TestLink Community [configure $tlCfg->document\_generator->company\_name]

Test Specification

Test Project: MingleQA  
Test Suite: MingleQA - SCRMCHAT-341

Printed by TestLink on 12/07/2018

2012 © TestLink Community

# Table Of Contents

[**1.SCRMCHAT-341**](#toc_1)

[MingleQA-22001: To get Authorization token](#toc_tc210985)

[MingleQA-21754: Create Group](#toc_tc205122)

[MingleQA-21755: User Groups](#toc_tc205124)

[MingleQA-21756: Update Group](#toc_tc205126)

[MingleQA-21757: Create Topic](#toc_tc205128)

[MingleQA-21758: Group Topics](#toc_tc205130)

[MingleQA-21759: Update Topic](#toc_tc205132)

[MingleQA-21760: PostTopicMessage](#toc_tc205134)

[MingleQA-21761: GetTopicMessages](#toc_tc205136)

[MingleQA-21762: PostTeammateMessage](#toc_tc205138)

[MingleQA-21763: GetTeammateMessages](#toc_tc205140)

[MingleQA-21877: Delete Group](#toc_tc206067)

# 1.Test Suite : SCRMCHAT-341

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-22001: To get Authorization token [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the Endpoint for JWT access\_token generated from Authorization service.    https://mingledev01-authservice.mingledev.infor.com/authorizationserver/as/v1/token    End Point : {{AuthURL}}/as/v1/token    Body : x-www-form-urlencoded     client\_id : {{clientId}}    client\_secret: {{clientSecret}}    grant\_type : client\_credentials  Note : authUrl, clientId and clientSecret depends upon the environment used for testing.    Tests :    tests["Status code should be 200"] = responseCode.code === 200;    try  {    var jsonData = JSON.parse(responseBody);    postman.setEnvironmentVariable("access\_token", jsonData.access\_token);  }  catch(e)  {      console.log("failed to get the access token" + e);  }    Method : POST | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL and Tests mentioned in the above summary and select the method as "POST" 3. Select the "x-www-form-urlencoded" radio button under body in postman. 4. Enter the "client\_id", "client\_secret", "grant\_type" values mentioned in the summary. 5. Click on Send button. 6. Validate the Tests mentioned in summary. | User should be able to open postman tool.  Body option will be enabled.  Grid view should be displayed to enter the request data.  User should able to enter the mentioned data in body.  Getting Response as below with 200 OK response code   {    "access\_token": "eyJhbGciOiJSUzUxMiIsImtpZCI6IjAwOWYifQ.eyJqdGkiOiJUUXN5cDFGUGdrZVo4dkFmLTQxYm13IiwiaWF0IjoxNTEwNTY0Mjk2LCJuYmYiOjE1MTA1NjQyOTYsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Q6ODA4MC9hdXRob3JpemF0aW9uc2VydmVyL3Jlc3QiLCJleHAiOjE1MTA1NjQ4OTYsImNsaWVudGlkIjoiMzQ5NjJjOTU3YzRjMDk2ZmJkMDQxYTRhNWY5NmFhYmEiLCJhdXRob3JpemF0aW9uIjp7InJlZ2lzdHJ5c3ZjIjp7InBlcm1pc3Npb25zIjpbInJlYWQiXX0sIm1lc2luZGV4c3ZjIjp7InBlcm1pc3Npb25zIjpbIndyaXRlIl19LCJ0ZW5hbnRjb25maWdzdmMiOnsicGVybWlzc2lvbnMiOlsicmVhZCJdfX0sImNsaWVudHR5cGUiOiJtZXNjb25maWdzdmMifQ.I67uRVsLfT6JdvYjlB7ZgRsf\_Ri-HBtH2e3fV8nMwKExkXQVo3wKdT8KuUZnEoZ6lwwwj-mpXjz\_DAx7uOcMAbLu\_8RR3umX4wgYwbGQ-AB4cCQAwjBjpzXigan2D7\_bacM0UcGDoL63hCcy9T0nrOBQttoFzBdehaFFHLr782cyLbe6EWckN6F2JP4WUO-6\_nPLI5iPvkjQjbmkYDrZlRv7nWVfYDsxJ7UqXmkoT3UUI9\_pa7ur0EsL6T3wc-3pdJNuLyiAzqGSrTSemnzblGrD\_wKCwCUtjLg8XvUSt\_5MR6n-8CD04y85ADG9oxfgayA6PwOEaMozcBsbh5jghA",    "token\_type": "Bearer",    "expires\_in": 600  }  All the mentioned tests should be passed |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: |  | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21754: Create Group [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the endpoint to Create a group  {{gatewayURL}}/api/v1/admin/groups  Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.  Request Body (Post Method)  {"name":"GRP {{$timestamp}}","groupType":"PUBLIC","visibility":false,"description":"Group Description","uploadIdentifier":"","members":[{"memberId":"{{GroupAdmin}}","role":"ADMIN"},{"memberId":"{{GroupMember1}}","role":"MEMBER"}]}  Tests:  try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["id not null"] = data.id !== (null) ;  tests["name is not null"] = data.name !== (null) ;  console.log("name" + (data.name));  tests["description is not null"] = data.description !== (null) ;  console.log("description" + (data.description));  tests["group type not null"] = data.grouptype !==  (null) ;  tests["visibility not null"] = data.visibility !== (null);  tests["ifsdistributiongroup not null"] = data.ifsdistributiongroup !== (null);  tests["groupguid is not null"] = data.groupguid !== (null) ;  tests["status is not null"] = data.status !== (null) ;  tests["updatedon is not null"] = data.updatedon !== (null) ;  tests["createdon is not null"] = data.createdon !== (null) ;  tests["profilephoto is not null"] = data.profilephoto !== (null) ;  tests["cdnurl is not null"] = data.cdnurl !== (null) ;  tests["groupcreator is not null"] = data.groupcreator !== (null) ;  tests["latestactivity is not null"] = data.latestactivity !== (null) ;  tests["badgecount is not null"] = data.badgecount !== (null) ;  tests["groupuserrole is not null"] = data.groupuserrole !== (null) ;  tests["groupuserstatus is not null"] = data.groupuserstatus !== (null) ;  pm.environment.set("groupguid", data.groupguid);      }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers, body and Method. 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK . This endpoint should create a new group in chat 4. Mentioned tests in summary should pass with out any errors     {      "id": 40182,      "name": "GG vvvv1513833237",      "description": "Group Description",      "groupType": "PUBLIC",      "visibility": true,      "ifsDistributionGroup": false,      "groupguid": "05199A83-F080-4BF7-955F-64EC8368C536",      "status": "ACTIVE",      "updatedOn": 1513833237353,      "createdOn": 0,      "profilePhoto": false,      "cdnURL": null,      "groupCreator": null,      "createdby": 209,      "latestActivity": 0,      "badgeCount": 0,      "groupUserRole": "ADMIN",      "groupUserStatus": "JOINED"  } |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21755: User Groups [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the Endpoint to get all user groups   {{gatewayURL}}/api/v1/user/groups    Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.    Method : Get    Tests:    try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["body contains id"] = responseBody.has("id");  tests["id not null"] = data.groupHeaders[0].id !== (null) ;  tests["name is not null"] = data.groupHeaders[0].name !== (null) ;  console.log("name" + (data.groupHeaders[0].name));  tests["description is not null"] = data.groupHeaders[0].description !== (null) ;  console.log("description" + (data.groupHeaders[0].description));  tests["group type not null"] = data.groupHeaders[0].grouptype !