COEN 6761

Software Validation and Testing

Winter 2026

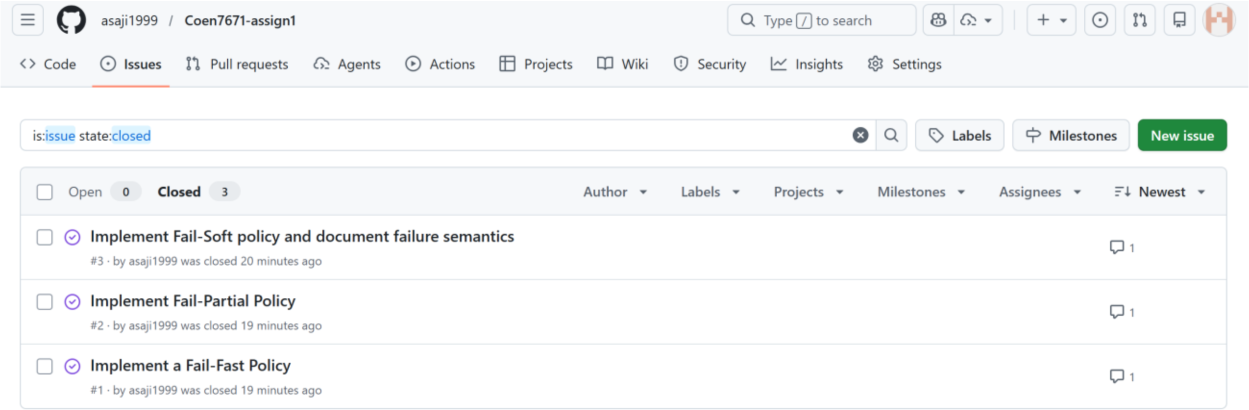
Submitting to:

Prof. Yan Liu

Submitted by:

Azan Ali Shah Syed (40241942)

