**POST tests:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Description** | **Test Steps** | **Test Data** | **Expected Results** |
| POST a message With all the parameters | 1. Create the message 2. Try and post it 3. Assert the response is 201 | {    "application\_id": 1,    "session\_id": "aaaa",    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 201 |
| POST a few message With all the parameters | 1. Send to a functions that creates a number of messages and post them 2. For all the messages we sent we check if the get by their id is returning 200 | 10 messages in the same format as:  {    "application\_id": 1,    "session\_id": "aaaa",    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | For all messages status should be 200 |
| Post a message with invalid application id | 1. Create a message with invalid application id 2. Post it 3. Assert the return status to be 400 | {    "application\_id": ‘invalid’,    "session\_id": "aaaa",    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 400 |
| Post a message with invalid session id | 1. Create a message with invalid session id 2. Post it 3. Assert the return status to be 400 | {    "application\_id": 1,    "session\_id": 1231,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 400 |
| Post a message with invalid message id | 1. Create a message with invalid message id 2. Post it 3. Assert the return status to be 400 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": 123 ,  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 400 |
| Post a message with invalid participants | 1. Create a message with invalid prticipants 2. Post it 3. Assert the return status to be 400 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": ‘bbbb’ ,  "participants": 123,  "content": "Hi, how are you today?"  } | Statuse\_code = 400 |
| Post a message with invalid contant | 1. Create a message with invalid contant 2. Post it 3. Assert the return status to be 400 | {    "application\_id": 1,    "session\_id": "aaaa",    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": 1323  } | Statuse\_code = 400 |
| Post a message with missing application id | 1. Create a message with missing application id 2. Post it 3. Assert the return status to be 400 | {    "session\_id": "aaaa",    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 400 |
| Post a message with missing session id | 1. Create a message with missing session id 2. Post it 3. Assert the return status to be 400 | {    "application\_id": 1,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 400 |
| Post a message with missing message id | 1. Create a message with missing message id 2. Post it 3. Assert the return status to be 400 | {    "application\_id": 1,    "session\_id": ‘aaaa’,  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 400 |
| Post a message with missing participants | 1. Create a message with missing participants 2. Post it 3. Assert the return status to be 400 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": ‘bbbb’ ,  "content": "Hi, how are you today?"  } | Statuse\_code = 400 |
| Post a message with missing contant | 1. Create a message with missing contant 2. Post it 3. Assert the return status to be 400 | {    "application\_id": 1,    "session\_id": "aaaa",    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"]  } | Statuse\_code = 400 |

**GET tests:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Description** | **Test Steps** | **Test Data** | **Expected Results** |
| GET message without POST using application id | 1. Try and get message using application id 2. Assert the response is 404 |  | Statuse\_code = 404 |
| GET message without POST using session id | 1. Try and get message using session id 2. Assert the response is 404 |  | Statuse\_code = 404 |
| GET message without POST using message id | 1. Try and get message using message id 2. Assert the response is 404 |  | Statuse\_code = 404 |
| Post a message and then GET the message using application id | 1. Create a message 2. Post it 3. Get message using the same application id 4. Assert the return status to be 200 5. Assert the return data is equal to the message posted | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 200  And the same message |
| Post a message and then GET the message using session id | 1. Create a message 2. Post it 3. Get message using the same session id 4. Assert the return status to be 200 5. Assert the return data is equal to the message posted | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 200  And the same message |
| Post a message and then GET the message using message id | 1. Create a message 2. Post it 3. Get message using the same message id 4. Assert the return status to be 200 5. Assert the return data is equal to the message posted | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 200  And the same message |
| Post a message and then GET the message using the wrong application id | 1. Create a message 2. Post it 3. Get message using the anther application id 4. Assert the return status to be 404 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  }  And GET using {applicationId : 2} | Statuse\_code = 404 |
| Post a message and then GET the message using the wrong session id | 1. Create a message 2. Post it 3. Get message using the anther session id 4. Assert the return status to be 404 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  }  And GET using {sessionId : ‘bbbb’} | Statuse\_code = 404 |
| Post a message and then GET the message using the wrong message id | 1. Create a message 2. Post it 3. Get message using the anther message id 4. Assert the return status to be 404 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  }  And GET using {message : ‘aaaa’} | Statuse\_code = 404 |
| POST a few messages with the same application id and then GET them all | 1. POST a few message with the same application id 2. Post it 3. Assert the return status to be 200 4. Assert the length of the response the same as all the messages we sent |  | Statuse\_code = 200  Length of response JSON like he data we POST |
| POST a few messages with the same session id and then GET them all | 1. POST a few message with the same session id 2. Post it 3. Assert the return status to be 200 4. Assert the length of the response the same as all the messages we sent |  | Statuse\_code = 200  Length of response JSON like he data we POST |
| Post four messages  3 has the same application id  And one with anther app id  3 has the same statuc id and one with anther | 1. Create four messages with missing contant 2. Post it 3. GET all the messages with application id and session id 4. Assert the return status to be 200 5. Assert the returned messages are the messages with the same session id and application id 6. Assert respons doesn’t contains the messages with different application and session id | {    "application\_id": 1,    "session\_id": "bbbb",    "message\_id": "a",  "participants": ["avi aviv", "moshe cohen"]  }  {    "application\_id": 1,    "session\_id": "aaaa",    "message\_id": "b",  "participants": ["avi aviv", "moshe cohen"]  }  {    "application\_id": 1,    "session\_id": "aaaa",    "message\_id": "c",  "participants": ["avi aviv", "moshe cohen"]  }  {    "application\_id": 2,    "session\_id": "aaaa",    "message\_id": "d",  "participants": ["avi aviv", "moshe cohen"]  } | Statuse\_code = 200  The data with the same application and session id |