==  (null) ;  tests["visibility not null"] = data.groupHeaders[0].visibility !== (null);  tests["ifsdistributiongroup not null"] = data.groupHeaders[0].ifsdistributiongroup !== (null);  tests["groupguid is not null"] = data.groupHeaders[0].groupguid !== (null) ;  tests["status is not null"] = data.groupHeaders[0].status !== (null) ;  tests["updatedon is not null"] = data.groupHeaders[0].updatedon !== (null) ;  tests["createdon is not null"] = data.groupHeaders[0].createdon !== (null) ;  tests["profilephoto is not null"] = data.groupHeaders[0].profilephoto !== (null) ;  tests["cdnurl is not null"] = data.groupHeaders[0].cdnurl !== (null) ;  tests["groupcreator is not null"] = data.groupHeaders[0].groupcreator !== (null) ;  tests["latestactivity is not null"] = data.groupHeaders[0].latestactivity !== (null) ;  tests["badgecount is not null"] = data.groupHeaders[0].badgecount !== (null) ;  tests["groupuserrole is not null"] = data.groupHeaders[0].groupuserrole !== (null) ;  tests["groupuserstatus is not null"] = data.groupHeaders[0].groupuserstatus !== (null) ;    }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers, and Get Method. 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK  and display all the group details 4. Mentioned tests in summary should pass with out any errors   {      "groupCount": 104,      "groupHeaders": [          {              "id": 40182,              "name": "GG vvvv1513833237",              "description": "Sample Group 2677",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "05199A83-F080-4BF7-955F-64EC8368C536",              "status": "ACTIVE",              "updatedOn": 1513833257833,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 40179,              "name": "GG 1513823482",              "description": "Sample Group 267",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "992D9477-43D3-46F8-9968-E34C530830A0",              "status": "ACTIVE",              "updatedOn": 1513833168877,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10075,              "name": "Chat &amp; Distribution Group",              "description": "Desc",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": true,              "groupguid": "5330F4E3-E29B-4625-AAE8-4DF6B5148E3C",              "status": "ACTIVE",              "updatedOn": 1510223073060,              "createdOn": 0,              "profilePhoto": true,              "cdnURL": null,              "groupCreator": null,              "createdby": 239,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "MEMBER",              "groupUserStatus": "JOINED"          },          {              "id": 30151,              "name": "UI Test",              "description": "sssss",              "groupType": "PRIVATE",              "visibility": false,              "ifsDistributionGroup": false,              "groupguid": "4AAEB1B8-6B4C-4AE8-9CDF-8AEDEC839651",              "status": "ACTIVE",              "updatedOn": 1513518323877,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 203,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "MEMBER",              "groupUserStatus": "JOINED"          },          {              "id": 40154,              "name": "Demo Sprint",              "description": "Demo",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "997E7704-4D0D-4406-8AD9-B03DB6921757",              "status": "ACTIVE",              "updatedOn": 1513579383520,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10154,              "name": "GG 1513220437",              "description": "Group Description",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "4E203513-78E5-4427-99A5-B30298F92DCA",              "status": "ACTIVE",              "updatedOn": 1513220437827,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10149,              "name": "GRP 1513071116",              "description": "Group Desc update",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "36A5ADE1-37E7-44B2-A742-1962CE156AF2",              "status": "ACTIVE",              "updatedOn": 1513071119700,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10148,              "name": "GRP 1512975137",              "description": "Group Desc update",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "48A715A7-99E6-447C-8D55-5C729FE2A0E8",              "status": "ACTIVE",              "updatedOn": 1512975139677,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10147,              "name": "GRP 1512974939",              "description": "Group Desc update",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "4C89C9EA-FE17-46B4-A0A1-8E99F2514DBF",              "status": "ACTIVE",              "updatedOn": 1512974942160,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10144,              "name": "GRP 1512974434",              "description": "Group Desc update",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "EC2D3E5D-284F-40E3-ABAA-C6CC420FA26F",              "status": "ACTIVE",              "updatedOn": 1512974437083,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10143,              "name": "GRP 1512970534",              "description": "Desc 1512970536",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "642D0679-C527-4B74-A248-7D4169F3254F",              "status": "ACTIVE",              "updatedOn": 1512970536407,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10142,              "name": "GRP 1512970423",              "description": "Desc 1512970424",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "8BAAC9C8-5853-459F-BF7C-4A59ACB5956E",              "status": "ACTIVE",              "updatedOn": 1512970424633,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10141,              "name": "GRP 1512970370",              "description": "Desc 1512970373",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "2D636D60-A940-4B1B-9C03-1514BD1BF848",              "status": "ACTIVE",              "updatedOn": 1512970373243,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10140,              "name": "GRP 1512969558",              "description": "Desc 1512969561",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "1966233A-16C4-4414-9351-917F688103BE",              "status": "ACTIVE",              "updatedOn": 1512969561507,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10139,              "name": "GRP 1512969270",              "description": "Desc 1512969273",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "261B9F03-BD18-466B-BB22-8E55004BCFAB",              "status": "ACTIVE",              "updatedOn": 1512969273207,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10138,              "name": "GRP 1512969184",              "description": "Desc 1512969186",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "2D5AFF79-4DA3-4DB9-B049-5DCAD3B9268C",              "status": "ACTIVE",              "updatedOn": 