## Github Issues:



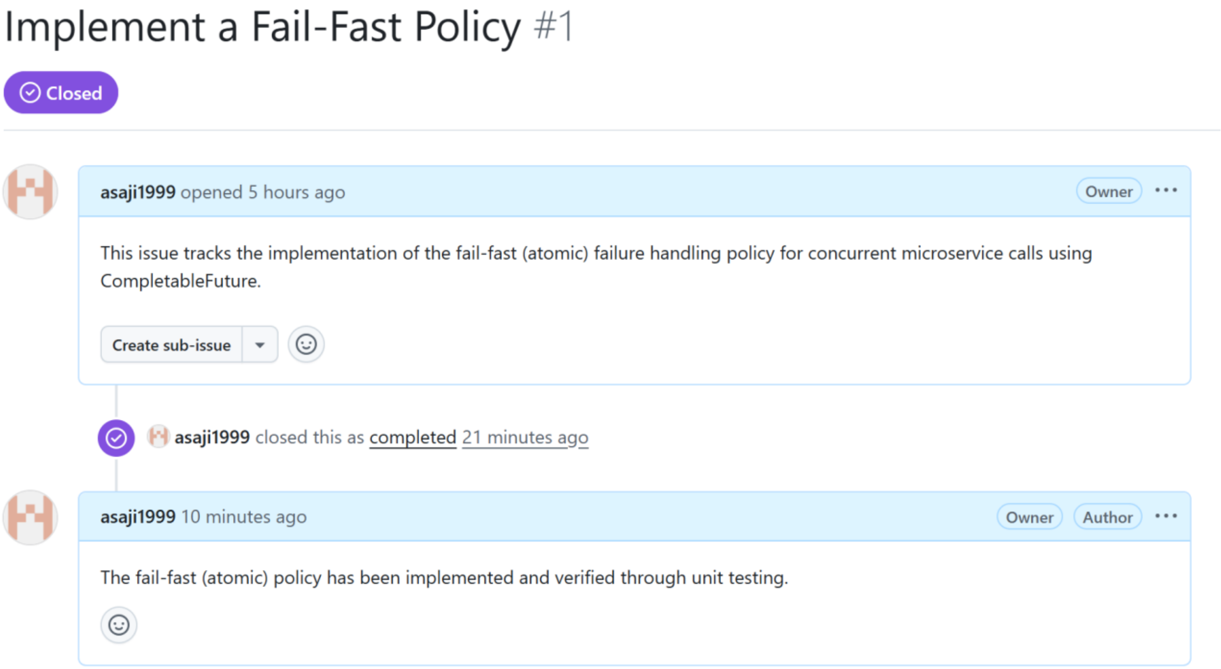
is:issue state:closed

Open 0 Closed 3

- Implement Fail-Soft policy and document failure semantics — #3 · by asaji1999 was closed 20 minutes ago — 1
- Implement Fail-Partial Policy — #2 · by asaji1999 was closed 19 minutes ago — 1
- Implement a Fail-Fast Policy — #1 · by asaji1999 was closed 19 minutes ago — 1

*Figure 1: GitHub issues created to track implementation of failure policies and documentation.*

## Issue 1:



Implement a Fail-Fast Policy #1

asaji1999 opened 5 hours ago

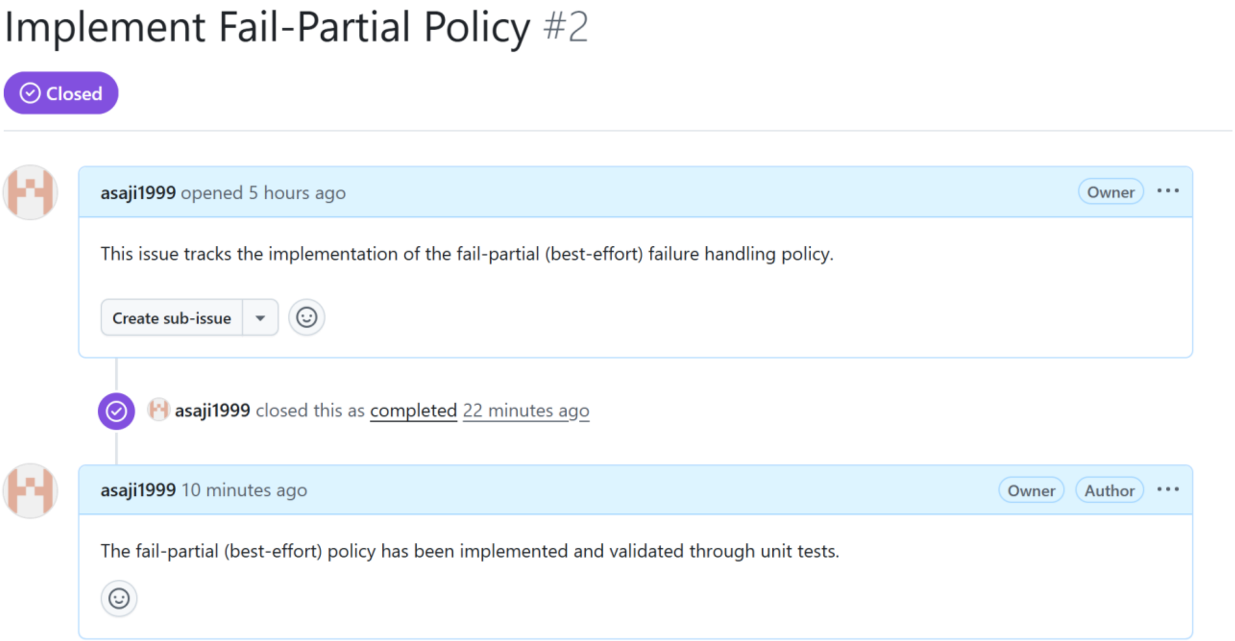
This issue tracks the implementation of the fail-fast (atomic) failure handling policy for concurrent microservice calls using CompletableFuture.

asaji1999 closed this as completed 21 minutes ago

asaji1999 10 minutes ago

The fail-fast (atomic) policy has been implemented and verified through unit testing.

