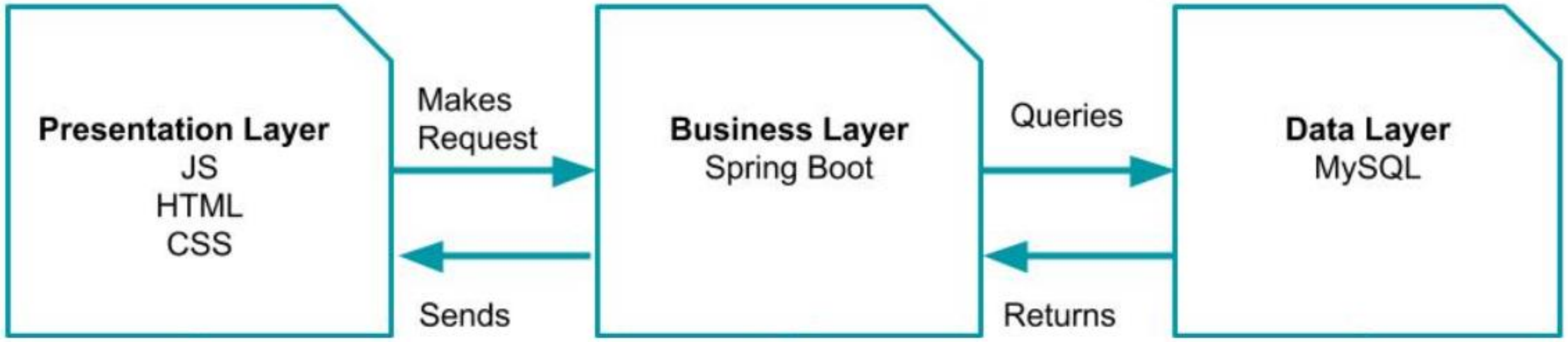
Practical Project-Hobby Web Application

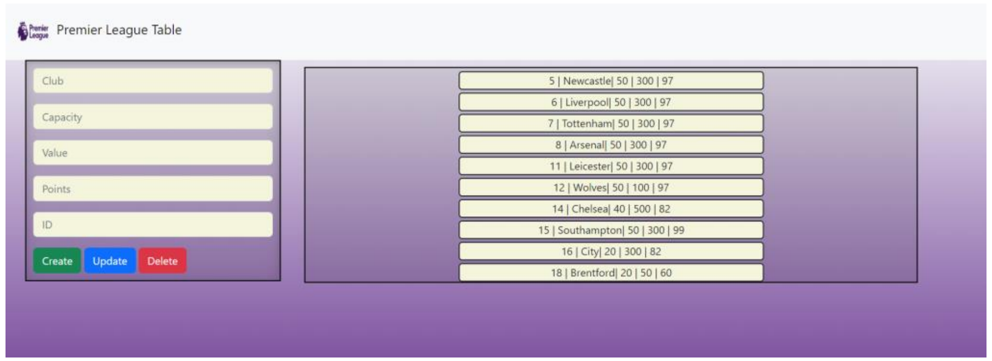
Jariful Hoque

Concept

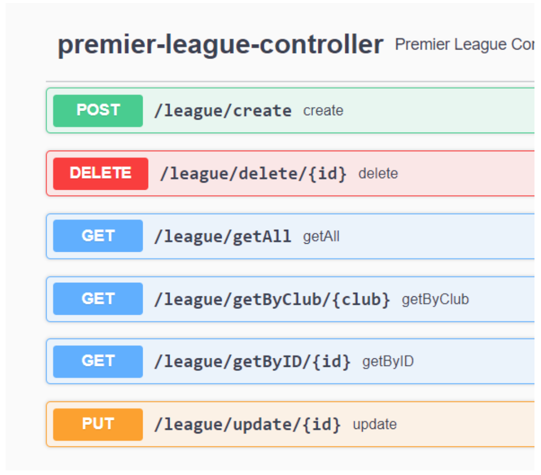
- OOP-based full-stack web application
- Database of Premier League football clubs
- CRUD functionality using API endpoints
- Front end with HTML, CSS, and JS
- Backend with Java, Spring, MySQL
- Reach 80% test coverage

Multi-Tier Architecture



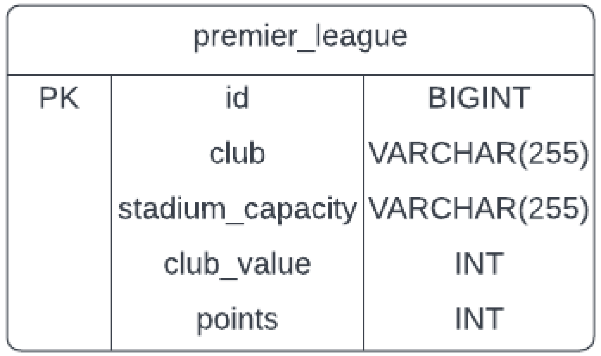


```
5 | Newcastle| 50 | 300 | 97
6 | Liverpool| 50 | 300 | 97
7 | Tottenham| 50 | 300 | 97
8 | Arsenal| 50 | 300 | 97
11 | Leicester| 50 | 300 | 97
12 | Wolves| 50 | 100 | 97
14 | Chelsea| 40 | 500 | 82
15 | Southampton| 50 | 300 | 99
16 | City| 20 | 300 | 82
18 | Brentford| 20 | 50 | 60
```



