**Semaphore – API**

**Post**

'Should acquire a lock if no blocking semaphores exist.'

'Should reject when acquiring and another exclusive semaphore exist for different process key.'

'Should reject exclusive semaphore when shared exist and process Run Keys are different.'

'Should acquire exclusive semaphore when shared exist and process Run Keys are the same.'

'Should acquire multiple locks for exclusive semaphore for the same process key.'

**GET**

should fetch semaphores.

**Release**

should release semaphore.

**Renew**

should renew lock.

should return 404 when renewing non-existent semaphore.

'Should acquire a lock if no blocking semaphores exist.'

'Should reject when acquiring and another exclusive semaphore exist for different process key.'

**'Should allow multiple shared semaphores when limit is not exceeded.'**

'Should reject acquiring shared semaphore when concurrent Process Limit reached.'

'Should reject acquiring shared semaphore when setting limit below current number of semaphores and total number of processes exceed it'

'Should allow downsizing concurrent Process Limit when total number of processes doesn’t exceed it.'

'Should allow multiple shared semaphores without any limit.'

'Should reject exclusive semaphore when shared exist and process Run Keys are different.'

'Should acquire exclusive semaphore when shared exist and process Run Keys are the same.'

'Should acquire multiple locks for exclusive semaphore for the same process key.'

**Archive**

**should remove semaphores older than provided retention time.**

Fetch

should fetch semaphores.

Release

should release orphaned semaphores.

should release semaphore.

should return conflict when releasing non-existent semaphore.

Renew

should renew lock.

should return 404 when renewing non-existent semaphore.

**I. Feature: Semaphore API GET**

**1.Scenario: Retrieve Semaphore File Content**

**Given:** semaphore Is available

**When: get request is sent and fetch the semaphore detail** the request sends a GET request to retrieve the semaphore content

**Then:** the response status code should be 200

**And:** the response body should contain the content of the semaphore

**2.Scenario: Retrieve Semaphore Not Found**

**Given:** the semaphore does not exist

**When:** the client sends a GET request to retrieve the semaphore content

**Then:** the response status code should be 404

**And:** the response body should contain an error message indicating the file is not found

**3.Scenario: Retrieve Semaphore File Unauthorized**

**Given:** the client is not authorized to access the semaphore

**When**: the client sends a GET request to retrieve the semaphore file content

**Then**: the response status code should be 401

**And**: the response body should contain an error message indicating unauthorized access

**4. Scenario: Retrieve Semaphore File Forbidden**

**Given:** the client does not have permission to access the semaphore file

When: the client sends a GET request to retrieve the semaphore file content

Then: the response status code should be 403

And: the response body should contain an error message indicating forbidden access

**5.Scenario: Retrieve Semaphore File Server Error**

Given: the server encounters an internal error while retrieving the semaphore file

When: the client sends a GET request to retrieve the semaphore file content

Then: the response status code should be 500

And: the response body should contain an error message indicating server error

**II. Feature: Semaphore File API POST**

**1.Scenario: Create Semaphore File**

Given: the semaphore file does not exist

When: the client sends a POST request to create a semaphore file with valid content

Then: the response status code should be 201

And: the response body should contain a success message indicating the file was created successfully

**2.Scenario: Create Semaphore File Conflict**

Given: the semaphore file already exists

When: the client sends a POST request to create a semaphore file with valid content

Then: the response status code should be 409

And: the response body should contain an error message indicating the file already exists

**3.Scenario: Create Semaphore File Bad Request**

Given: the client sends a POST request with invalid content

When: the client sends a POST request to create a semaphore file

Then: the response status code should be 400

And: the response body should contain an error message indicating bad request

**4.Scenario: Create Semaphore File Unauthorized**

Given: the client is not authorized to create a semaphore file

When: the client sends a POST request to create a semaphore file

Then: the response status code should be 401

And: the response body should contain an error message indicating unauthorized access

**5.Scenario: Create Semaphore File Forbidden**

Given: the client does not have permission to create a semaphore file

When: the client sends a POST request to create a semaphore file

Then: the response status code should be 403

And: the response body should contain an error message indicating forbidden access

**6.Scenario: Create Semaphore File Server Error**

Given: the server encounters an internal error while creating a semaphore file

When: the client sends a POST request to create a semaphore file

Then: the response status code should be 500

And: the response body should contain an error message indicating server error

**III. Feature: Semaphore File API PUT (Renew)**

**1.Scenario: Renew Semaphore File**

Given: the semaphore file exists

When: the client sends a PUT request to renew the semaphore file

Then: the response status code should be 200

And: the response body should contain a success message indicating the file was renewed successfully

**2.Scenario: Renew Semaphore File Not Found**

Given: the semaphore file does not exist

When: the client sends a PUT request to renew the semaphore file

Then: the response status code should be 404

And: the response body should contain an error message indicating the file is not found

**3.Scenario: Renew Semaphore File Unauthorized**

Given: the client is not authorized to renew the semaphore file

When: the client sends a PUT request to renew the semaphore file

Then: the response status code should be 401

And: the response body should contain an error message indicating unauthorized access

**4.Scenario: Renew Semaphore File Forbidden**

Given: the client does not have permission to renew the semaphore file

When: the client sends a PUT request to renew the semaphore file

Then: the response status code should be 403

And: the response body should contain an error message indicating forbidden access

**5.Scenario: Renew Semaphore File Bad Request**

Given: the client sends a PUT request with invalid parameters

When: the client sends a PUT request to renew the semaphore file

Then: the response status code should be 400

And: the response body should contain an error message indicating bad request

**6.Scenario: Renew Semaphore File Server Error**

Given: the server encounters an internal error while renewing the semaphore file

When: the client sends a PUT request to renew the semaphore file

Then: the response status code should be 500

And: the response body should contain an error message indicating server error

**IV. Feature: Semaphore File API PUT (Release)**

**1.Scenario: Release Semaphore File**

Given: the semaphore file exists and is locked

When: the client sends a PUT request to release the semaphore file

Then: the response status code should be 200

And: the response body should contain a success message indicating the file was released successfully

**2.Scenario: Release Semaphore File Not Found**

Given: the semaphore file does not exist

When: the client sends a PUT request to release the semaphore file

Then: the response status code should be 404

And: the response body should contain an error message indicating the file is not found

**3.Scenario: Release Semaphore File Unauthorized**

Given: the client is not authorized to release the semaphore file

When: the client sends a PUT request to release the semaphore file

Then: the response status code should be 401

And: the response body should contain an error message indicating unauthorized access

**4.Scenario: Release Semaphore File Forbidden**

Given: the client does not have permission to release the semaphore file

When: the client sends a PUT request to release the semaphore file

Then: the response status code should be 403

And: the response body should contain an error message indicating forbidden access

**5. Scenario: Release Semaphore File Bad Request**

Given: the client sends a PUT request with invalid parameters

When: the client sends a PUT request to release the semaphore file

Then: the response status code should be 400

And: the response body should contain an error message indicating bad request

**6.Scenario: Release Semaphore File Server Error**

Given: the server encounters an internal error while releasing the semaphore file

When: the client sends a PUT request to release the semaphore file

Then: the response status code should be 500

And: the response body should contain an error message indicating server error