1512969186877,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10137,              "name": "GRP 1512968983",              "description": "Desc 1512968985",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "76F0D9B6-7927-45A1-A8C5-096553BCB5F8",              "status": "ACTIVE",              "updatedOn": 1512968985373,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10136,              "name": "GRP 1512968802",              "description": "Desc 1512968805",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "0CDCD764-CCF9-42F4-A4C3-7C0E94C56126",              "status": "ACTIVE",              "updatedOn": 1512968805163,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10135,              "name": "GRP 1512968578",              "description": "Desc 1512968580",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "B0A5A6A2-4CCA-4DA5-B0A7-14CA5D9C46BE",              "status": "ACTIVE",              "updatedOn": 1512968581007,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10134,              "name": "GRP 1512968319",              "description": "Desc 1512968322",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "E96CB1C4-55F2-4FD8-8B43-006A14E01009",              "status": "ACTIVE",              "updatedOn": 1512968322680,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10133,              "name": "GRP 1512968152",              "description": "Desc 1512968155",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "9C893DD5-2FFA-4B00-ADDB-42C2E8324B15",              "status": "ACTIVE",              "updatedOn": 1512968155673,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10132,              "name": "GRP 1512968085",              "description": "Desc 1512968087",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "CBAAF11E-89CA-40E0-9632-0D7B1C5F678D",              "status": "ACTIVE",              "updatedOn": 1512968087913,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10131,              "name": "GRP 1512968014",              "description": "Desc 1512968016",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "45C483F0-DDAE-4D0B-A707-3B07F3CC5126",              "status": "ACTIVE",              "updatedOn": 1512968016833,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10130,              "name": "GRP 1512967684",              "description": "Desc 1512967686",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "BA3F59E5-CDF8-431F-8929-50FD10DC5267",              "status": "ACTIVE",              "updatedOn": 1512967686647,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10129,              "name": "GRP 1512966520",              "description": "Sample Group 233",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "3F28801C-55FF-475C-BDC2-0BBB4731297C",              "status": "ACTIVE",              "updatedOn": 1512966521723,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10128,              "name": "GRP 1512966432",              "description": "Sample Group 233",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "D94B8809-A255-4A2F-B2FC-91EEABC70A33",              "status": "ACTIVE",              "updatedOn": 1512966434030,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10127,              "name": "GRP 1512966005",              "description": "Sample Group 233",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "E20A09BE-E91D-4557-8A04-A6BB18C86E8B",              "status": "ACTIVE",              "updatedOn": 1512966007087,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10126,              "name": "GRP 1512965771",              "description": "Desc 1512967558",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "3AE664C3-5C81-4A3F-BCD8-6205EA0553C7",              "status": "ACTIVE",              "updatedOn": 1512967559247,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10125,              "name": "GRP 1512965509",              "description": "Sample Group 233",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "3AAB755A-3C44-4CA5-A201-73E1B2256DD8",              "status": "ACTIVE",              "updatedOn": 1512965512020,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10119,              "name": "GRP 1512964313",              "description": "Sample Group1",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "A2C4E35C-B092-4D7B-9AEE-7C691AA68FCF",              "status": "ACTIVE",              "updatedOn": 1512964313740,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10114,              "name": "GRP 1512715171",              "description": "Sample Group1",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "59F4B7AD-E62A-4114-B1F1-7860355FE00C",              "status": "ACTIVE",              "updatedOn": 1512715171587,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          },          {              "id": 10112,              "name": "GG11",              "description": "Sample Group1",              "groupType": "PUBLIC",              "visibility": true,              "ifsDistributionGroup": false,              "groupguid": "2893B595-4D37-4C61-B6A0-7D2B6CA5122E",              "status": "ACTIVE",              "updatedOn": 1512714561400,              "createdOn": 0,              "profilePhoto": false,              "cdnURL": null,              "groupCreator": null,              "createdby": 209,              "latestActivity": 0,              "badgeCount": 0,              "groupUserRole": "ADMIN",              "groupUserStatus": "JOINED"          }      ]  } |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21756: Update Group [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the endpoint to Update a group  {{gatewayURL}}/api/v1/admin/groups/{{groupguid}}    Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.    Method : Post    Request Body:  {"groupType":"PUBLIC","visibility":true,"description":"Group Desc update","members":[{"memberId":"{{GroupMember1}}","role":"MEMBER"},{"memberId":"{{GroupAdmin}}","role":"ADMIN"}],"addMembers":[{"memberId":"41684ED3-9854-42AD-9E22-F1FCB7E1496E","role":"MEMBER"}],"updateMembers":[],"removeMembers":[]}    Tests:  try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["id not null"] = data.id !== (null) ;  tests["name is not null"] = data.name !== (null) ;  console.log("name" + (data.name));  tests["description is not null"] = data.description !== (null) ;  console.log("description" + (data.description));  tests["group type not null"] = data.grouptype !==  (null) ;  tests["visibility not null"] = data.visibility !== (null);  tests["ifsdistributiongroup not null"] = data.ifsdistributiongroup !== (null);  tests["groupguid is not null"] = data.groupguid !== (null) ;  tests["status is not null"] = data.status !== (null) ;  tests["updatedon is not null"] = data.updatedon !== (null) ;  tests["createdon is not null"] = data.createdon !