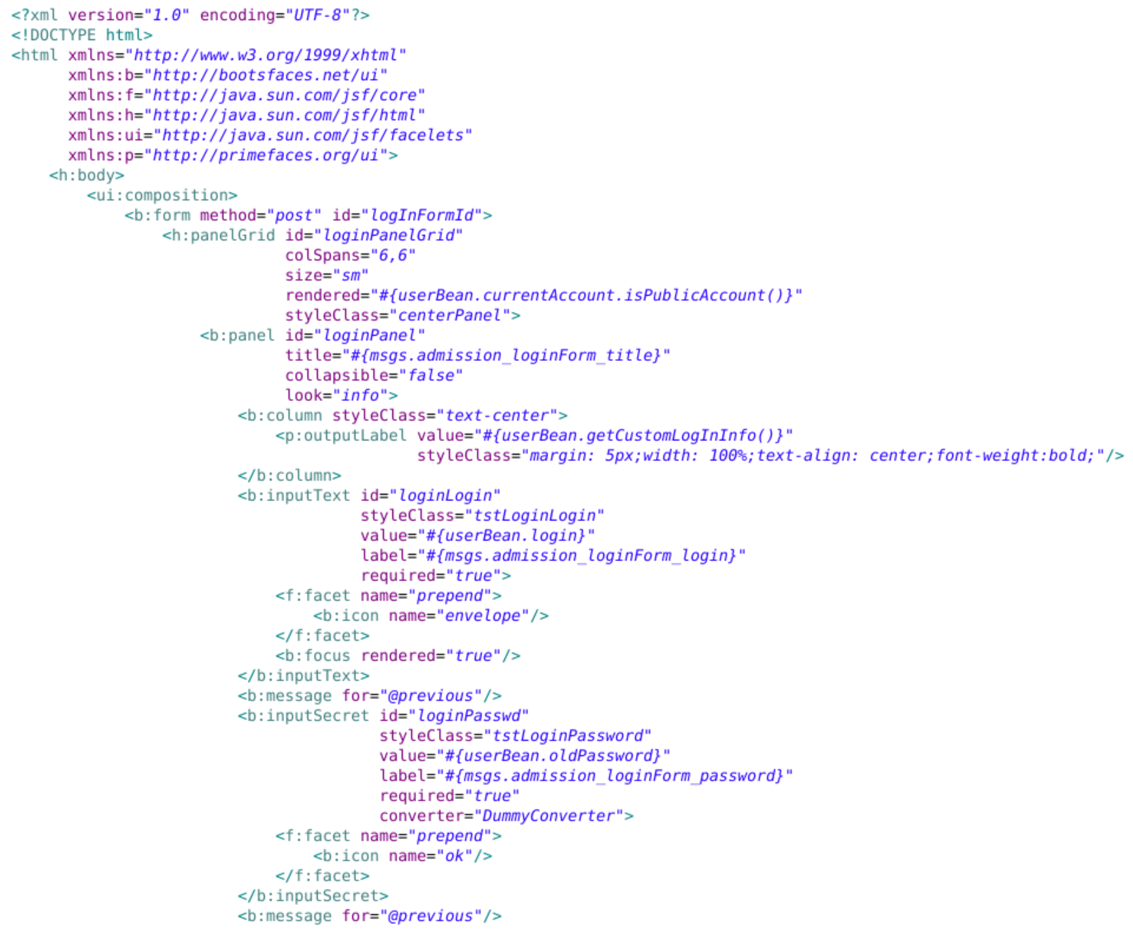
Service tests with Arquillian



```java
@RunWith(Arquillian.class)
public class ItemServiceTest extends TestBase {
    @Inject
    private ItemService instance;

    @Inject
    private EntityManagerService emService;

    @Test
    public void test001_saveItem() {
        Item item = createItem();

        instance.saveItem(item);

        Assert.assertEquals(1, emService.doSqlQuery("select * from items").size());
    }
```

