Test Cases for CKEditor Proxy REST API

/flush Endpoint

Test Case ID: FLUSH_01

- **Description:** Verify that the endpoint flushes the specified document session.
- **Preconditions:** A document with ID `test-document1` exists and has an active session.
- **Test Steps:**
    1. Send a GET request to `http://localhost:9001/flush?document_id=test-document1`.
- **Expected Result:** The session for the document `test-document1` is flushed, and the response status is 204 No Content.

Test Case ID: FLUSH_02

- **Description:** Verify that the endpoint returns an error when the document ID is missing.
- **Preconditions:** None.
- **Test Steps:**
    1. Send a GET request to `http://localhost:9001/flush`.
- **Expected Result:** The response status is 400 Bad Request with a message indicating that `document_id` is required.

Test Case ID: FLUSH_03

- **Description:** Verify that the endpoint returns an error when the document ID does not exist.
- **Preconditions:** A document with ID `nonexistent-document` does not exist.
- **Test Steps:**
    1. Send a GET request to `http://localhost:9001/flush?document_id=nonexistent-document`.
- **Expected Result:** The response status is 404 Not Found.

Test Case ID: FLUSH_04

- **Description:** Verify that the endpoint handles concurrent flush requests.
- **Preconditions:** A document with ID `test-document1` exists and has an active session.
- **Test Steps:**
    1. Send multiple concurrent GET requests to `http://localhost:9001/flush?document_id=test-document1`.
- **Expected Result:** All requests are processed, and the response status is 204 No Content for all.

Test Case ID: FLUSH_05

- **Description:** Verify that the endpoint returns an error for invalid document ID format.
- **Preconditions:** None.
- **Test Steps:**
    1. Send a GET request to `http://localhost:9001/flush?document_id=invalid!document@id`.
- **Expected Result:** The response status is 400 Bad Request.

/flushall Endpoint

Test Case ID: FLUSHALL_01

- **Description:** Verify that the endpoint flushes all active sessions.
- **Preconditions:** Multiple documents have active sessions.
- **Test Steps:**
    1. Send a GET request to `http://localhost:9001/flushall`.
- **Expected Result:** All active sessions are flushed, and the response status is 204 No Content.

Test Case ID: FLUSHALL_02

- **Description:** Verify that the endpoint handles cases when no sessions are active.

- **Preconditions:** No active sessions exist.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/flushall`.

- **Expected Result:** The response status is 204 No Content.

Test Case ID: FLUSHALL_03

- **Description:** Verify that the endpoint handles a large number of active sessions.
- **Preconditions:** A large number of documents have active sessions.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/flushall`.

- **Expected Result:** All active sessions are flushed, and the response status is 204 No Content.

Test Case ID: FLUSHALL_04

- **Description:** Verify that the endpoint is idempotent.
- **Preconditions:** Multiple documents have active sessions.
- **Test Steps:**

    1. Send multiple GET requests to `http://localhost:9001/flushall` in quick succession.

- **Expected Result:** All requests are processed, and the response status is 204 No Content for all.

/users Endpoint

Test Case ID: USERS_01

- **Description:** Verify that the endpoint returns the list of users for a specified document.
- **Preconditions:** A document with ID `test-document1` has active users.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/users?document_id=test-document1`.

- **Expected Result:** The response contains a list of users for `test-document1`.

Test Case ID: USERS_02

- **Description:** Verify that the endpoint returns an error when the document ID is missing.
- **Preconditions:** None.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/users`.

- **Expected Result:** The response status is 400 Bad Request with a message indicating that `document_id` is required.

Test Case ID: USERS_03

- **Description:** Verify that the endpoint returns an error when the document ID does not exist.
- **Preconditions:** A document with ID `nonexistent-document` does not exist.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/users?document_id=nonexistent-document`.

- **Expected Result:** The response status is 404 Not Found.

Test Case ID: USERS_04

- **Description:** Verify that the endpoint handles documents with no active users.
- **Preconditions:** A document with ID `test-document2` exists but has no active users.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/users?document_id=test-document2`.

- **Expected Result:** The response contains an empty list.

Test Case ID: USERS_05

- **Description:** Verify that the endpoint returns an error for invalid document ID format.

- **Preconditions:** None.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/users?document_id=invalid!document@id`.

- **Expected Result:** The response status is 400 Bad Request.

/comments Endpoint

Test Case ID: COMMENTS_01

- **Description:** Verify that the endpoint returns the list of comments.
- **Preconditions:** Comments exist in the system.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/comments`.

- **Expected Result:** The response contains a list of comments.

Test Case ID: COMMENTS_02

- **Description:** Verify that the endpoint returns an empty list when no comments exist.
- **Preconditions:** No comments exist in the system.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/comments`.

- **Expected Result:** The response contains an empty list.

Test Case ID: COMMENTS_03

- **Description:** Verify that the endpoint handles a large number of comments.
- **Preconditions:** A large number of comments exist in the system.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/comments`.

- **Expected Result:** The response contains a list of comments.

Test Case ID: COMMENTS_04

- **Description:** Verify that the endpoint filters comments by document ID.
- **Preconditions:** Comments exist for multiple documents.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/comments?document_id=test-document1`.

- **Expected Result:** The response contains a list of comments for `test-document1`.

Test Case ID: COMMENTS_05

- **Description:** Verify that the endpoint returns an error for invalid document ID format.
- **Preconditions:** None.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/comments?document_id=invalid!document@id`.

- **Expected Result:** The response status is 400 Bad Request.

/suggestions Endpoint

Test Case ID: SUGGESTIONS_01

- **Description:** Verify that the endpoint returns the list of suggestions.
- **Preconditions:** Suggestions exist in the system.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/suggestions`.

- **Expected Result:** The response contains a list of suggestions.

Test Case ID: SUGGESTIONS_02

- **Description:** Verify that the endpoint returns an empty list when no suggestions exist.
- **Preconditions:** No suggestions exist in the system.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/suggestions`.

- **Expected Result:** The response contains an empty list.

Test Case ID: SUGGESTIONS_03

- **Description:** Verify that the endpoint handles a large number of suggestions.
- **Preconditions:** A large number of suggestions exist in the system.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/suggestions`.

- **Expected Result:** The response contains a list of suggestions.

Test Case ID: SUGGESTIONS_04

- **Description:** Verify that the endpoint filters suggestions by document ID.
- **Preconditions:** Suggestions exist for multiple documents.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/suggestions?document_id=test-document1`.

- **Expected Result:** The response contains a list of suggestions for `test-document1`.

Test Case ID: SUGGESTIONS_05

- **Description:** Verify that the endpoint returns an error for invalid document ID format.
- **Preconditions:** None.
- **Test Steps:**

    1. Send a GET request to `http://localhost:9001/suggestions?document_id=invalid!document@id`.

- **Expected Result:** The response status is 400 Bad Request.