== (null) ;  tests["profilephoto is not null"] = data.profilephoto !== (null) ;  tests["cdnurl is not null"] = data.cdnurl !== (null) ;  tests["groupcreator is not null"] = data.groupcreator !== (null) ;  }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers, body and Post Method. 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK and display the updated Group description. 4. Mentioned tests in summary should pass with out any errors     {      "id": 40182,      "name": "GG vvvv1513833237",      "description": "Sample Group 2677",      "groupType": "PUBLIC",      "visibility": true,      "ifsDistributionGroup": false,      "groupguid": "05199A83-F080-4BF7-955F-64EC8368C536",      "status": "ACTIVE",      "updatedOn": 1513833237353,      "createdOn": 1513833257833,      "profilePhoto": false,      "cdnURL": null,      "groupCreator": "Sudhakara vintha",      "createdby": 209  } |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21757: Create Topic [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the endpoint to create a topic  {{gatewayURL}}/api/v1/admin/groups/{{groupguid}}/topics    Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.    Method : Post    Request Body :  {"topicDescription":"Sample Topic 2","topicMembers":["{{GroupAdmin}}","{{GroupMember1}}"],"topicName":"Sample Topic","addAllGroupMembers":false}    Tests:  try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["topicId not null"] = data.topicid !== (null) ;  tests["topicName not null"] = data.topicName !== (null) ;  console.log("topicName" + (data.topicName));  tests["topicDescription not null"] = data.topicDescription !== (null) ;  console.log("topicDescription" + (data.topicDescription));  tests["topicType not null"] = data.topicType !== (null) ;  tests["topicStatus not null"] = data.topicStatus !== (null) ;  tests["topicGuid not null"] = data.topicGuid !== (null) ;  tests["groupId not null"] = data.groupId !== (null) ;  tests["groupGuid not null"] = data.groupGuid !== (null) ;  tests["groupName not null"] = data.groupName !== (null) ;  tests["createdon not null"] = data.createdon !== (null) ;  tests["updatedon not null"] = data.updatedon !== (null) ;  pm.environment.set("topicguid", data.topicGuid);    }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers, body and Post Method. 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK and create a new topic 4. Mentioned tests in summary should pass with out any errors     {      "topicId": 40438,      "topicName": "Sample Topic 7",      "topicDescription": "Sample Topic",      "topicType": "OTHER",      "topicStatus": "ACTIVE",      "topicGuid": "2792DB37-A5C2-4D21-840F-481C93C52051",      "groupId": 40179,      "groupGuid": "992D9477-43D3-46F8-9968-E34C530830A0",      "groupName": "GG 1513823482",      "createdon": 1513823549113,      "updatedon": 1513823549113  } |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21758: Group Topics [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the endpoint to get all the Group topics  {{gatewayURL}}/api/v1/user/groups/{{groupguid}}/topics    Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.    Method : Get    Tests :    try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["body contains id"] = responseBody.has("id");  tests["id not null"] = data.groupHeader.id !== (null) ;  tests["name is not null"] = data.groupHeader.name !== (null) ;  tests["description is not null"] = data.groupHeader.description !==  (null);  tests["group type not null"] = data.groupHeader.grouptype !==  (null) ;  tests["visibility not null"] = data.groupHeader.visibility !== (null);  tests["ifsdistributiongroup not null"] = data.groupHeader.ifsdistributiongroup !== (null);  tests["groupguid is not null"] = data.groupHeader.groupguid !== (null) ;  tests["status is not null"] = data.groupHeader.status !== (null) ;  tests["updatedon is not null"] = data.groupHeader.updatedon !== (null) ;  tests["createdon is not null"] = data.groupHeader.createdon !== (null) ;  tests["profilephoto is not null"] = data.groupHeader.profilephoto !== (null) ;  tests["cdnurl is not null"] = data.groupHeader.cdnurl !== (null) ;  tests["groupcreator is not null"] = data.groupHeader.groupcreator !== (null) ;  tests["latestactivity is not null"] = data.groupHeader.latestactivity !== (null) ;  tests["badgecount is not null"] = data.groupHeader.badgecount !== (null) ;  tests["groupuserrole is not null"] = data.groupHeader.groupuserrole !== (null) ;  tests["groupuserstatus is not null"] = data.groupHeader.groupuserstatus !== (null) ;  tests["topic1 is not null"] = data.topicCount !== (null) ;  tests["General not null"] = data.topicList[0].topicName !== (null) ;  console.log("General" + (data.topicList[0].topicName));  tests["topic2 is not null"] = data.topicCount !== (null) ;  tests["HelpName not null"] = data.topicList[1].topicName !== (null) ;  console.log("Help" + (data.topicList[1].topicName));  tests["topic3 is not null"] = data.topicCount !== (null);  tests["Topic3 not null"] = data.topicList[0].topicName !== (null) ;  console.log("Topic3" + (data.topicList[2].topicName));  tests["topicDescription not null"] = data.topicList[2].topicDescription !== (null) ;  console.log("topicDescription" + (data.topicList[2].topicDescription));  tests["topicType not null"] = data.topicList[2].topicType !== (null) ;  tests["topicStatus not null"] = data.topicList[2].topicStatus !== (null) ;  tests["topicGuid not null"] = data.topicList[2].topicGuid !== (null) ;  tests["groupId not null"] = data.topicList[2].groupId !== (null) ;  tests["topicgroupGuid not null"] = data.topicList[2].groupGuid !== (null) ;  tests["groupName not null"] = data.topicList[2].groupName !== (null) ;  tests["createdon not null"] = data.topicList[2].createdon !== (null) ;  tests["updatedon not null"] = data.topicList[2].updatedon !== (null) ;  tests["latestActivity not null"] = data.topicList[2].latestActivity !== (null) ;  tests["badgeCount not null"] = data.topicList[2].badgeCount !== (null) ;  tests["topicUserStatus not null"] = data.topicList[2].topicUserStatus !== (null) ;  tests["lastVisitedOn not null"] = data.topicList[2].lastVisitedOn !== (null) ;  tests["topicUserCount not null"] = data.topicList[2].topicUserCount !== (null) ;  }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers and Method. 