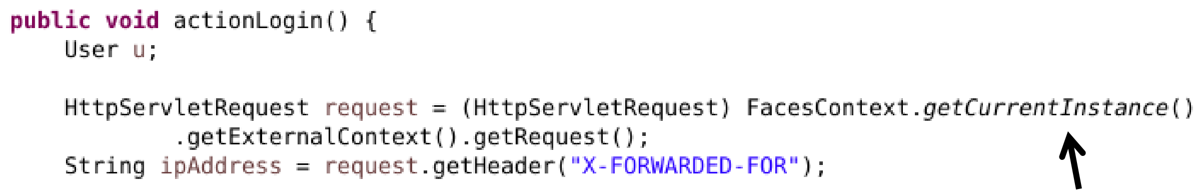
Testing JSF?

Facelet (JSF's templating language)



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:b="http://bootsfaces.net/ui"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">
    <h:body>
        <ui:composition>
            <b:form method="post" id="logInFormId">
                <h:panelGrid id="loginPanelGrid"
                             colSpans="6,6"
                             size="sm"
                             rendered="#{userBean.currentAccount.isPublicAccount()}"
                             styleClass="centerPanel">
                    <b:panel id="loginPanel"
                             title="#{msgs.admission_loginForm_title}"
                             collapsible="false"
                             look="info">
                        <b:column styleClass="text-center">
                            <p:outputLabel value="#{userBean.getCustomLogInInfo()}"
                                           styleClass="margin: 5px;width: 100%;text-align: center;font-weight:bold;"/>
                        </b:column>
                        <b:inputText id="loginLogin"
                                     styleClass="tstLoginLogin"
                                     value="#{userBean.login}"
                                     label="#{msgs.admission_loginForm_login}"
                                     required="true">
                            <f:facet name="prepend">
                                <b:icon name="envelope"/>
                            </f:facet>
                            <b:focus rendered="true"/>
                        </b:inputText>
                        <b:message for="@previous"/>
                        <b:inputSecret id="loginPasswd"
                                       styleClass="tstLoginPassword"
                                       value="#{userBean.oldPassword}"
                                       label="#{msgs.admission_loginForm_password}"
                                       required="true"
                                       converter="DummyConverter">
                            <f:facet name="prepend">
                                <b:icon name="ok"/>
                            </f:facet>
                        </b:inputSecret>
                        <b:message for="@previous"/>
```

JSF's internal objects

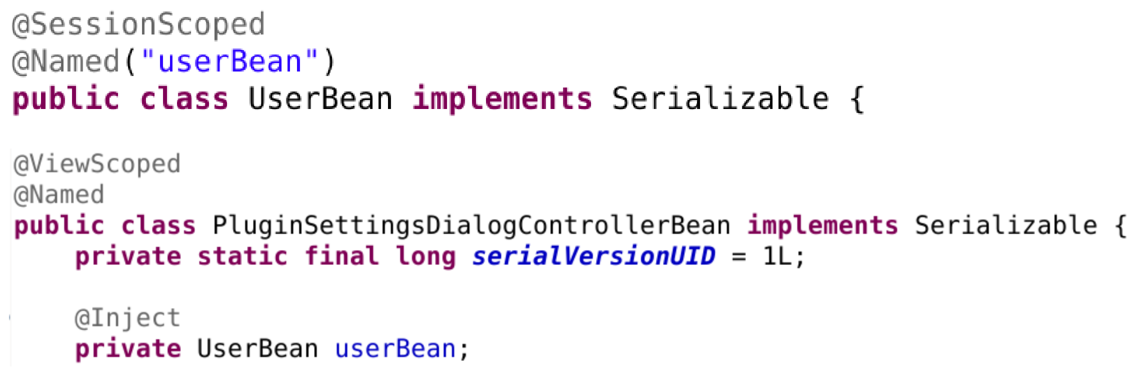


```
public void actionLogin() {
    User u;