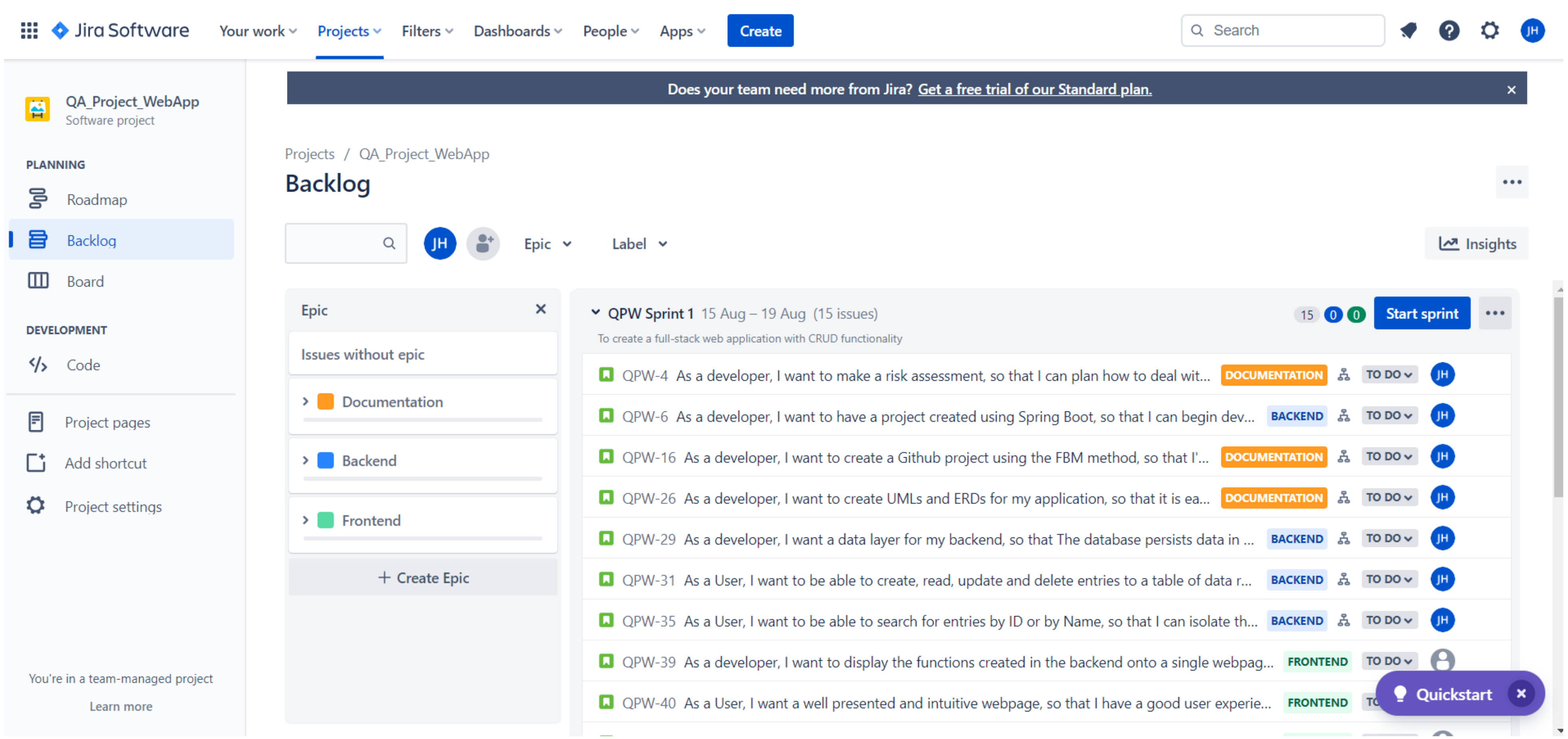
premier-league-controller Premier League Co

POST	`/league/create`	create
DELETE	`/league/delete/{id}`	delete
GET	`/league/getAll`	getAll
GET	`/league/getByClub/{club}`	getByClub
GET	`/league/getByID/{id}`	getByID
PUT	`/league/update/{id}`	update



premier_league		
PK	id	BIGINT
	club	VARCHAR(255)
	stadium_capacity	VARCHAR(255)
	club_value	INT
	points	INT

Agile



Jira Software · Your work · Projects · Filters · Dashboards · People · Apps · Create · Search

Does your team need more from Jira? Get a free trial of our Standard plan.

QA_Project_WebApp
Software project

PLANNING
- Roadmap
- Backlog
- Board

DEVELOPMENT
- Code
- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
Learn more

Projects / QA_Project_WebApp

Backlog

JH · Epic · Label · Insights

Epic
- Issues without epic
- Documentation
- Backend
- Frontend
- + Create Epic

QPW Sprint 1 15 Aug – 19 Aug (15 issues) 15 0 0 Start sprint

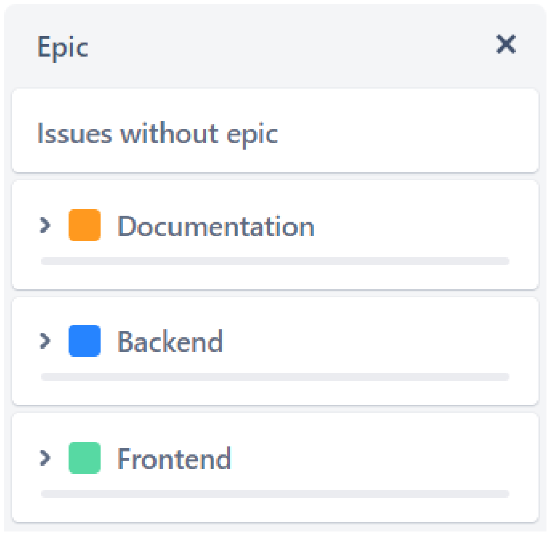
To create a full-stack web application with CRUD functionality