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK  and display all topics in the group 4. Mentioned tests in summary should pass with out any errors     {      "groupHeader": {          "id": 40179,          "name": "GG 1513823482",          "description": "Sample Group 263",          "groupType": "PUBLIC",          "visibility": true,          "ifsDistributionGroup": false,          "groupguid": "992D9477-43D3-46F8-9968-E34C530830A0",          "status": "ACTIVE",          "updatedOn": 1513823538150,          "createdOn": 0,          "profilePhoto": false,          "cdnURL": null,          "groupCreator": null,          "createdby": 209,          "latestActivity": 0,          "badgeCount": 0,          "groupUserRole": "ADMIN",          "groupUserStatus": "JOINED"      },      "topicCount": 3,      "topicList": [          {              "topicId": 40436,              "topicName": "General",              "topicDescription": "General announcements and group notifications",              "topicType": "GENERAL",              "topicStatus": "ACTIVE",              "topicGuid": "A8687769-4BF2-42C4-B2C0-80C0BD0405B7",              "groupId": 40179,              "groupGuid": "992D9477-43D3-46F8-9968-E34C530830A0",              "groupName": "GG 1513823482",              "createdon": 0,              "updatedon": 0,              "latestActivity": 0,              "badgeCount": 0,              "topicUserStatus": "SUBSCRIBED",              "lastVisitedOn": 1513823481867,              "topicUserCount": 3          },          {              "topicId": 40437,              "topicName": "Help Desk",              "topicDescription": "Questions about this group",              "topicType": "HELP",              "topicStatus": "ACTIVE",              "topicGuid": "3F413EBB-2274-4136-99DF-E7570BC8BACB",              "groupId": 40179,              "groupGuid": "992D9477-43D3-46F8-9968-E34C530830A0",              "groupName": "GG 1513823482",              "createdon": 0,              "updatedon": 0,              "latestActivity": 0,              "badgeCount": 0,              "topicUserStatus": "SUBSCRIBED",              "lastVisitedOn": 1513823481867,              "topicUserCount": 3          },          {              "topicId": 40438,              "topicName": "Sample Topic 7",              "topicDescription": "Sample Topic",              "topicType": "OTHER",              "topicStatus": "ACTIVE",              "topicGuid": "2792DB37-A5C2-4D21-840F-481C93C52051",              "groupId": 40179,              "groupGuid": "992D9477-43D3-46F8-9968-E34C530830A0",              "groupName": "GG 1513823482",              "createdon": 0,              "updatedon": 0,              "latestActivity": 0,              "badgeCount": 0,              "topicUserStatus": "SUBSCRIBED",              "lastVisitedOn": 1513823549113,              "topicUserCount": 1          }      ]  } |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21759: Update Topic [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the endpoint to Update a topic  {{gatewayURL}}/api/v1/admin/groups/{{groupguid}}/topics/{{topicguid}}    Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.    Method : Post    {"topicDescription":"Topic Desc Update1","topicMembers":["{{GroupAdmin}}","{{GroupMember1}}"],"addAllGroupMembers":false}    Tests :  try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["topicId not null"] = data.topicid !== (null) ;  tests["topicName not null"] = data.topicName !== (null) ;  console.log("topicName" + (data.topicName));  tests["topicDescription not null"] = data.topicDescription !== (null) ;  console.log("topicDescription" + (data.topicDescription));  tests["topicType not null"] = data.topicType !== (null) ;  tests["topicStatus not null"] = data.topicStatus !== (null) ;  tests["topicGuid not null"] = data.topicGuid !== (null) ;  tests["groupId not null"] = data.groupId !== (null) ;  tests["groupGuid not null"] = data.groupGuid !== (null) ;  tests["groupName not null"] = data.groupName !== (null) ;  tests["createdon not null"] = data.createdon !== (null) ;  tests["updatedon not null"] = data.updatedon !== (null) ;  }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers, body and Post Method. 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK and update the topic description. 4. Mentioned tests in summary should pass with out any errors     {      "topicId": 40438,      "topicName": "Sample Topic 7",      "topicDescription": "Sample Topic 42",      "topicType": "OTHER",      "topicStatus": "ACTIVE",      "topicGuid": "2792DB37-A5C2-4D21-840F-481C93C52051",      "groupId": 40179,      "groupGuid": "992D9477-43D3-46F8-9968-E34C530830A0",      "groupName": "GG 1513823482",      "createdon": 1513823549113,      "updatedon": 1513823910543  } |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21760: PostTopicMessage [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the endpoint to PostTopicMessage  {{gatewayURL}}/api/v1/topic/{{topicguid}}/messages    Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.    Method : Post    {"recipient":"{{GroupMember1}}","message":"hi","documentIds":[],"base64Audio":"","taggedUserGuids":[]}    Tests :  try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["body contains msgId"] = responseBody.has("msgId");  tests["msgId is not null"] = data.chatMessage.msgId !== (null) ;  tests["msgText is not null"] = data.chatMessage.msgText !== (null) ;  console.log("msgText" + (data.chatMessage.msgText));  tests["postedBy is not null"] = data.chatMessage.postedBy !== (null) ;  tests["id is not null"] = data.chatMessage.postedUser.id !== (null) ;  tests["userType is not null"] = data.chatMessage.postedUser.userType !== (null) ;  tests["userIdentifier is not null"] = data.chatMessage.postedUser.userIdentifier !== (null) ;  tests["firstName is not null"] = data.chatMessage.postedUser.firstName !== (null) ;  tests["lastName is not null"] = data.chatMessage.postedUser.lastName!== (null) ;  tests["email is not null"] = data.chatMessage.postedUser.email!== (null) ;  tests["body contains title"] = responseBody.has("title");  tests["title is not null"] = data.chatMessage.postedUser.title !== undefined ;  tests["body contains department"] = responseBody.has("department");  tests["department is not null"] = data.chatMessage.postedUser.department!== undefined ;  tests["body contains location"] = responseBody.has("location");  tests["location is not null"] = data.chatMessage.postedUser.location!== undefined ;  tests["userGuid is not null"] = data.chatMessage.postedUser.userGuid!== (null) ;  tests["socialuserGuid is not null"] = data.chatMessage.postedUser.socialuserGuid!== (null) ;  tests["body contains cdnURL"] = responseBody.has("cdnURL");  tests["cdnURL is not null"] = data.chatMessage.postedUser.cdnURL!== undefined ;  tests["updatedon is not null"] = data.chatMessage.postedUser.updatedon!== (null) ;  tests["status is not null"] = data.chatMessage.postedUser.status === "ACTIVE" ;  tests["msgGuid is not null"] = data.chatMessage.msgGuid!== (null) ;  tests["status is not null"] = data.chatMessage.status === "ACTIVE" ;  tests["postedOn is not null"] = data.chatMessage.postedOn!== (null) ;  tests["updatedOn is not null"] = data.chatMessage.updatedOn!