**Issue 2:**



Implement Fail-Partial Policy #2

asaji1999 opened 5 hours ago

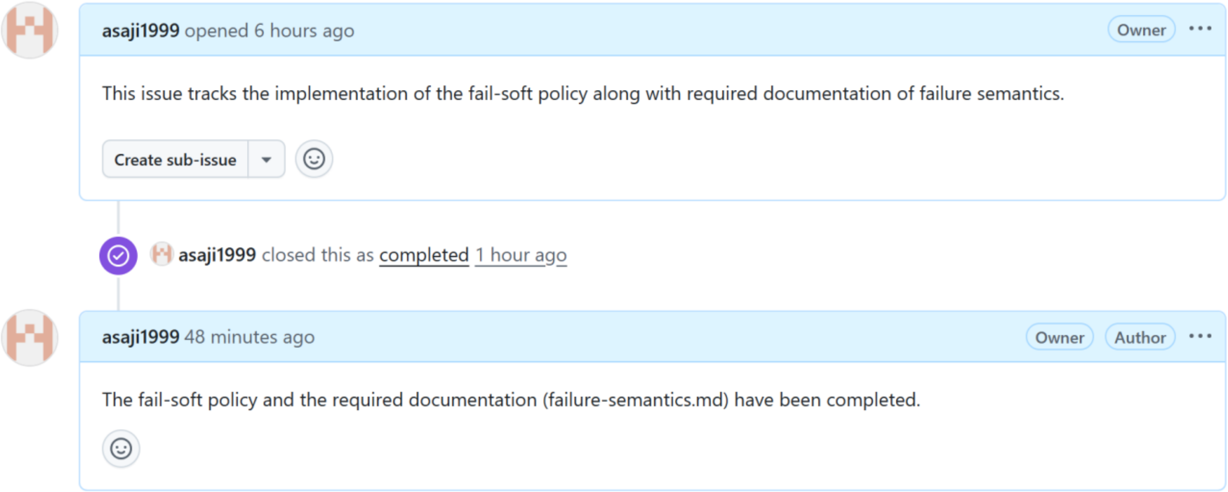
This issue tracks the implementation of the fail-partial (best-effort) failure handling policy.

asaji1999 closed this as completed 22 minutes ago

asaji1999 10 minutes ago

The fail-partial (best-effort) policy has been implemented and validated through unit tests.

**Issue 3:**



asaji1999 opened 6 hours ago

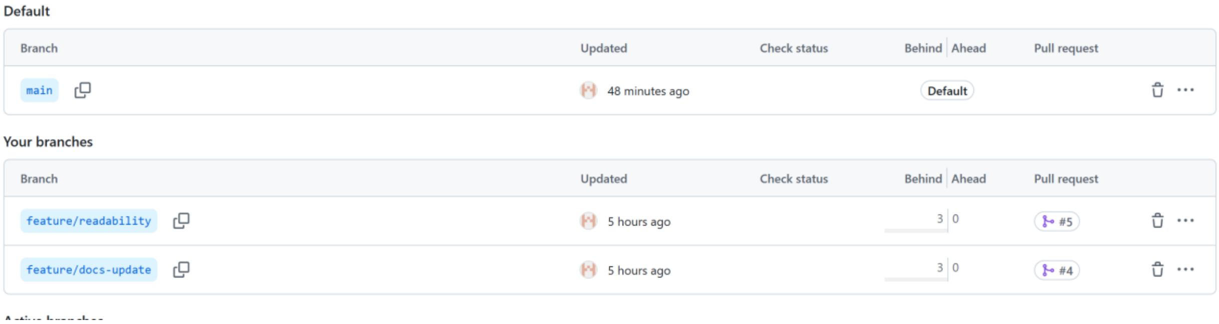
This issue tracks the implementation of the fail-soft policy along with required documentation of failure semantics.

asaji1999 closed this as completed 1 hour ago

asaji1999 48 minutes ago

The fail-soft policy and the required documentation (failure-semantics.md) have been completed.

## Feature Branches:



Default

| Branch | Updated | Check status | Behind | Ahead | Pull request |
|---|---|---|---|---|---|
| main | 48 minutes ago | | Default | | |

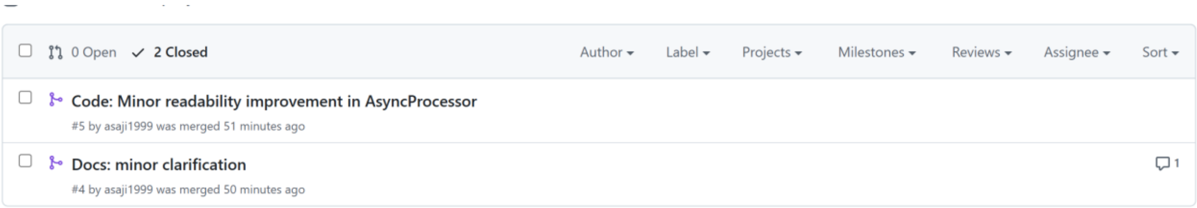
Your branches

| Branch | Updated | Check status | Behind | Ahead | Pull request |
|---|---|---|---|---|---|
| feature/readability | 5 hours ago | | 3 | 0 | #5 |
| feature/docs-update | 5 hours ago | | 3 | 0 | #4 |