QPW-4	As a developer, I want to make a risk assessment, so that I can plan how to deal wit...	DOCUMENTATION	TO DO	JH
QPW-6	As a developer, I want to have a project created using Spring Boot, so that I can begin dev...	BACKEND	TO DO	JH
QPW-16	As a developer, I want to create a Github project using the FBM method, so that I'...	DOCUMENTATION	TO DO	JH
QPW-26	As a developer, I want to create UMLs and ERDs for my application, so that it is ea...	DOCUMENTATION	TO DO	JH
QPW-29	As a developer, I want a data layer for my backend, so that The database persists data in ...	BACKEND	TO DO	JH
QPW-31	As a User, I want to be able to create, read, update and delete entries to a table of data r...	BACKEND	TO DO	JH
QPW-35	As a User, I want to be able to search for entries by ID or by Name, so that I can isolate th...	BACKEND	TO DO	JH
QPW-39	As a developer, I want to display the functions created in the backend onto a single webpag...	FRONTEND	TO DO	
QPW-40	As a User, I want a well presented and intuitive webpage, so that I have a good user experie...	FRONTEND	TO	

Quickstart

Agile

Epics, User Stories & Tasks

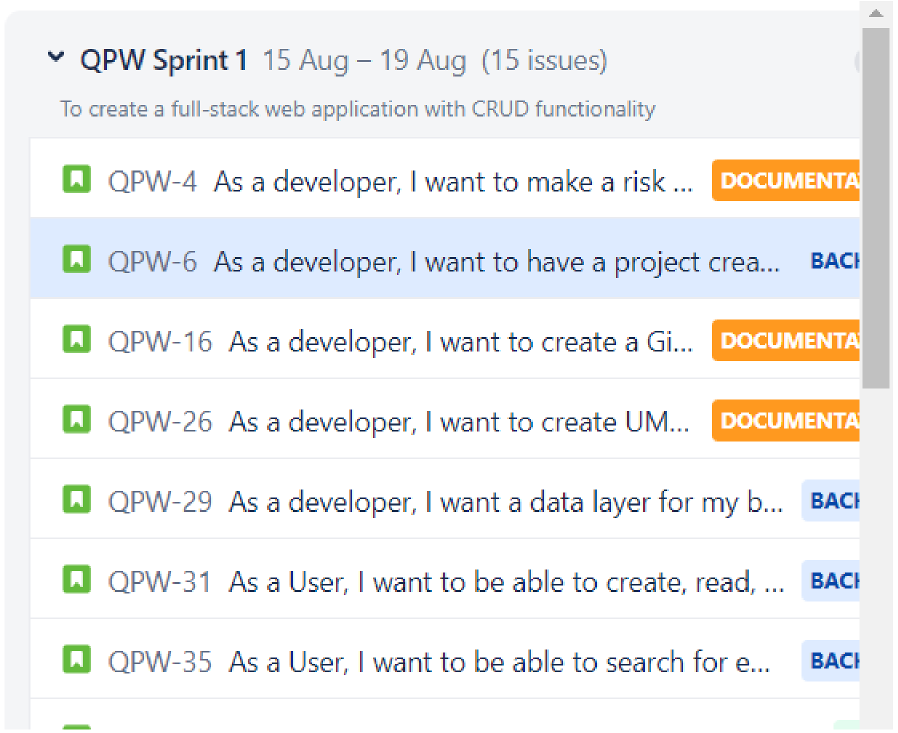

Epic

Issues without epic

🟧 Documentation

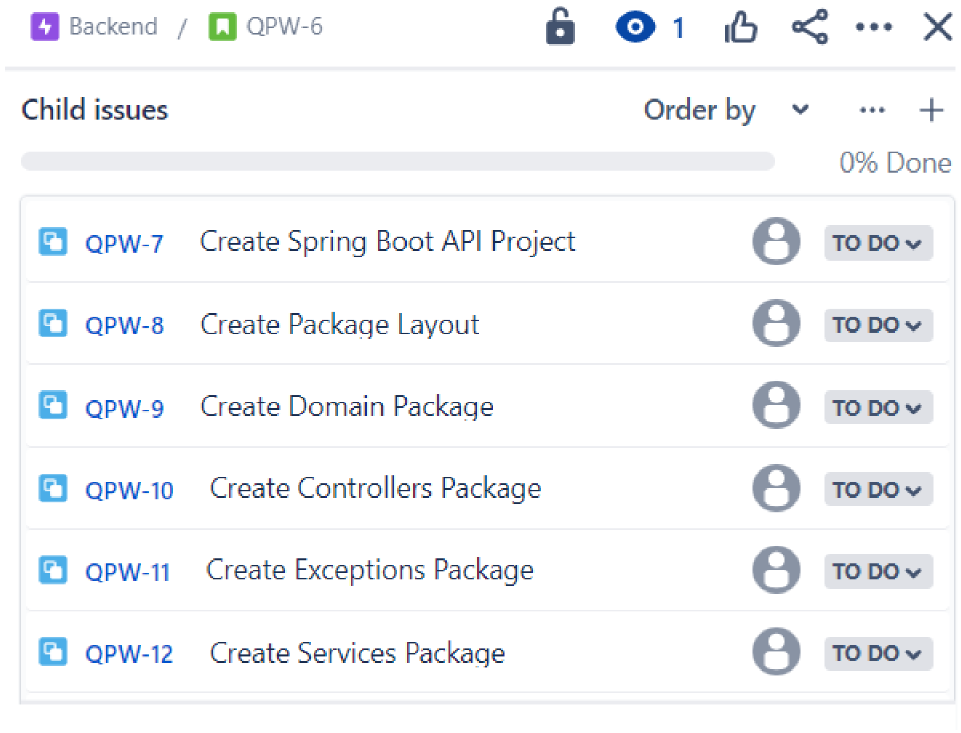
🟦 Backend

🟩 Frontend


QPW Sprint 1 15 Aug – 19 Aug (15 issues)