    HttpServletRequest request = (HttpServletRequest) FacesContext.getCurrentInstance()
            .getExternalContext().getRequest();
    String ipAddress = request.getHeader("X-FORWARDED-FOR");
```

↑

== null

strange bean scopes
(tied to HTTP sessions/requests)

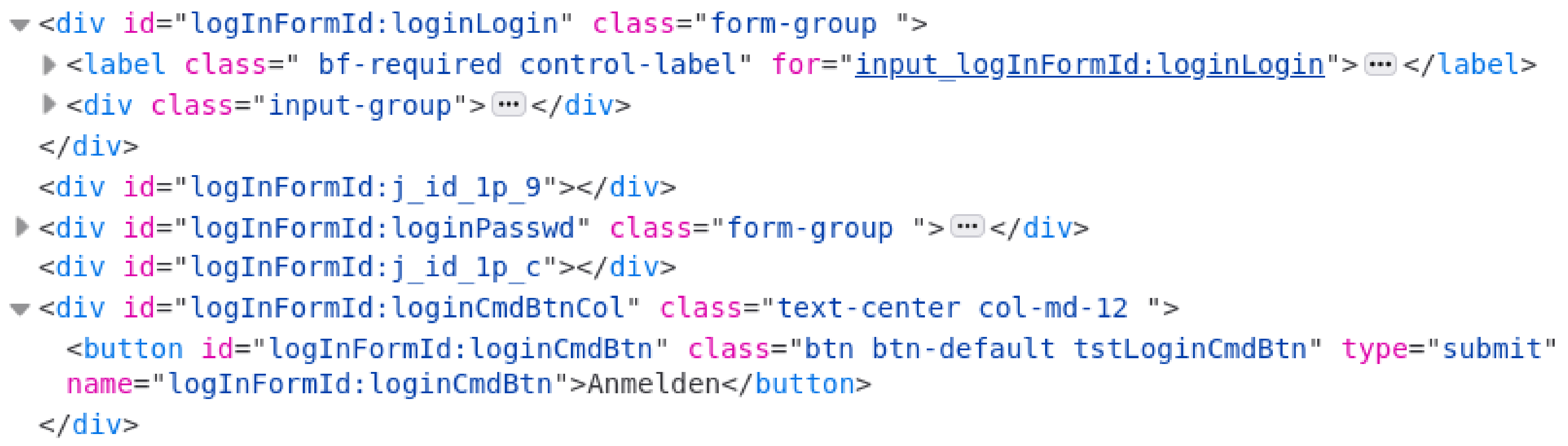


```
@SessionScoped
@Named("userBean")
public class UserBean implements Serializable {

@ViewScoped
@Named
public class PluginSettingsDialogControllerBean implements Serializable {
    private static final long serialVersionUID = 1L;