Active branches

*Main branch and two Feature Branches.*

## Pull Requests:



□ Code: Minor readability improvement in AsyncProcessor

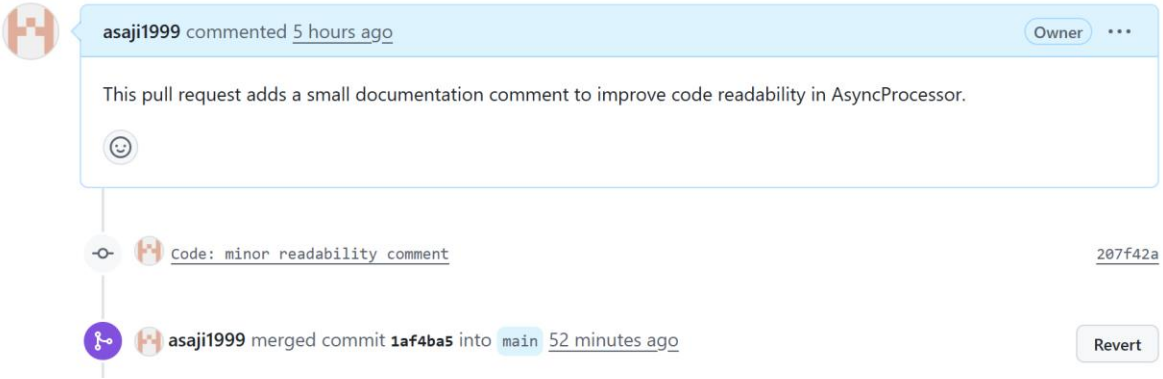
#5 by asaji1999 was merged 51 minutes ago