To create a full-stack web application with CRUD functionality

QPW-4	As a developer, I want to make a risk ...	DOCUMENTA
QPW-6	As a developer, I want to have a project crea...	BACK
QPW-16	As a developer, I want to create a Gi...	DOCUMENTA
QPW-26	As a developer, I want to create UM...	DOCUMENTA
QPW-29	As a developer, I want a data layer for my b...	BACK
QPW-31	As a User, I want to be able to create, read, ...	BACK
QPW-35	As a User, I want to be able to search for e...	BACK

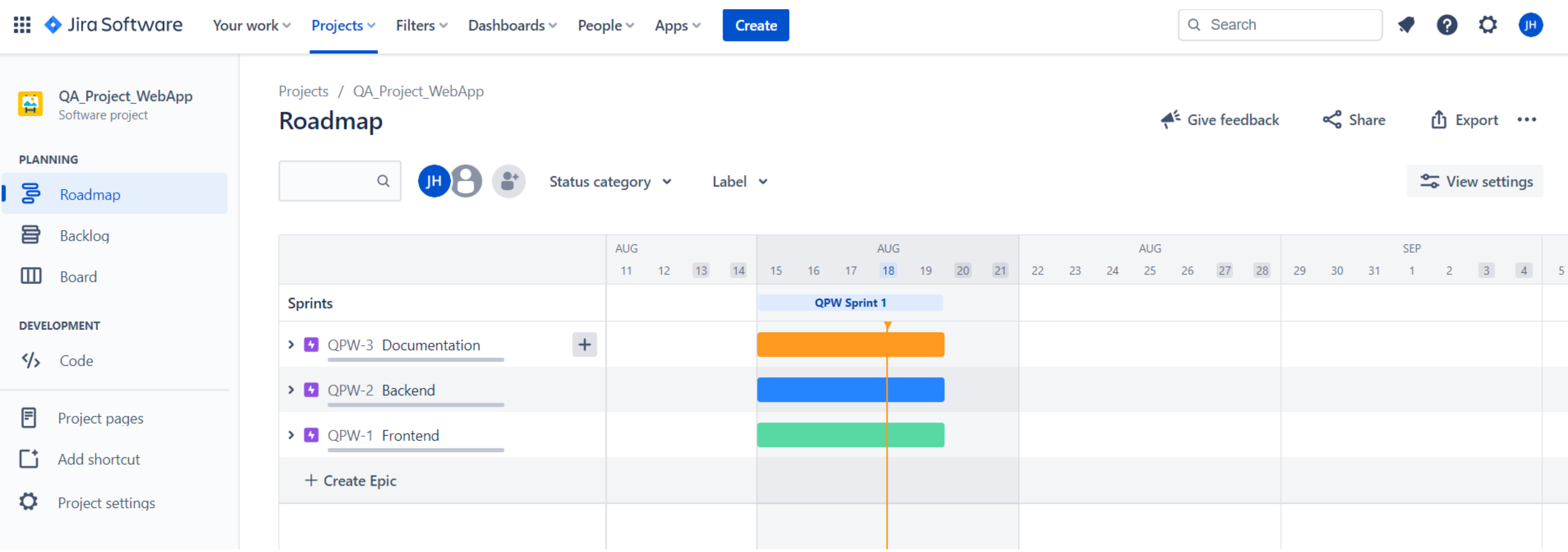

Backend / QPW-6

Child issues — Order by

0% Done

QPW-7	Create Spring Boot API Project	TO DO
QPW-8	Create Package Layout	TO DO
QPW-9	Create Domain Package	TO DO
QPW-10	Create Controllers Package	TO DO
QPW-11	Create Exceptions Package	TO DO
QPW-12	Create Services Package	TO DO

