

baeprojectone

- Clare Pritchard
- Chose to start with the very simplest minimum viable product
- A 'todo' list with
 - Name of thing todo
 - Description of the task
 - Priority of task (with the user completely free to define this)

baeprojectone

A ToDo List

- Persistence Layer mySQL server 8
- Backend Java, Maven, Springboot framework
- Frontend HTML/JavaScript/css (axios and bootstrap libraries)

Baeprojectone – planning – the approach

- Plan project, risk assessment, set up Jira(scrum) backlog and sprint
- Back end – java springboot application
- Test and production persistence layer (databases)
- Check it works using postman(because this is a very small app)
- Swagger can also be used(see screen shot on next slide)
- Front End – html page with javascript and css support
- Test it works from my new gui
- Write integration and unit tests

Bootstrap Buttons

Sign in to QA Community

Swagger UI


BAEP board - Agile board

cepGH1/BAEProjectOne at

H2 Console

New Tab

localhost:8080/swagger-ui/index.html#/

 **Swagger**
Supported by SMARTBEAR

/v3/api-docs

Explore

OpenAPI definition

v0OAS3

/v3/api-docs

Servers

http://localhost:8080 - Generated server url

thing-controller

PUT/Thing/update/{id}

POST/Thing/create

GET/Thing/hello

GET/Thing/getById/{id}

GET/Thing/getAll

DELETE/Thing/delete/{id}

Baeprojectone – planning – backlog and sprint

The screenshot displays the Jira Software interface for a project named 'BAEProjectOne'. The interface is divided into several sections:

- Left Sidebar:** Contains navigation links for 'PLANNING' (Roadmap, Backlog, Board) and 'DEVELOPMENT' (Code, Project pages, Add shortcut, Project settings).
- Top Bar:** Shows the project name 'BAEProjectOne' and a 'Set project and issue permissions' notification.
- Main Content Area:**
 - Backlog View:** Displays a list of issues under the 'BAEP Sprint 1' epic. The issues are categorized by type (e.g., Documentation, Frontend, Backend, Testing) and their progress is shown as a bar chart.
 - Sprint View:** Shows a detailed view of the 'BAEP Sprint 1' (25 Apr – 29 Apr) with a list of issues and their progress. The issues are categorized by type and their progress is shown as a bar chart.
- Right Panel:** Displays the 'Details' for the selected issue 'BAEP-16'. It includes the assignee 'clare pritchard', priority 'Medium', labels 'Should', sprint 'BAEP Sprint 1', and story point estimate '0.5'.

Baeprojectone – planning – backlog and sprint

- Create logical epics of related user stories
- Try to order the user stories in the same order that they will be tackled
- Split user stories into tasks (link tasks to user stories)
- Give issues descriptions, priority and storypoint estimations
- Load all issues into a time boxed sprint
- Start sprint

This should act as a todo list. Should ensure that no steps are missing and help to get to MVP.

Baeprojectone – what I learned

- The best thing I learned : axios to create and receive http requests carrying JSON payloads.
- Also learned lots about javascript and html dom manipulation
- Revised and reinforced what I knew about SpringBoot, maven
- Made me realise how little I knew about html/javascript
- Introduced me to bootstrap(I want to get to know this much better)

Baeprojectone – Version Control

- Git and GitHub
- Using main, dev and feature branches

The image shows a GitHub repository page for 'cepGH1 / BAEProjectOne' and a terminal window demonstrating Git commands.

GitHub Repository View:

- Repository: cepGH1 / BAEProjectOne (Public)
- Navigation: <> Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings
- Branches: 8 branches, 0 tags
- Current branch: dev (selected)
- Branch list (from top to bottom):
 - main (default)
 - dev (checked)
 - feature-backend-domain
 - feature-controller
 - feature-frontend
 - feature-risk-1
 - feature-service
 - feature-testing