== (null) ;  tests["body contains recipient"] = responseBody.has("recipient");  tests["recipient is not null"] = data.chatMessage.recipient!== undefined ;  tests["canDelete is not null"] = data.chatMessage.canDelete!== (null) ;  tests["canEdit is not null"] = data.chatMessage.canEdit!== (null) ;  tests["messageType is not null"] = data.chatMessage.messageType!== (null) ;  tests["systemGenereated is not null"] = data.chatMessage.systemGenerated!== (null) ;  tests["body contains attachmentDocIds"] = responseBody.has("attachmentDocIds");  tests["attachmentsDocIds is not null"] = data.chatMessage.attachmentDocIds!== undefined ;  tests["reactionSummary is not null"] = data.chatMessage.reactionSummary!== (null) ;  tests["userReactions is not null"] = data.chatMessage.userReactions!== (null) ;  tests["body contains drillbacks"] = responseBody.has("drillbacks");  tests["drillbacks is not null"] = data.chatMessage.drillbacks!== undefined ;  tests["body contains directMentionUserHeaderList"] = responseBody.has("directMentionUserHeaderList");  tests["directMentionUserHeaderList is not null"] = data.chatMessage.directMentionUserHeaderList!== undefined ;  tests["body contains additionalInfo"] = responseBody.has("additionalInfo");  tests["additionalInfo is not null"] = data.chatMessage.additionalInfo!== undefined ;  tests["body contains notification"] = responseBody.has("notification");  tests["notification is not null"] = data.chatMessage.notification!== undefined ;  tests["body contains location"] = responseBody.has("location");  tests["location is not null"] = data.chatMessage.location!== undefined ;      }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers, body and  post Method. 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK and post a message to a topic. 4. Mentioned tests in summary should pass with out any errors       {      "chatMessage": {          "msgId": 41093,          "msgText": "hi",          "postedBy": 209,          "postedUser": {              "id": 209,              "userType": "MINGLEUSER",              "userIdentifier": "1e8bf208-bbc5-42e5-a863-ab14d1c77371",              "firstName": "Sudhakara",              "lastName": "vintha",              "nickName": null,              "email": "Sudhakara.Vintha@infor.com",              "title": "SE",              "department": null,              "location": null,              "userGuid": "6109F27E-2F62-4346-9DC9-D000889BF967",              "socialUserGuid": "E2E6B62C-BDFE-45CF-8CA0-F2A5E3B4141D",              "cdnURL": null,              "updatedon": 1509565923440,              "status": "ACTIVE"          },          "msgGuid": "8468ef09-c56d-4580-82c4-8f5ade49370f",          "status": "ACTIVE",          "postedOn": 1513823922795,          "updatedOn": 1513823922795,          "recipient": null,          "canDelete": true,          "canEdit": true,          "messageType": "CHAT",          "systemGenerated": false,          "attachmentDocIds": null,          "reactionSummary": [],          "userReactions": [],          "drillbacks": null,          "directMentionUserHeaderList": null,          "additionalInfo": null,          "notification": null,          "location": null      }  } |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21761: GetTopicMessages [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the endpoint to test the GetTopicMessages  {{gatewayURL}}/api/v1/topic/{{topicguid}}/messages    Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.    Method : Get    try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["body contains id"] = responseBody.has("msgId");  tests["msgId not null"] = data.messages[0].msgid !== (null) ;  tests["msgText is not null"] = data.messages[0].msgText !== (null) ;  console.log("msgText" + (data.messages[0].msgText));  tests["postedBy is not null"] = data.messages[0].postedBy !== (null) ;  tests["id is not null"] = data.messages[0].postedUser.id !== (null) ;  tests["userType is not null"] = data.messages[0].postedUser.userType !== (null) ;  tests["userIdentifier is not null"] = data.messages[0].postedUser.userIdentifier !== (null) ;  tests["firstName is not null"] = data.messages[0].postedUser.firstName !== (null) ;  tests["lastName is not null"] = data.messages[0].postedUser.lastName!== (null) ;  tests["body contains nickName"] = responseBody.has("nickName");  tests["nickName is not null"] = data.messages[0].postedUser.nickName!== undefined ;  tests["email is not null"] = data.messages[0].postedUser.email!== (null) ;  tests["body contains title"] = responseBody.has("title");  tests["title is not null"] = data.messages[0].postedUser.title !== undefined ;  tests["body contains department"] = responseBody.has("department");  tests["department is not null"] = data.messages[0].postedUser.department!== undefined ;  tests["body contains location"] = responseBody.has("location");  tests["location is not null"] = data.messages[0].postedUser.location!== undefined ;  tests["userGuid is not null"] = data.messages[0].postedUser.userGuid!== (null) ;  tests["socialuserGuid is not null"] = data.messages[0].postedUser.socialuserGuid!== (null) ;  tests["body contains cdnURL"] = responseBody.has("cdnURL");  tests["cdnURL is not null"] = data.messages[0].postedUser.cdnURL!== undefined ;  tests["updatedon is not null"] = data.messages[0].postedUser.updatedon!== (null) ;  tests["status is not null"] = data.messages[0].postedUser.status === "ACTIVE" ;  tests["msgGuid is not null"] = data.messages[0].msgGuid!== (null) ;  tests["status is not null"] = data.messages[0].status === "ACTIVE" ;  tests["postedOn is not null"] = data.messages[0].postedOn!== (null) ;  tests["updatedOn is not null"] = data.messages[0].updatedOn!== (null) ;  tests["body contains recipient"] = responseBody.has("recipient");  tests["recipient is not null"] = data.messages[0].recipient!== undefined ;  tests["canDelete is not null"] = data.messages[0].canDelete!== (null) ;  tests["canEdit is not null"] = data.messages[0].canEdit!== (null) ;  tests["messageType is not null"] = data.messages[0].messageType!== (null) ;  tests["systemGenereated is not null"] = data.messages[0].systemGenerated!== (null) ;  tests["body contains attachmentDocIds"] = responseBody.has("attachmentDocIds");  tests["attachmentsDocIds is not null"] = data.messages[0].attachmentDocIds!== undefined ;  tests["reactionSummary is not null"] = data.messages[0].reactionSummary!== (null) ;  tests["userReactions is not null"] = data.messages[0].userReactions!== (null) ;  tests["body contains drillbacks"] = responseBody.has("drillbacks");  tests["drillbacks is not null"] = data.messages[0].drillbacks!== undefined ;  tests["body contains directMentionUserHeaderList"] = responseBody.has("directMentionUserHeaderList");  tests["directMentionUserHeaderList is not null"] = data.messages[0].directMentionUserHeaderList!