Agile



Projects / QA_Project_WebApp

Roadmap

Sprints: QPW Sprint 1

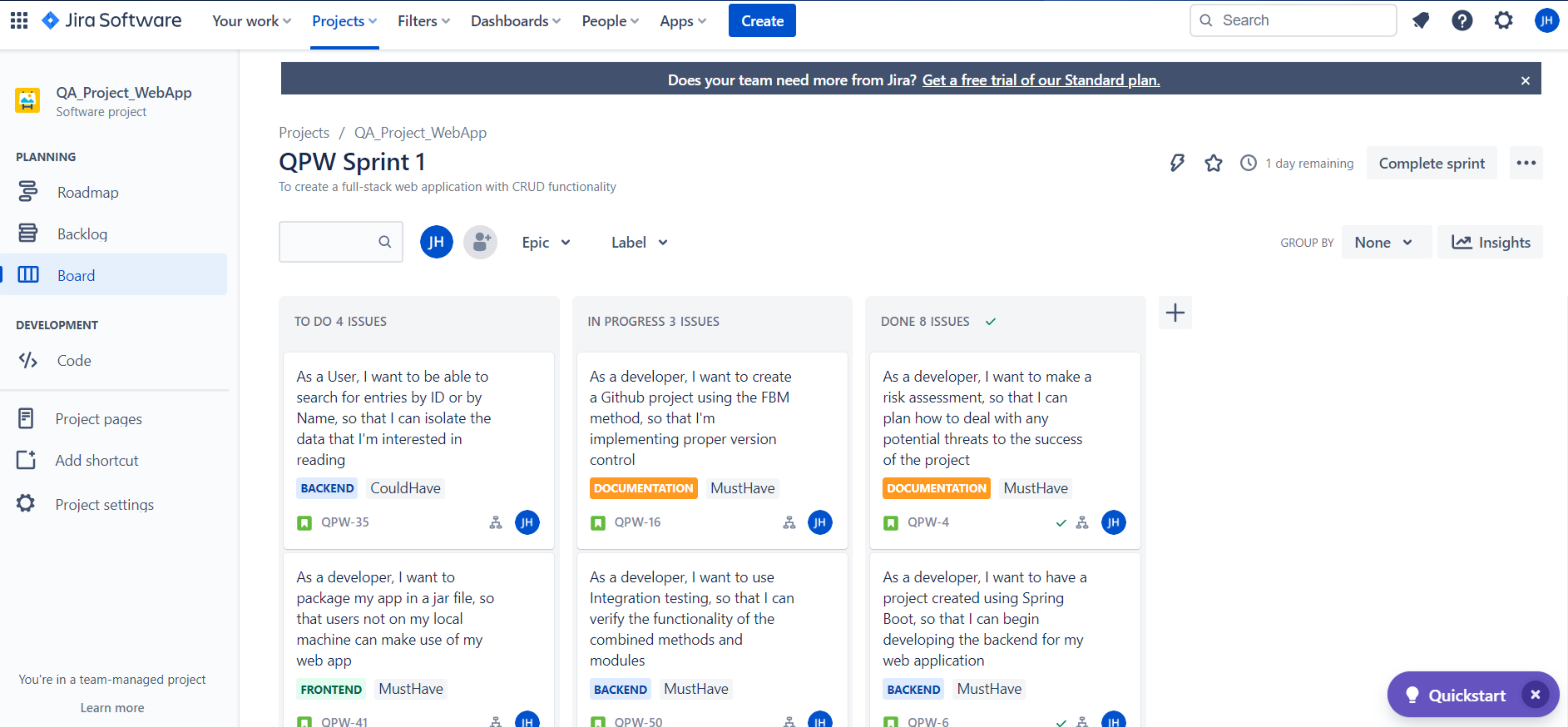
QPW-3 Documentation

QPW-2 Backend

QPW-1 Frontend

+ Create Epic

Agile



Does your team need more from Jira? Get a free trial of our Standard plan.

QA_Project_WebApp
Software project

PLANNING
Roadmap
Backlog
Board

DEVELOPMENT
Code

Project pages
Add shortcut
Project settings

You're in a team-managed project
Learn more

Projects / QA_Project_WebApp

QPW Sprint 1

To create a full-stack web application with CRUD functionality

1 day remaining · Complete sprint

Epic · Label · GROUP BY None · Insights

TO DO 4 ISSUES

- As a User, I want to be able to search for entries by ID or by Name, so that I can isolate the data that I'm interested in reading — BACKEND, CouldHave — QPW-35 — JH
- As a developer, I want to package my app in a jar file, so that users not on my local machine can make use of my web app — FRONTEND, MustHave — QPW-41 — JH

IN PROGRESS 3 ISSUES

- As a developer, I want to create a Github project using the FBM method, so that I'm implementing proper version control — DOCUMENTATION, MustHave — QPW-16 — JH
- As a developer, I want to use Integration testing, so that I can verify the functionality of the combined methods and modules — BACKEND, MustHave — QPW-50 — JH

DONE 8 ISSUES ✓

- As a developer, I want to make a risk assessment, so that I can plan how to deal with any potential threats to the success of the project — DOCUMENTATION, MustHave — QPW-4 ✓ — JH
- As a developer, I want to have a project created using Spring Boot, so that I can begin developing the backend for my web application — BACKEND, MustHave — QPW-6 ✓ — JH