    @Inject
    private UserBean userBean;
```

(almost) untestable!

Testing JSF?

rendered HTML



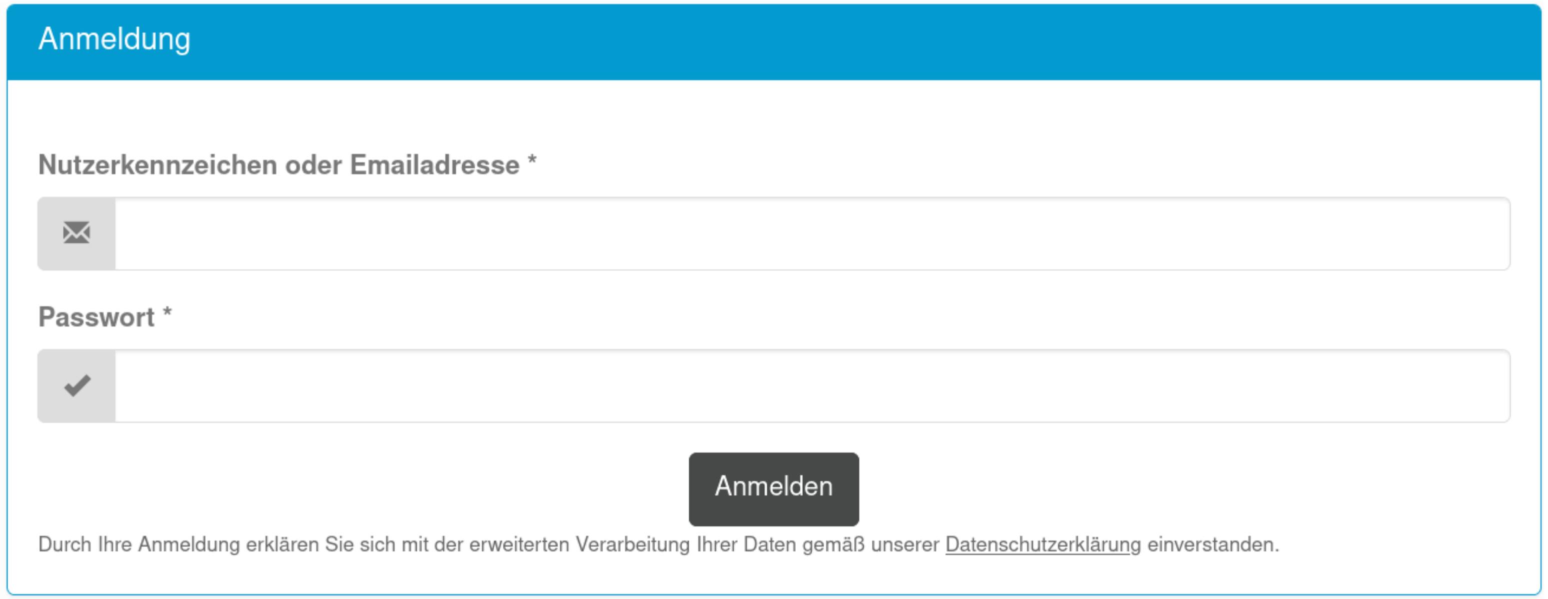
```
▼<div id="logInFormId:loginLogin" class="form-group ">
  ▶<label class=" bf-required control-label" for="input_logInFormId:loginLogin">⋯</label>
  ▶<div class="input-group">⋯</div>
 </div>
 <div id="logInFormId:j_id_1p_9"></div>
▶<div id="logInFormId:loginPasswd" class="form-group ">⋯</div>
 <div id="logInFormId:j_id_1p_c"></div>
▼<div id="logInFormId:loginCmdBtnCol" class="text-center col-md-12 ">
   <button id="logInFormId:loginCmdBtn" class="btn btn-default tstLoginCmdBtn" type="submit"
   name="logInFormId:loginCmdBtn">Anmelden</button>
 </div>
```

Testable?

Leibniz-Institut für Pflanzenbiochemie

Testing JSF?

Website rendered in the browser


Anmeldung

Nutzerkennzeichen oder Emailadresse *

Passwort *

Durch Ihre Anmeldung erklären Sie sich mit der erweiterten Verarbeitung Ihrer Daten gemäß unserer Datenschutzerklärung einverstanden.

Testable!

Leibniz-Institut für Pflanzenbiochemie

Browser tests with the Selenium WebDriver

ПЕРІОДИЧЕСКІЙ ЗАКОНЪ Д.И. МЕНДЕЛѢЕВА. 1869 г.
ТАБЛИЦА ИЗГОТОВЛЕННАЯ ПО УКАЗАНІЮ АВТОРА въ 1876 г.
78 Se

https://www.selenium.dev/documentation/getting_started/

Supports: Java, Python, C#, Ruby, JS, PHP, ...

Leibniz-Institut für Pflanzenbiochemie

Saint-Petersburg - Mendeleevsky center - Periodic table 2.JPG, CC-BY 3.0, Wiki user Christophe.Finot

Browser tests with the Selenium WebDriver

See

code/2_selenium/

Browser tests with Selenide

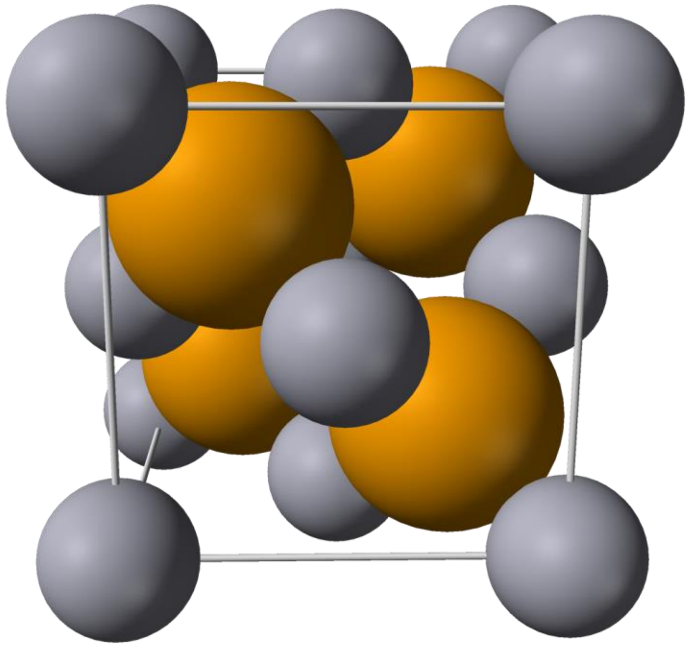