**DELET tests**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Description** | **Test Steps** | **Test Data** | **Expected Results** |
| DELETE message without POST using application id | 1. Try and delete message using application id 2. Assert the response is 404 |  | Statuse\_code = 404 |
| DELETE message without POST using session id | 1. Try and delete message using session id 2. Assert the response is 404 |  | Statuse\_code = 404 |
| DELETE message without POST using message id | 1. Try and delete message using message id 2. Assert the response is 404 |  | Statuse\_code = 404 |
| Post a message and then DELETE the message using application id | 1. Create a message 2. Post it 3. DELETE message using the same application id 4. Assert the return status to be 200 5. Try GET with the application id 6. Assert the get response is 404 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 200  And get respons 404 |
| Post a message and then DELETE the message using session id | 1. Create a message 2. Post it 3. DELETE message using the same session id 4. Assert the return status to be 200 5. Try GET with the session id 6. Assert the get response is 404 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 200  And get respons 404 |
| Post a message and then DELETE the message using message id | 1. Create a message 2. Post it 3. DELETE message using the same message id 4. Assert the return status to be 200 5. Try GET with the message id 6. Assert the get response is 404 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  } | Statuse\_code = 200  And get respons 404 |
| Post a message and then DELETE the message using the wrong application id | 1. Create a message 2. Post it 3. DELETE message using the anther application id 4. Assert the return status to be 404 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  }  And GET using {applicationId : 2} | Statuse\_code = 404 |
| Post a message and then DELETE the message using the wrong session id | 1. Create a message 2. Post it 3. DELETE message using the anther session id 4. Assert the return status to be 404 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  }  And GET using {sessionId : ‘bbbb’} | Statuse\_code = 404 |
| Post a message and then DELETE the message using the wrong message id | 1. Create a message 2. Post it 3. DELETE message using the anther message id 4. Assert the return status to be 404 | {    "application\_id": 1,    "session\_id": ‘aaaa’,    "message\_id": "bbbb",  "participants": ["avi aviv", "moshe cohen"],  "content": "Hi, how are you today?"  }  And GET using {message : ‘aaaa’} | Statuse\_code = 404 |
| POST a few messages with the same application id and then DELTE them all | 1. POST a few message with the same application id 2. Post it 3. DELETE with parms of the same application id 4. Assert the return status to be 200 5. GET the messages with the same application id and assert response 404 |  | Statuse\_code = 200  GET response =404 |
| POST a few messages with the same session id and then DELTE them all | 1. POST a few message with the same session id 2. Post it 3. DELETE with parms of the same session id 4. Assert the return status to be 200 5. GET the messages with the same session id and assert response 404 |  | Statuse\_code = 200  GET response =404 |
| Post four messages  3 has the same application id  And one with anther app id  3 has the same statuc id and one with anther and DELTE the ones with the same application and session id | 1. Create four messages with missing contant 2. Post it 3. DELETE all the messages with application id and session id 4. Assert the return status to be 200 5. GET all the messages 6. Assert the messages with the same application and session id are not in the returned messages 7. Assert that the ones without dame application and session id are in the response messages | {    "application\_id": 1,    "session\_id": "bbbb",    "message\_id": "a",  "participants": ["avi aviv", "moshe cohen"]  }  {    "application\_id": 1,    "session\_id": "aaaa",    "message\_id": "b",  "participants": ["avi aviv", "moshe cohen"]  }  {    "application\_id": 1,    "session\_id": "aaaa",    "message\_id": "c",  "participants": ["avi aviv", "moshe cohen"]  }  {    "application\_id": 2,    "session\_id": "aaaa",    "message\_id": "d",  "participants": ["avi aviv", "moshe cohen"]  } | Statuse\_code = 200 |

**DB tests:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Description** | **Test Steps** | **Test Data** | **Expected Results** |
| Test create Message.db | 1. Create the table 2. Try insert into the table |  | 1. SQLITE ok |
| Test insert to db without running db | 1. Try and insert 2. Get respons |  |  |
| Test get all messages | 1. insert messages into the db 2. Get all messages and check if they all returned |  | SQLITE ok |
| Test insert two messages with the same message id | 1. Insert one message 2. Insert the same message |  |  |
| Test delete message that is not in the db | 1. Delete a message |  |  |
| Test delete a message that is in the db | 1. Add a message 2. Delete the message |  |  |
| Test add message with missing data | Try to add message with missing data |  |  |
|  |  |  |  |