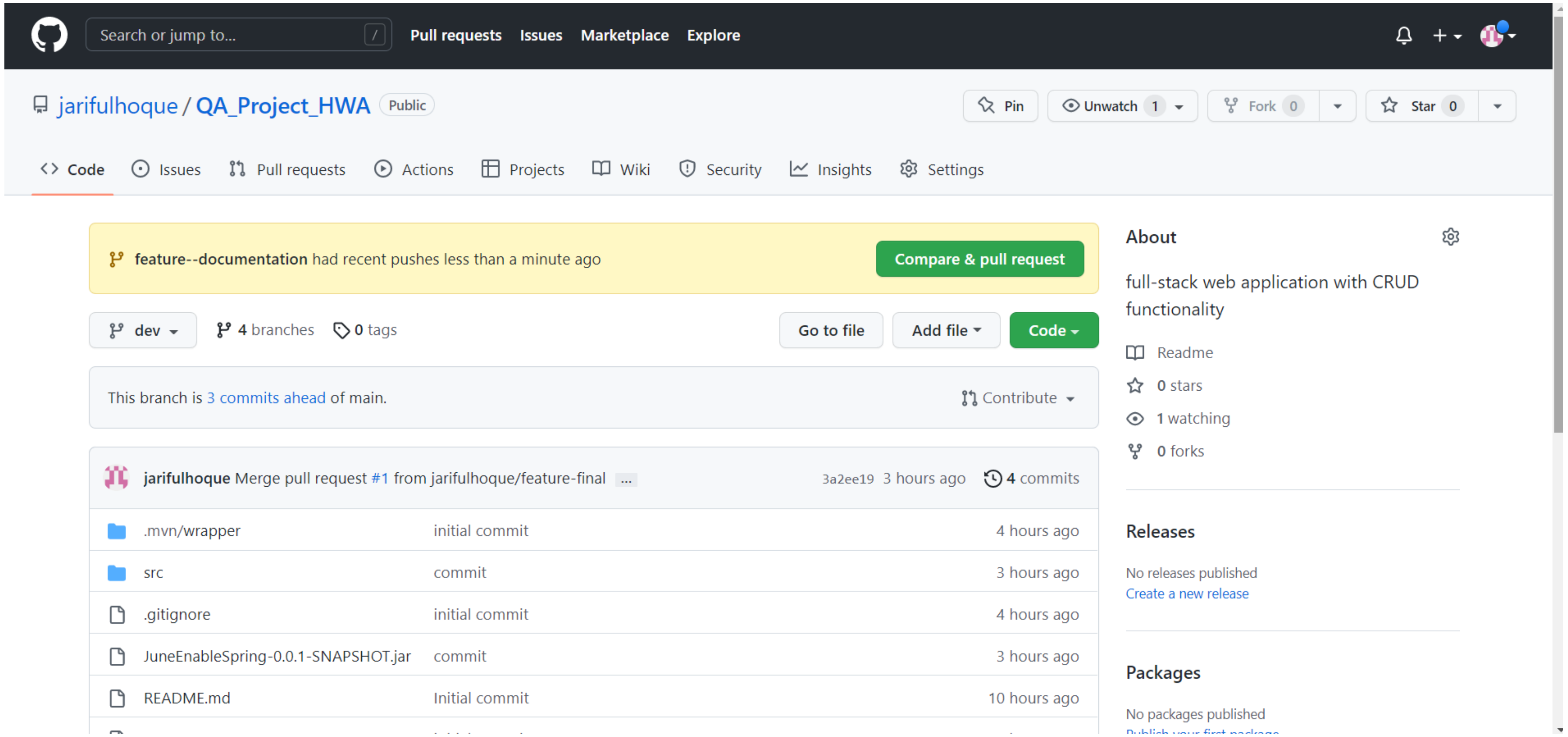
Quickstart

Consultant Journey

Technologies learned and applied for this project:

- Spring Boot API
- HTML, CSS, and JS
- MySQL
- Postman
- API endpoints- Post, Get, Put, Delete
- Swagger

Version Control



Search or jump to... | Pull requests | Issues | Marketplace | Explore

jarifulhoque / QA_Project_HWA Public

Pin | Unwatch 1 | Fork 0 | Star 0

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

feature--documentation had recent pushes less than a minute ago — Compare & pull request

dev | 4 branches | 0 tags | Go to file | Add file | Code

This branch is 3 commits ahead of main. Contribute

jarifulhoque Merge pull request #1 from jarifulhoque/feature-final ... 3a2ee19 3 hours ago 4 commits

.mvn/wrapper	initial commit	4 hours ago
src	commit	3 hours ago
.gitignore	initial commit	4 hours ago
JuneEnableSpring-0.0.1-SNAPSHOT.jar	commit	3 hours ago
README.md	Initial commit	10 hours ago

About

full-stack web application with CRUD functionality

Readme

0 stars

1 watching

0 forks

Releases

No releases published

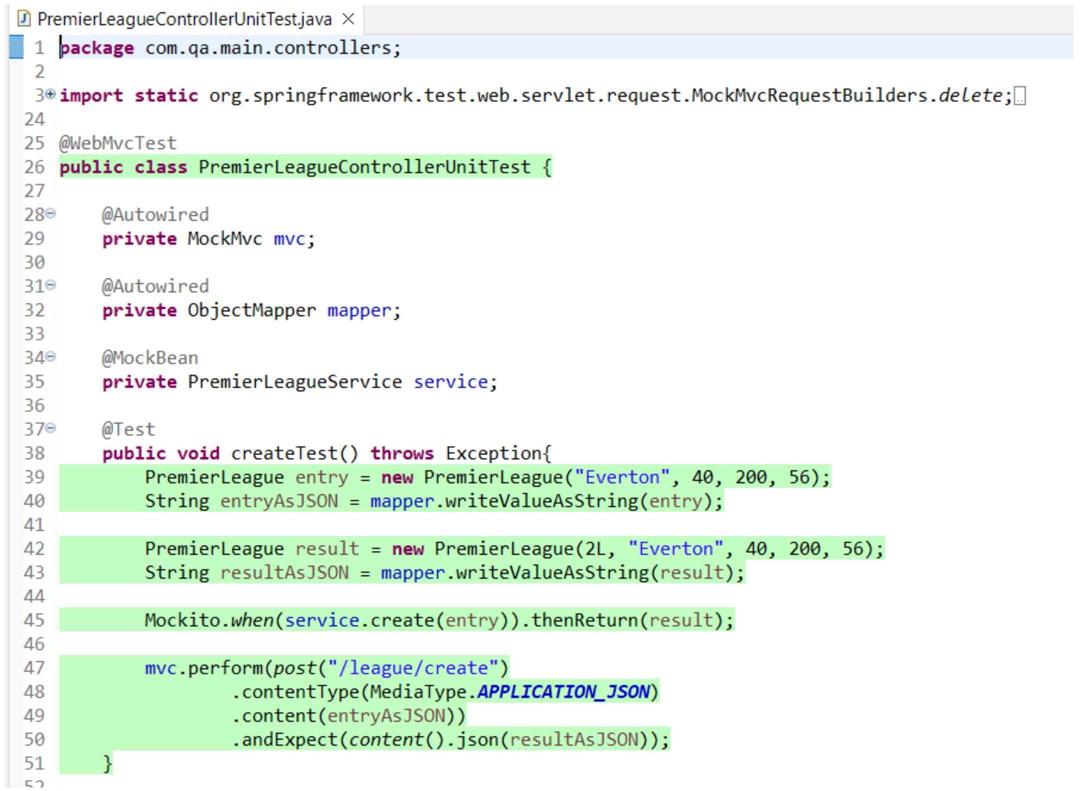
Create a new release

Packages

No packages published

Unit Tests

Unit testing is a testing approach that uses conditions to test smaller isolated portions of code that may be utilised logically. Unit testing was used in this project on the controllers and services classes, an example of which will be displayed here:

