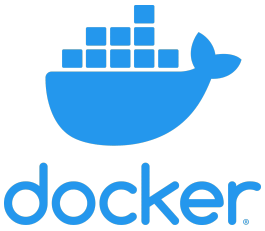
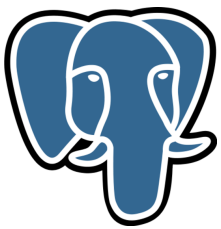
Was ist passiert

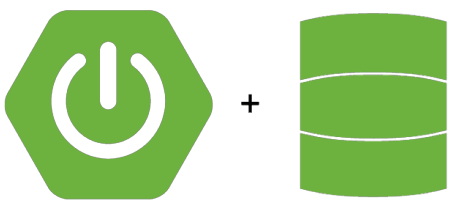
- Reguläre meetings
- Datenbankanbindung
- Benutzeranmeldung & Registrierung
- Frontend durch Thymeleaf ersetzt
- Testabdeckung verbessert
- Review Prozess durchgespielt

Datenbankanbindung



- docker & docker-compose
- PostgreSQL
- Liquibase
- JPA





Altes Frontend...

- LitComponents
- NPM
- webpack
- TypeScript
- Babel
- Kommunikation über Rest-API
- DevServer mit hot reloading
- mit gradle Frontend bauen und dann ins Backend kopieren

... mit Thymeleaf ersetzt

- server-side Java template engine
- sehr nah an HTML
- schnelle Entwicklung
- gute Integration in IntelliJ Spring Boot
- Nachteil: kein DevServer

```
<table>
  <thead>
    <tr>
      <th th:text="#{msgs.headers.name}">Name</th>
      <th th:text="#{msgs.headers.price}">Price</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="prod: ${allProducts}">
      <td th:text="${prod.name}">Oranges</td>
      <td th:text="${#numbers.formatDecimal(prod.price, 1, 2)}">
          0.99</td>
    </tr>
  </tbody>
</table>
```

Testabdeckung verbessert

- Unit und Integrationstests eingesetzt
- Benutzte Bibliotheken:
  - JUnit
  - Mockito
  - spring-boot-starter-test
  - spring-security-test
  - H2 in memory DB
- Testabdeckung auf 82% erhöhen

Beispiel für einen Unit Test mit Mockito

```
1  public class TaskServiceTest {
2
3      DBAbstraction dbAbstraction = Mockito.mock(DBAbstraction.class);
4      private final TaskService cut = new TaskService(dbAbstraction);
5
6      @Test
7      void getAllTasks_databaseReturnsData_arrayEquals() {
8          // Given
9          var tasks = new CodeTask[]{new CodeTask("taskName", "taskDesc", "psvm")};
0          Mockito.when(dbAbstraction.getAllTasks()).thenReturn(tasks);
1
2          // When
3          var result = cut.getAllTasks();
4
5          // Then
6          Assertions.assertArrayEquals(tasks, result);
7      }
8
9  }
```

Beispiel für einen Integrations Test

```
1  @SpringBootTest
2  @AutoConfigureMockMvc
3  class CodeTaskControllerTest {
4  // test was shorted for presentation
5
6      @Autowired
7      MockMvc mockMvc;
8
9      @MockBean
0      UserRepository repository;
1
2      @MockBean
3      TaskService taskService;
4
5      @Test
6      void listAllTasks_noLoginProvided_redirectedToLoginPage() throws Exception {
7          mockMvc.perform(get("/api/task/listAll"))
8                  .andExpect(status().is3xxRedirection())
9                  .andExpect(redirectedUrlPattern("**/login"));
0      }
1  }
```

Review Prozess durchgespielt

- Leon hat Testabdeckung verbessert
- Gemeinsames Review mit der Gruppe
- Findings wurden besprochen
- Pull request wurde gemerged

→ Integration in GitHub wurde erforscht

FIN