== undefined ;  tests["body contains additionalInfo"] = responseBody.has("additionalInfo");  tests["additionalInfo is not null"] = data.messages[0].additionalInfo!== undefined ;  tests["body contains notification"] = responseBody.has("notification");  tests["notification is not null"] = data.messages[0].notification!== undefined ;  tests["body contains location"] = responseBody.has("location");  tests["location is not null"] = data.messages[0].location!== undefined ;  }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers and Method (Get). 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK and display all topic messages 4. Mentioned tests in summary should pass with out any errors       {      "messages": [          {              "msgId": 41093,              "msgText": "hi",              "postedBy": 209,              "postedUser": {                  "id": 209,                  "userType": "MINGLEUSER",                  "userIdentifier": "1e8bf208-bbc5-42e5-a863-ab14d1c77371",                  "firstName": "Sudhakara",                  "lastName": "vintha",                  "nickName": null,                  "email": "Sudhakara.Vintha@infor.com",                  "title": "SE",                  "department": null,                  "location": null,                  "userGuid": "6109F27E-2F62-4346-9DC9-D000889BF967",                  "socialUserGuid": "E2E6B62C-BDFE-45CF-8CA0-F2A5E3B4141D",                  "cdnURL": null,                  "updatedon": 1509565923440,                  "status": "ACTIVE"              },              "msgGuid": "8468EF09-C56D-4580-82C4-8F5ADE49370F",              "status": "ACTIVE",              "postedOn": 1513823922797,              "updatedOn": 1513823922797,              "recipient": null,              "canDelete": true,              "canEdit": true,              "messageType": "CHAT",              "systemGenerated": false,              "attachmentDocIds": null,              "reactionSummary": [],              "userReactions": [],              "drillbacks": null,              "directMentionUserHeaderList": null,              "additionalInfo": null,              "notification": null,              "location": null          }      ],      "moreMessages": false,      "morePrevMessages": false  } |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21762: PostTeammateMessage [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the Endpoint to Test the PostTeammateMessage    {{gatewayURL}}/api/v1/user/{{GroupMember1}}/messages    Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.    Method : Post    Request Body    {"message":"hello","documentIds":[],"base64Audio":"","taggedUserGuids":[]}    Tests :  try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["body contains msgId"] = responseBody.has("msgId");  tests["msgId is not null"] = data.chatMessage.msgId !== (null) ;  tests["msgText is not null"] = data.chatMessage.msgText !== (null) ;  console.log("msgText" + (data.chatMessage.msgText));  tests["postedBy is not null"] = data.chatMessage.postedBy !== (null) ;  tests["id is not null"] = data.chatMessage.postedUser.id !== (null) ;  tests["userType is not null"] = data.chatMessage.postedUser.userType !== (null) ;  tests["userIdentifier is not null"] = data.chatMessage.postedUser.userIdentifier !== (null) ;  tests["firstName is not null"] = data.chatMessage.postedUser.firstName !== (null) ;  tests["lastName is not null"] = data.chatMessage.postedUser.lastName!== (null) ;  tests["body contains nickName"] = responseBody.has("nickName");  tests["nickName is not null"] = data.chatMessage.postedUser.nickName!== undefined ;  tests["body contains email"] = responseBody.has("email");  tests["email is not null"] = data.chatMessage.postedUser.email!== undefined ;  tests["title is not null"] = data.chatMessage.postedUser.title !== null ;  tests["body contains deparment"] = responseBody.has("department");  tests["department is not null"] = data.chatMessage.postedUser.department!== undefined ;  tests["body contains location"] = responseBody.has("location");  tests["location is not null"] = data.chatMessage.postedUser.location!== undefined ;  tests["userGuid is not null"] = data.chatMessage.postedUser.userGuid!== (null) ;  tests["socialuserGuid is not null"] = data.chatMessage.postedUser.socialuserGuid!== (null) ;  tests["body contains cdnURL"] = responseBody.has("cdnURL");  tests["cdnURL is not null"] = data.chatMessage.postedUser.cdnURL!== undefined ;  tests["updatedon is not null"] = data.chatMessage.postedUser.updatedon!== (null) ;  tests["status is not null"] = data.chatMessage.postedUser.status === "ACTIVE" ;  tests["msgGuid is not null"] = data.chatMessage.msgGuid!== (null) ;  tests["status is not null"] = data.chatMessage.status === "ACTIVE" ;  tests["postedOn is not null"] = data.chatMessage.postedOn!== (null) ;  tests["updatedOn is not null"] = data.chatMessage.updatedOn!== (null) ;  tests["recipient is not null"] = data.chatMessage.recipient!== (null) ;  tests["canDelete is not null"] = data.chatMessage.canDelete!== (null) ;  tests["canEdit is not null"] = data.chatMessage.canEdit!== (null) ;  tests["messageType is not null"] = data.chatMessage.messageType!== (null) ;  tests["systemGenereated is not null"] = data.chatMessage.systemGenerated!== (null) ;  tests["body contains attachmentDocIds"] = responseBody.has("attachmentDocIds");  tests["attachmentsDocIds is not null"] = data.chatMessage.attachmentDocIds!== undefined ;  tests["reactionSummary is not null"] = data.chatMessage.reactionSummary!== (null) ;  tests["userReactions is not null"] = data.chatMessage.userReactions!== (null) ;  tests["body contains drillbacks"] = responseBody.has("drillbacks");  tests["drillbacks is not null"] = data.chatMessage.drillbacks!== undefined ;  tests["body contains directMentionUserHeaderList"] = responseBody.has("directMentionUserHeaderList");  tests["directMentionUserHeaderList is not null"] = data.chatMessage.directMentionUserHeaderList!== undefined ;  tests["body contains additionalInfo"] = responseBody.has("additionalInfo");  tests["additionalInfo is not null"] = data.chatMessage.additionalInfo!== undefined ;  tests["body contains notification"] = responseBody.has("notification");  tests["notification is not null"] = data.chatMessage.notification!== undefined ;  tests["body contains location"] = responseBody.has("location");  tests["location is not null"] = data.chatMessage.location!== undefined ;    }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers, body and Post Method. 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK and post a message to Teammate. 