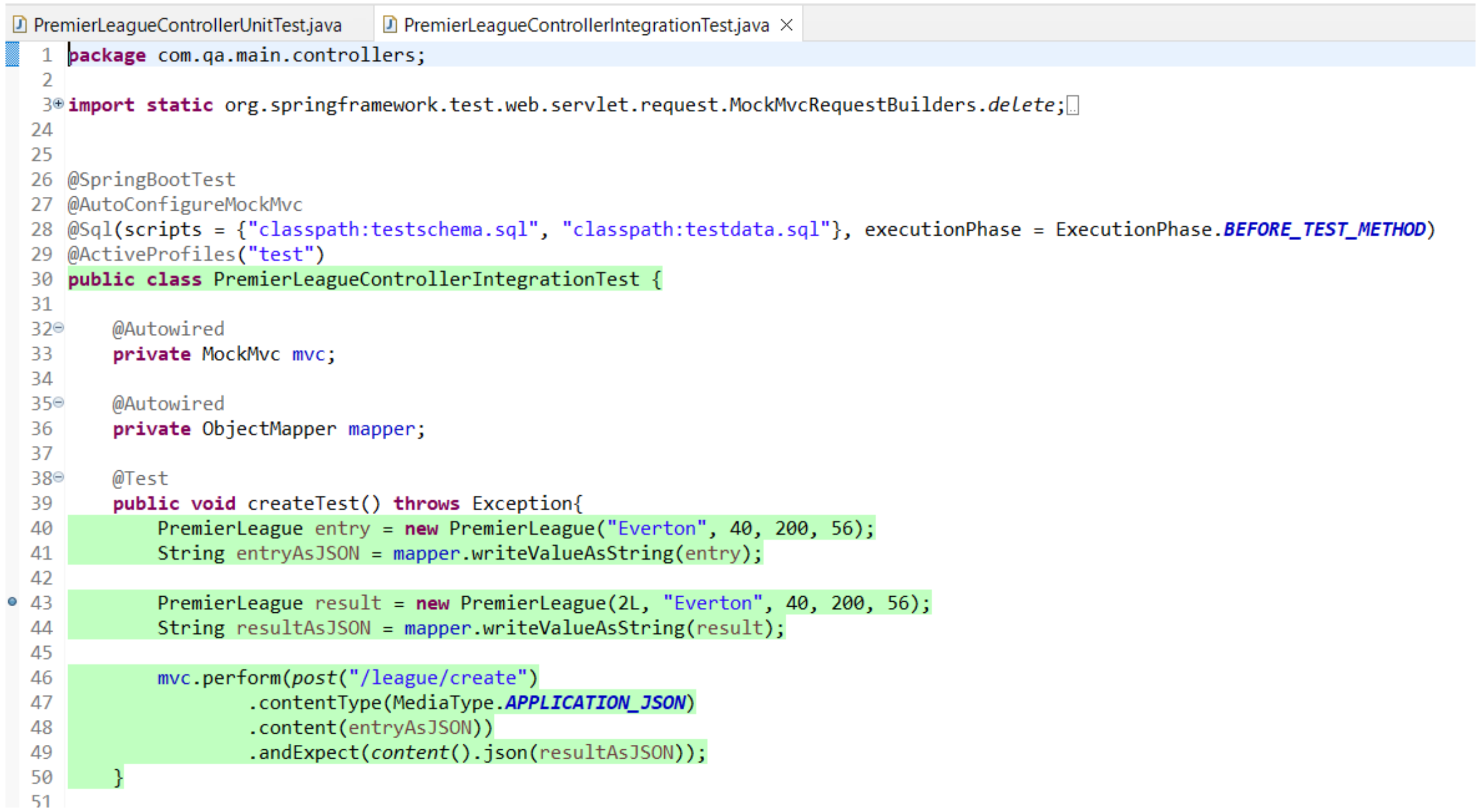


PremierLeagueControllerUnitTest.java

```java
package com.qa.main.controllers;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.delete;

@WebMvcTest
public class PremierLeagueControllerUnitTest {

    @Autowired
    private MockMvc mvc;

    @Autowired
    private ObjectMapper mapper;

    @MockBean
    private PremierLeagueService service;

    @Test
    public void createTest() throws Exception{
        PremierLeague entry = new PremierLeague("Everton", 40, 200, 56);
        String entryAsJSON = mapper.writeValueAsString(entry);

        PremierLeague result = new PremierLeague(2L, "Everton", 40, 200, 56);
        String resultAsJSON = mapper.writeValueAsString(result);

        Mockito.when(service.create(entry)).thenReturn(result);

        mvc.perform(post("/league/create")
                .contentType(MediaType.APPLICATION_JSON)
                .content(entryAsJSON))
                .andExpect(content().json(resultAsJSON));
    }
```