□ Docs: minor clarification 1

#4 by asaji1999 was merged 50 minutes ago

*Pull requests used to merge feature branches into main.*

## PR # 1:



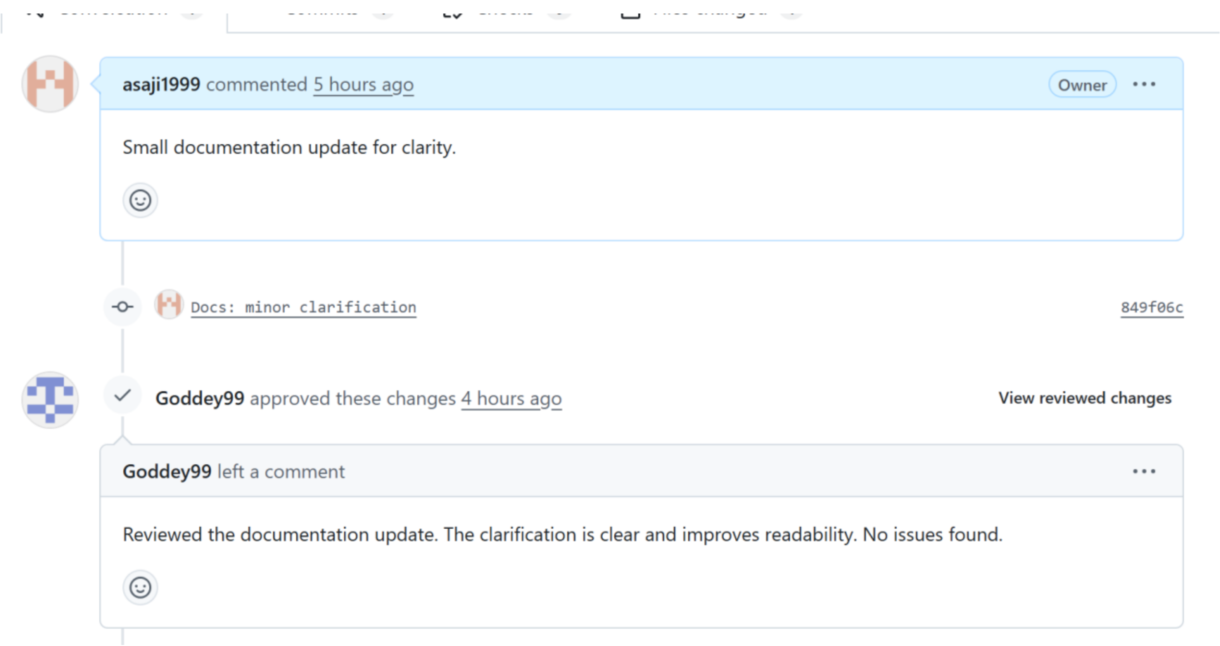
asaji1999 commented 5 hours ago

This pull request adds a small documentation comment to improve code readability in AsyncProcessor.

Code: minor readability comment 207f42a

asaji1999 merged commit 1af4ba5 into main 52 minutes ago

**PR# 2:**



asaji1999 commented 5 hours ago

Small documentation update for clarity.

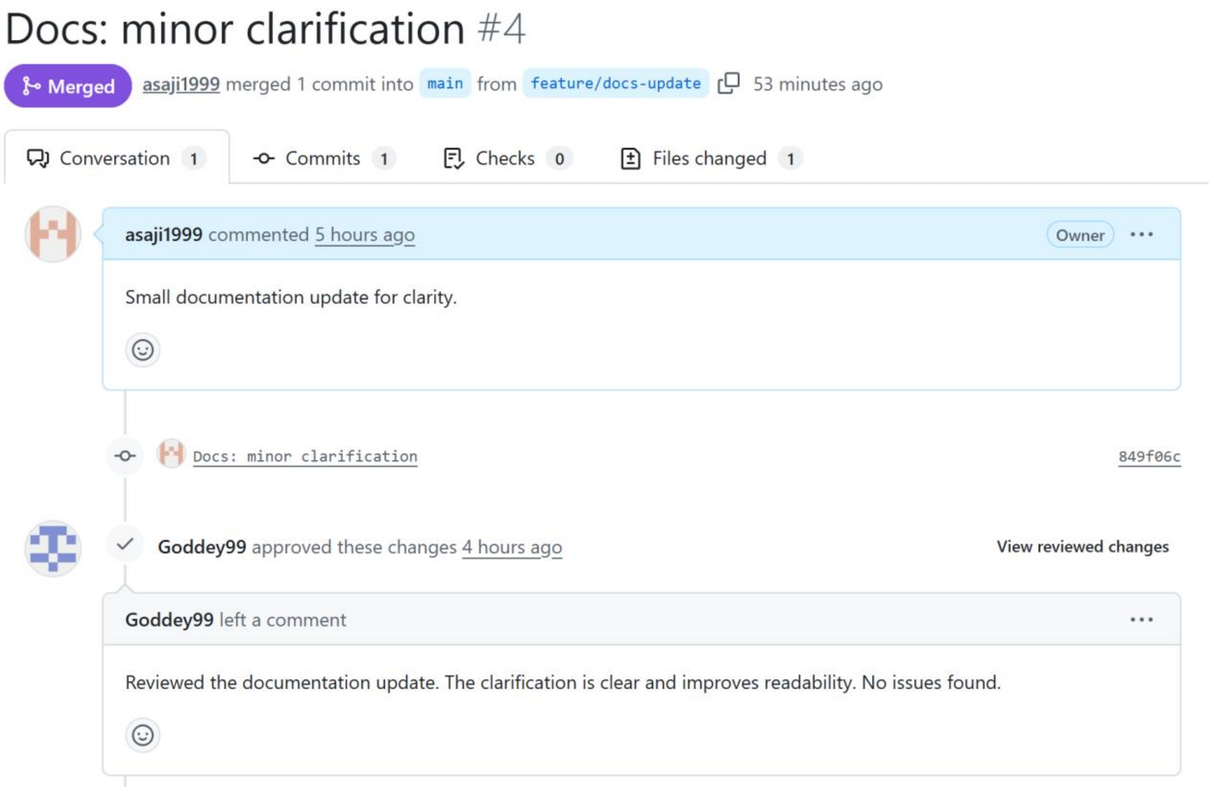
Docs: minor clarification 849f06c

Goddey99 approved these changes 4 hours ago

Goddey99 left a comment

Reviewed the documentation update. The clarification is clear and improves readability. No issues found.