Terminal Window (MINGW64):

```
$ git remote -v
origin https://github.com/cepGH1/BAEProjectOne.git (fetch)
origin https://github.com/cepGH1/BAEProjectOne.git (push)

clare@NEW-JR88F0T1DEJ MINGW64 ~/eclipse-workspace/baeprojectone (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean

clare@NEW-JR88F0T1DEJ MINGW64 ~/eclipse-workspace/baeprojectone (main)
$ git checkout -b dev
Switched to a new branch 'dev'

clare@NEW-JR88F0T1DEJ MINGW64 ~/eclipse-workspace/baeprojectone (dev)
$ git branch
* dev
  main

clare@NEW-JR88F0T1DEJ MINGW64 ~/eclipse-workspace/baeprojectone (dev)
$ git push -u origin dev
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:   https://github.com/cepGH1/BAEProjectOne/pull/new/dev
remote:
To https://github.com/cepGH1/BAEProjectOne.git
 * [new branch]      dev -> dev
Branch 'dev' set up to track remote branch 'dev' from 'origin'.

clare@NEW-JR88F0T1DEJ MINGW64 ~/eclipse-workspace/baeprojectone (dev)
$
```


Baeprojectone – Testing

- Integration testing using mock mvc
- Unit testing using Mockito

The screenshot displays the Eclipse IDE interface for a project named 'Baeprojectone'. The top toolbar includes standard IDE functions like File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the toolbar, the Package Explorer on the left shows the project structure, including 'src/main/java' and 'src/test/java'. The central editor window displays the code for 'ThingServiceUnitTest.java', which includes Mockito annotations and assertions for testing the 'ThingService'. The bottom console shows the test results, indicating that all tests passed successfully. The bottom status bar shows the current file is 'Writable' and the cursor is at '31 : 41 : 728'.

Finished after 20.164 seconds

Runs: 21/21 Errors: 0 Failures: 0

Test Results:

- ThingTest [Runner: JUnit 5] (0.566 s)
- ThingControllerIntegrationTest [Runner: JUnit 5] (1.178 s)
- ThingControllerUnitTest [Runner: JUnit 5] (0.171 s)
- BaeprojectoneApplicationTests [Runner: JUnit 5] (0.008 s)
- ThingServiceUnitTest [Runner: JUnit 5] (0.095 s)

Code Snippet (ThingServiceUnitTest.java):

```
43 Mockito.when(this.repo.findAll()).thenReturn(output);
44
45 assertEquals(output, this.service.getAll());
46
47 Mockito.verify(this.repo, Mockito.times(1)).findAll();
48 }
49
50 @Test
51 public void getByIdTest() {
52     Optional<Thing> OptionalOutput = Optional.of(new Thing(1L, "kid", "green army", "low"));
53     Thing output = new Thing(1L, "kid", "green army", "low");
54
55     Mockito.when(this.repo.findById(1L)).thenReturn(OptionalOutput);
56
57     assertEquals(output, this.service.getById(1L));
58
59     Mockito.verify(this.repo, Mockito.times(1)).findById(1L);
60 }
61
62 @Test
63 public void createTest() {
64     Thing input = new Thing("kid", "green army", "low");
65     Thing output = new Thing(1L, "kid", "green army", "low");
66
67     Mockito.when(this.repo.saveAndFlush(input)).thenReturn(output);
68
69     assertEquals(output, this.service.create(input));
70 }
```

Console Output:

java (28 Apr 2022 15:41:25)

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
baeprojectone	91.5 %	872	81	953
src/main/java	74.1 %	232	81	313
com.qa.main.domain	63.5 %	99	57	156
com.qa.main.service	80.9 %	72	17	89
com.qa.main	37.5 %	3	5	8
com.qa.main.controller	96.7 %	58	2	60
src/test/java	100.0 %	640	0	640

Writable Smart Insert 31 : 41 : 728

Demo

Baeprjectone – Sprint Review

- Completed all of my (not very ambitious) sprint issues
- Would like better test coverage

Baeprojectone – Sprint retrospective

- The Front End could be improved a lot
 - Adding interactivity to my overall list
- Version 2 of this project would include a deadline date in the entity and javascript to show how close to deadline the item is. (amongst other features)

Baeprojectone – Conclusion

- I could have been more ambitious with my entity, used more columns in my table and demonstrated that I do know that other data-types can be held in a MySQL schema
- This is very good practice for using springboot
- I particularly need to practise with bootstrap and javascript Dom methods.