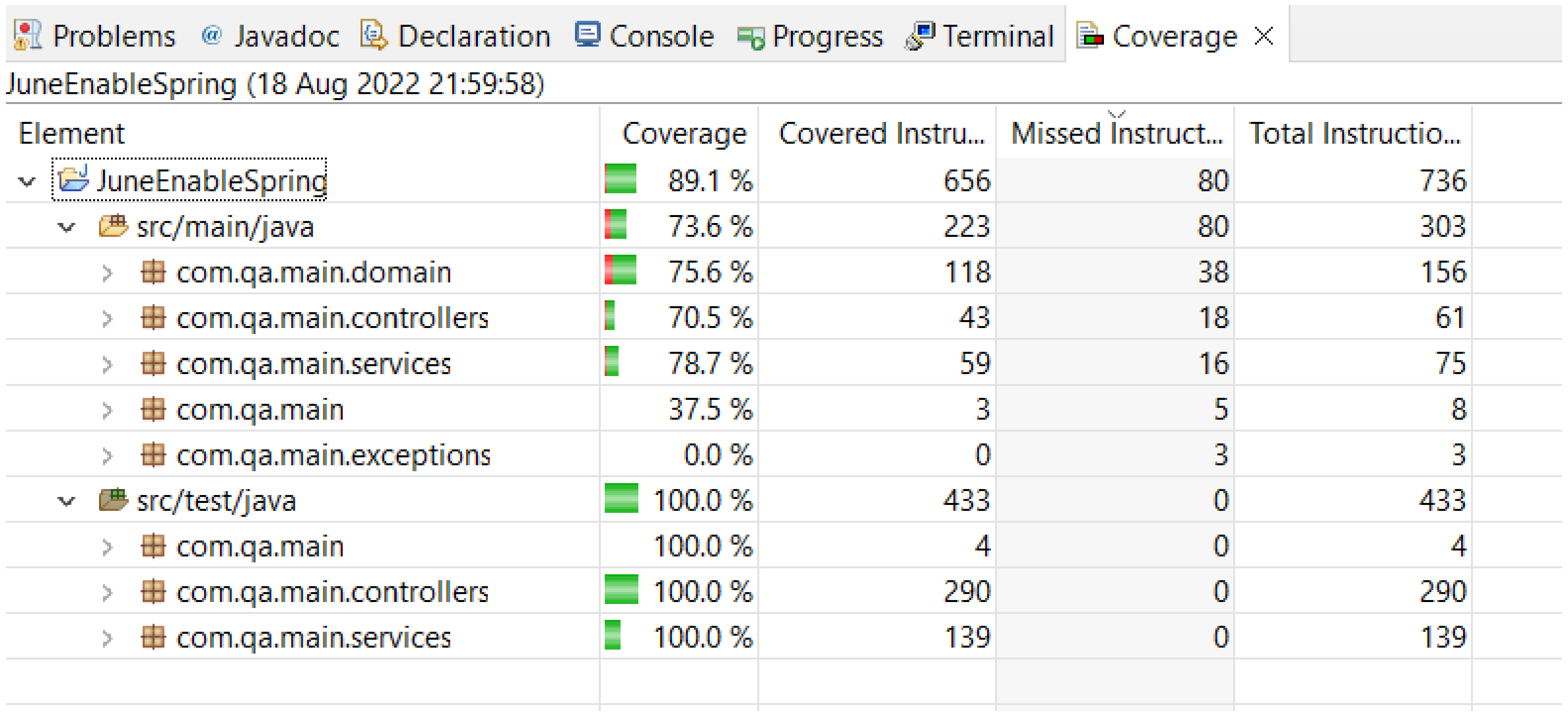
Integration Test

Integration testing is the process of evaluating various coupled components of an application to check if they logically operate together and achieve the desired result. Integration testing was used for the controller class as shown below:


PremierLeagueControllerUnitTest.java
PremierLeagueControllerIntegrationTest.java

```java
package com.qa.main.controllers;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.delete;


@SpringBootTest
@AutoConfigureMockMvc
@Sql(scripts = {"classpath:testschema.sql", "classpath:testdata.sql"}, executionPhase = ExecutionPhase.BEFORE_TEST_METHOD)
@ActiveProfiles("test")
public class PremierLeagueControllerIntegrationTest {

    @Autowired
    private MockMvc mvc;

    @Autowired
    private ObjectMapper mapper;

    @Test
    public void createTest() throws Exception{
        PremierLeague entry = new PremierLeague("Everton", 40, 200, 56);
        String entryAsJSON = mapper.writeValueAsString(entry);

        PremierLeague result = new PremierLeague(2L, "Everton", 40, 200, 56);
        String resultAsJSON = mapper.writeValueAsString(result);

        mvc.perform(post("/league/create")
                .contentType(MediaType.APPLICATION_JSON)
                .content(entryAsJSON))
                .andExpect(content().json(resultAsJSON));
    }

```

Coverage



JuneEnableSpring (18 Aug 2022 21:59:58)

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
JuneEnableSpring	89.1 %	656	80	736
src/main/java	73.6 %	223	80	303
com.qa.main.domain	75.6 %	118	38	156
com.qa.main.controllers	70.5 %	43	18	61
com.qa.main.services	78.7 %	59	16	75
com.qa.main	37.5 %	3	5	8
com.qa.main.exceptions	0.0 %	0	3	3
src/test/java	100.0 %	433	0	433
com.qa.main	100.0 %	4	0	4
com.qa.main.controllers	100.0 %	290	0	290
com.qa.main.services	100.0 %	139	0	139

Sprint Review

Further Improvements:

- Styling
- GetByID + Testing
- GetByClub + Testing
- In-line update and delete functions

Conclusion

- MVP Achieved
- Improvements could be made
- Cleaner Version Control
- More Comfortable with Jira
- Frontend Practice