https://selenide.org/quick-start.html

Java only :(

Ports: https://github.com/selenide/selenide/wiki/Ports-of-Selenide (Python, JS, PHP, C#)

Claims:

- Concise fluent API for tests
- Ajax support for stable tests
- Powerful selectors
- Simple configuration



Leibniz-Institut für Pflanzenbiochemie

Wikimedia Commons: Mercury(II)-selenide-unit-cell-3D-ionic.png

Browser tests with Selenide

See

code/3_selenide/

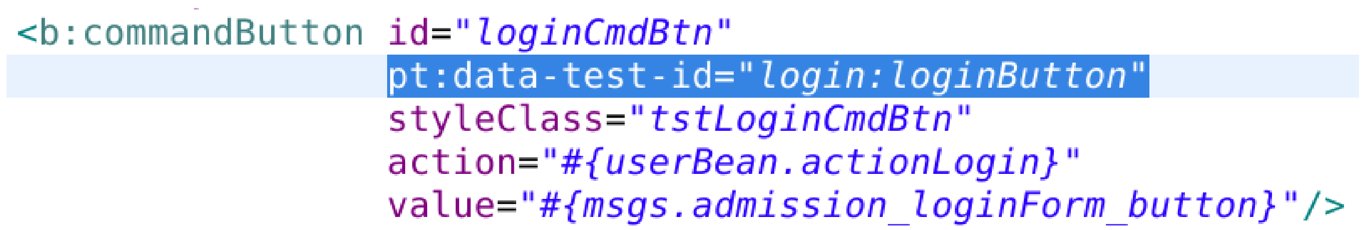
Element selectors

- by ID: driver.findElement(By.id("dtLj_id_s:j_id_11"));
- CSS selector: $("button[name=\"logInFormId:loginCmdBtn\"]").click();
- XPATH selector: driver.findElement(By.xpath("//span[contains(.,\'Benzene\')]"));
- by text: $(byText("Anmelden")).click();

…

Are your selectors stable?

- Define your own test-IDs!



```
<b:commandButton id="loginCmdBtn"
                 pt:data-test-id="login:loginButton"
                 styleClass="tstLoginCmdBtn"
                 action="#{userBean.actionLogin}"
                 value="#{msgs.admission_loginForm_button}"/>
```

- CSS selector: $("button[data-test-id=\"login:loginButton\"]").click();

Page objects

See

code/4_page_objects/

Leibniz-Institut für Pflanzenbiochemie

Dockerization with Selenoid
(= the geoid of the Moon; "σελήνη" = Moon)

https://aerokube.com/selenoid/latest/

See

code/5_selenoid/

Leibniz-Institut für Pflanzenbiochemie

Other browser test automation frameworks

- Playwright: JS, Python, Java, C#, (Ruby)
  also needs implicit waits for AJAX :(

See
code/6_playwright/

- Cypress: JS only
- Puppeteer: JS only

Leibniz-Institut für Pflanzenbiochemie