4. Mentioned tests in summary should pass with out any errors       {      "chatMessage": {          "msgId": 41094,          "msgText": "hello 1513824002",          "postedBy": 209,          "postedUser": {              "id": 209,              "userType": "MINGLEUSER",              "userIdentifier": "1e8bf208-bbc5-42e5-a863-ab14d1c77371",              "firstName": "Sudhakara",              "lastName": "vintha",              "nickName": null,              "email": "Sudhakara.Vintha@infor.com",              "title": "SE",              "department": null,              "location": null,              "userGuid": "6109F27E-2F62-4346-9DC9-D000889BF967",              "socialUserGuid": "E2E6B62C-BDFE-45CF-8CA0-F2A5E3B4141D",              "cdnURL": null,              "updatedon": 1509565923440,              "status": "ACTIVE"          },          "msgGuid": "94c84c7a-c923-428b-8f44-78cbe1074139",          "status": "ACTIVE",          "postedOn": 1513824002507,          "updatedOn": 1513824002507,          "recipient": "015155f6-70d5-49dc-93c3-dded10e9d01b",          "canDelete": true,          "canEdit": true,          "messageType": "CHAT",          "systemGenerated": false,          "attachmentDocIds": null,          "reactionSummary": [],          "userReactions": [],          "drillbacks": null,          "directMentionUserHeaderList": null,          "additionalInfo": null,          "notification": null,          "location": null      }  } |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21763: GetTeammateMessages [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the endpoint to test the Get Teammate Messages  {{gatewayURL}}/api/v1/user/{{GroupMember1}}/messages      Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.    Method : Get    Tests :    try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["body contains idd"] = responseBody.has("id");  tests["name is not null"] = data.name !== (null) ;  tests["description is not null"] = data.description !== (null) ;  tests["groupType is not null"] = data.groupType !== (null) ;  tests["visibility is not null"] = data.visibility !== (null) ;  tests["ifsDistributionGroup is not null"] = data.ifsDistributionGroup !== (null) ;  tests["groupguid is not null"] = data.groupguid !== (null) ;  tests["status is not null"] = data.status !== (null) ;  console.log("Deleted Group status  " + (data.status));  tests["updatedOn is not null"] = data.updatedOn !== (null) ;  tests["createdOn is not null"] = data.createdOn !== (null) ;  tests["profilePhoto is not null"] = data.profilePhoto !== (null) ;  tests["body contains cdnURL"] = responseBody.has("cdnURL");  tests["cdnURL is not null"] = data.cdnURL !== undefined ;  tests["groupCreator is not null"] = data.groupCreator !== (null) ;  tests["createdby is not null"] = data.createdby !== (null) ;      }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers, and Method (Get). 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK and display all the teammate messages. 4. Mentioned tests in summary should pass with out any errors       {      "messages": [            {              "msgId": 41094,              "msgText": "hello 1513824002",              "postedBy": 209,              "postedUser": {                  "id": 209,                  "userType": "MINGLEUSER",                  "userIdentifier": "1e8bf208-bbc5-42e5-a863-ab14d1c77371",                  "firstName": "Sudhakara",                  "lastName": "vintha",                  "nickName": null,                  "email": "Sudhakara.Vintha@infor.com",                  "title": "SE",                  "department": null,                  "location": null,                  "userGuid": "6109F27E-2F62-4346-9DC9-D000889BF967",                  "socialUserGuid": "E2E6B62C-BDFE-45CF-8CA0-F2A5E3B4141D",                  "cdnURL": null,                  "updatedon": 1509565923440,                  "status": "ACTIVE"              },              "msgGuid": "94C84C7A-C923-428B-8F44-78CBE1074139",              "status": "ACTIVE",              "postedOn": 1513824002507,              "updatedOn": 1513824002507,              "recipient": null,              "canDelete": true,              "canEdit": true,              "messageType": "CHAT",              "systemGenerated": false,              "attachmentDocIds": null,              "reactionSummary": [],              "userReactions": [],              "drillbacks": null,              "directMentionUserHeaderList": null,              "additionalInfo": null,              "notification": null,              "location": null          }      ],      "moreMessages": true,      "morePrevMessages": false  } |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case MingleQA-21877: Delete Group [Version : 1]** | | | | | |
| Author: | vkotari | | | | |
| Summary:  Verify the endpoint to delete a group    {{gatewayURL}}/api/v1/admin/groups/{{groupguid}}/delete      Note : gatewayUrl depend upon the environment used for execution, it will provide through data file.      try  {      if(responseCode.code === 200)  {  var data = JSON.parse(responseBody);  tests["body contains idd"] = responseBody.has("id");  tests["name is not null"] = data.name !== (null) ;  tests["description is not null"] = data.description !== (null) ;  tests["groupType is not null"] = data.groupType !== (null) ;  tests["visibility is not null"] = data.visibility !== (null) ;  tests["ifsDistributionGroup is not null"] = data.ifsDistributionGroup !== (null) ;  tests["groupguid is not null"] = data.groupguid !== (null) ;  tests["status is not null"] = data.status !== (null) ;  console.log("Deleted Group status  " + (data.status));  tests["updatedOn is not null"] = data.updatedOn !== (null) ;  tests["createdOn is not null"] = data.createdOn !== (null) ;  tests["profilePhoto is not null"] = data.profilePhoto !== (null) ;  tests["body contains cdnURL"] = responseBody.has("cdnURL");  tests["cdnURL is not null"] = data.cdnURL !== undefined ;  tests["groupCreator is not null"] = data.groupCreator !== (null) ;  tests["createdby is not null"] = data.createdby !== (null) ;      }  }  catch(e)  {      console.log("failed to execute get user groups" + e);  } | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1. Open the Postman tool. 2. Enter the Endpoint URL with mentioned Headers, body and Method. 3. Click on Send button. 4. Verify all tests mentioned in summary | 1. User should be able to open postman tool. 2. All the mentioned data should be entered properly. 3. Should display the response body  with 200 OK . This endpoint should delete the group in chat 4. Mentioned tests in summary should pass with out any errors |  |  |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Importance: | Medium | | | | |
|  | | | | | |
| Requirement Jira: | SCRMCHAT-341 | | | | |
| Requirements | 117.0: Chat Microservices | | | | |
| Keywords: | None | | | | |
| Last Result | **Not Run** | | | | |