## Peer Code Review:



Docs: minor clarification #4

Merged asaji1999 merged 1 commit into main from feature/docs-update 53 minutes ago

asaji1999 commented 5 hours ago

Small documentation update for clarity.

Docs: minor clarification 849f06c

Goddey99 approved these changes 4 hours ago

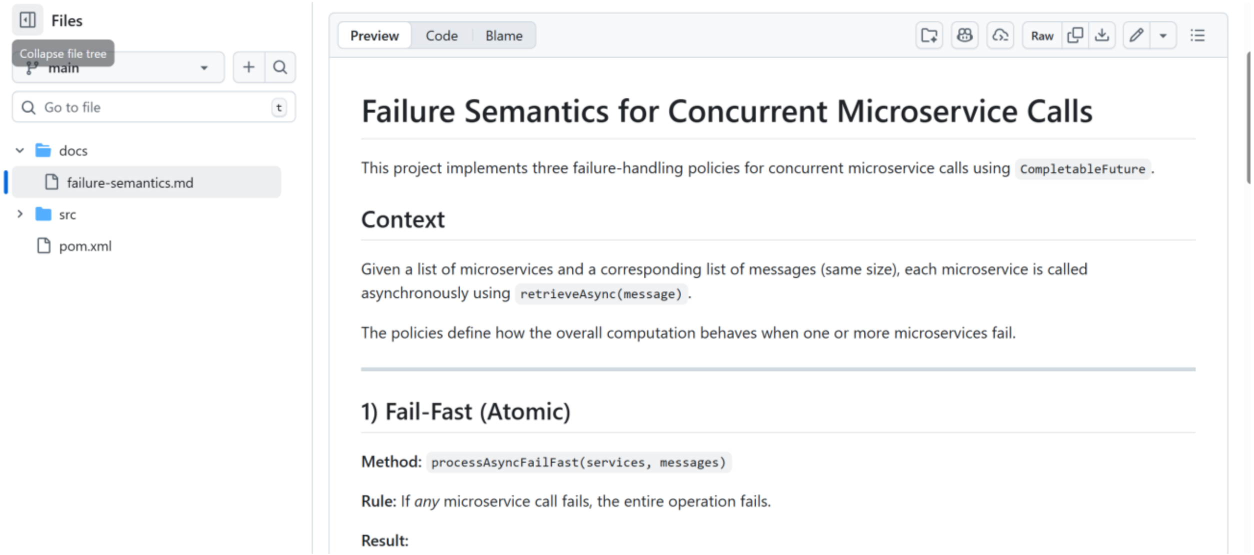
Goddey99 left a comment

Reviewed the documentation update. The clarification is clear and improves readability. No issues found.

Peer review was done on Pull Request Docs:Minor Clarification by Godred.

## Failure Semantics:

A failure-semantics.md file was created in the docs folder. Below is a snapshot of it:



# Failure Semantics for Concurrent Microservice Calls

This project implements three failure-handling policies for concurrent microservice calls using `CompletableFuture`.

## Context

Given a list of microservices and a corresponding list of messages (same size), each microservice is called asynchronously using `retrieveAsync(message)`.

The policies define how the overall computation behaves when one or more microservices fail.

## 1) Fail-Fast (Atomic)

**Method:** `processAsyncFailFast(services, messages)`

**Rule:** If *any* microservice call fails, the entire operation fails.

**Result:**

## AI Usage Claim:

The AI tool ChatGPT Version 5.2  was used for proofreading of failure-semaintics.md and guidance regarding markdown styling for .md files. Further it was used for reviewing of code and gaining insight on how to improve the code by the author. All code changes , decisions were done by the author whereas ChatGPT was used mostly as a learning aid and as a reviewing tool.

Prompts:

“For the code as given below, I need to differentiate between the three policies Fail-Fast, Fail-Partial, Fail-Soft. I have to create it in a .md file format , how should I structure it and inform me of the various markdown commands and tips to have a clean and neat .md report.”

“Explain fail-fast vs fail-partial vs fail-soft for the following assignment which I have uploaded as a pdf. How should I start , and what I should look out for each step.”