



Sprint 2 Planning

Estimated time needed: 30 minutes

Welcome to the hands-on **Sprint 2 Planning** lab. With your first sprint complete, it is now time to plan for Sprint 2. In this lab, you will add automation for continuous integration and add security features to your microservice.

Objectives

In this lab, you will:

- Create stories for Sprint 2
- Add the stories to the kanban board
- Add labels and estimates to make them sprint ready
- Assemble a Sprint backlog

Note on Screenshots

Throughout this lab, you will be prompted to take screenshots and save them on your device. You will need these screenshots to answer graded quiz questions or upload as your submission for peer review at the end of this course. Your screenshot must have either the .jpg or .png extension.

To take screenshots, you can use various free screen-capture tools or your operating system's shortcut keys. For example:

- **Mac:** You can use **Shift + Command + 3** (⌘ + ⌘ + 3) on your keyboard to capture your entire screen, or **Shift + Command + 4** (⌘ + ⌘ + 4) to capture a window or area. They will be saved as a file on your Desktop.
- **Windows:** You can capture your active window by pressing **Alt + Print Screen** on your keyboard. This command copies an image of your active window to the clipboard. Next, open an image editor, paste the image from your clipboard to the image editor, and save the image.

New Requirements

Management has been looking for ways to increase developer productivity and has noticed that developers spend a lot of time checking that all the tests pass before approving each pull request. Management has decided it is time to automate this task by implementing continuous integration (CI) using GitHub Actions.

There have also been many stories in the news about security breaches and exploits, and management is concerned that about the security of your microservice. In an effort to be proactive, they have decided that you need to add defensive security measures to your microservice in the form of security headers and cross-origin resource sharing (CORS) policies.

To satisfy these requirements, you need to add two new stories to your **Sprint Backlog**.

- Need the ability to automate continuous integration checks
- Need to add security headers and CORS policies

Then you will make a sprint plan from them.

Story 1: Need the Ability to Automate Continuous Integration Checks

See the first story below. You can just cut and paste it to get started.

1. Create the following story about CI automation in the **Product Backlog** column of your Zenhub kanban board:

Note: you can cut-n-paste the markdown text below:

1. 1

2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20

```
1.
2.   Title: Need the ability to automate continuous integration checks
3.
4.   **As a** Developer
5.   **I need** automation to build and test every pull request
6.   **So that** I do not have to rely on manual testing of each request, which is time-consuming
7.
8.   #### Assumptions
9.   * GitHub Actions will be used for the automation workflow
10.  * The workflow must include code linting and testing
11.  * The Docker image should be postgres:alpine for the database
12.  * A GitHub Actions badge should be added to the README.md to reflect the build status
13.
14.  #### Acceptance Criteria
15.  ```gherkin
16.  Given code is ready to be merged
17.  When a pull request is created
18.  Then GitHub Actions should run linting and unit tests
19.  And the badge should show that the build is passing
20.  ```
```

Copied!

1. Since this story adds no visible stakeholder value and is primarily to make development more efficient, assign this story a technical debt label.
2. Give the story an estimate. Remember the values are Small = 3, Medium = 5, Large = 8, and Extra Large = 13.
3. Next, assign the story to Sprint 2.
4. Finally, move the story to the top of the Sprint Backlog.

Story 2: Need to Add Security Headers and CORS Policies

1. Create the following story about security enhancements in the Product Backlog column of your Zenhub kanban board:

Note: you can cut-n-paste the markdown text below:

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15

16. 16

```
1.      Title: Need to add security headers and CORS policies
2.
3.      **As a** service provider
4.      **I need** my service to use security headers and CORS policies
5.      **So that** my web site is not vulnerable to CORS attacks
6.
7.      #### Assumptions
8.      * Flask-Talisman will be used for security headers
9.      * Flask-Cors will be used to establish cross-origin resource sharing (CORS) policies
10.
11.     #### Acceptance Criteria
12.     ``gherkin
13.     Given the site is secured
14.     When a REST API request is made
15.     Then secure headers and a CORS policy should be returned
16.     ``
```

Copied!

- 1. This story will make the microservice more secure, and better security adds stakeholder value. Assign the appropriate label to this story.
- 2. Give the story an estimate. Remember the values are Small = 3, Medium = 5, Large = 8, and Extra Large = 13.
- 3. Next, assign the story to Sprint 2.
- 4. Finally, move the the story to the Sprint Backlog, ranking it second.

You are now ready to start the sprint.

Evidence

- 1. Take a screenshot of your kanban board as sprint2-plan.jpg (or sprint2-plan.png).

Conclusion

Congratulations! You have created your second sprint plan. You will use this in the next two labs to implement those two stories.

Next Steps

In the next lesson, you will begin to implement the stories in Sprint 2.

Author

[John J. Rofrano](#)

Other Contributor(s)

Change Log

Date	Version	Changed by	Change Description
2022-10-13	0.1	John Rofrano	Initial version created
2022-10-14	0.2	Amy Norton	ID review
2022-10-14	0.3	John Rofrano	Updated screenshot image names
2022-10-18	0.4	Beth Larsen	QA pass
2022-10-20	0.5	Beth Larsen	Updated assumptions wording per input from John
2022-10-28	0.6	John Rofrano	Updated markdown in story text
2023-03-16	0.7	Lavanya Rajalingam	Updated SN Logo