Test Specification

Test Project: WSO2 API Manager

Printed by TestLink on 11/06/2015

Table Of Contents

1.1.2.1.7.Specify Inline Response

APIM-52: Create an API from the scratch and give necessary details for design and implementation and save

APIM-281: Prototype API's should be visible on the API store

1.1.3.API manage stage

APIM-70: When user select subscription available to specific tenants visibility should always set to public

APIM-53: Add throttling tires and other details to a prototyped API and save

APIM-55: Save an API in manage stage.

1.1.3.1.Transport

APIM-56: Create an API with a WSDL , enable only the http transport and save

APIM-57: Create an API with a WSDL , enable only the https transport and save

APIM-58: Create an API with a WSDL , enable both https and http transports and save

APIM-59: Create an API from the scratch and complete the API until you save and publish it in manage page

APIM-69: When user select subscription available to all tenants visibility should always set to public

APIM-81: When user selects available to current tenants only, visibility can be selected as users choice

APIM-142: When having many copies of the API publish one of the copies given require re subscription option

1.2.Edit API

APIM-217: Check whether user can successfully edit icon.

1.3.Delete API

1.4.API Versioning/Copy API

1.4.1.default version

1.4.1.1.With default version

APIM-421: Copy an api with {version} strategy in context as the default api and invoke it sucesfully with its new endpoint

APIM-366: Check whether user can publish new copy given deprecate old versions option.

APIM-370: Check whether user can publish new copy given require re-subscription .

APIM-371: Check whether user can publish new copy with both deprecate old version and require re-subscription

1.4.1.2.without default version

APIM-363: Check whether user can set the new copy as the default version.

APIM-374: Check whether user can publish new copy with both deprecate old version and require re-subscription

APIM-373: Check whether user can publish new copy given require re-subscription .

APIM-372: Check whether user can publish new copy given deprecate old versions option.

APIM-420: Copy an api with {version} strategy in context and invoke it sucesfully with its endpoint

1.4.2.lifecycle

APIM-127: Check whether system allows to get a copy of an api which is in blocked state.

APIM-122: Check whether system allows to get a copy of an api which is in created state.

APIM-123: Check whether system allows to get a copy of an api which is in published state.

APIM-124: Check whether system allows to get a copy of an api which is in prototyped state.

APIM-125: Check whether system allows to get a copy of an api which is in deprecated state.

APIM-126: Check whether system allows to get a copy of an api which is in retired state.

1.4.3.Docs

1.4.3.1.Existing docs

1.4.3.1.1.Edit existing doc content

APIM-419: Check whether user can edit and change content type of an existing hot to type document.

APIM-418: Check whether user can edit existing 'other' type docs which has URL content type.

APIM-417: Check whether user can edit existing 'Support forum' type docs which has URL content type.

APIM-416: Check whether user can edit existing 'Public forum' type docs which has URL content type.

APIM-415: Check whether user can edit existing 'Samples & SDK' type docs which has URL content type.

APIM-413: Check whether user can edit existing 'other' type docs which has in-line content type.

APIM-412: Check whether user can edit existing 'samples & SDK' type docs which has in-line content type.

APIM-411: Check whether user can edit existing how to type docs which has in-line content type.

APIM-414: Check whether user can edit existing 'how to' type docs which has URL content type.

1.4.3.2.New docs

1.4.4.Other

APIM-128: Check whether system validate for existing versions of the same api

APIM-131: Check whether cancel button on add new version section is working.

1.5.API lifecycle management

APIM-161: Check whether auto subscription works for new versions of Blocked.

APIM-159: Check whether "DisplayMultipleVersions" functionality is working properly

APIM-158: Publish an API copy after editing the throttling tier and without re subscription option.

APIM-135: check whether user can change api lifecycle status of a published API to created.

APIM-141: Publish a copy of an API with a new version, giving require re subscription & deprecate old version.

APIM-140: Publish a copy of an API with a new version without giving require re subscription option.

APIM-139: Publish a copy of an API with a new version given require re subscription option.

APIM-138: check behaviour when lifecycle is changed to created from published when there are subscriptions.

APIM-137: Check whether API behave right when status is changed to blocked to published.

APIM-136: Check whether life API behave right when lifecycle changes to publish>depre;>blocked>depre;>retired

APIM-134: check whether user can change API life cycle, create > blocked

APIM-133: check whether user can change API life cycle, create > deprecated

APIM-132: check whether user can change API life cycle, create > retired

APIM-84: copy an published API and create a new version and publish it without deprecating the old version

APIM-85: copy and published API given a new version and publish it given deprecate old versions option.

APIM-86: Publish newest version of an api which has many versions given deprecate old versions option

APIM-87: create two versions in created state; publish the new one given deprecate old then publish old one

APIM-89: check whether user can change API life cycle, create > prototyped

APIM-160: Check whether auto subscription works for new versions of deprecated.

APIM-349: User should not be create to add a document without document name

1.6.Scopes

APIM-175: Check whether user can add different scopes for different resources.

APIM-172: Check whether user gets a validation message when trying to add scopes given invalid roles.

APIM-170: User should get validation message for special characters.

APIM-168: Check whether compulsory fields are marked with a red star.

APIM-166: User should be able to add scopes in the manage stage when creating an API

APIM-167: Check whether user can add scope to a created API using edit view.

APIM-169: Check whether user gets a validation message for empty compulsory fields.

APIM-171: Check whether user gets an validation message when trying to add scopes with duplicate key.

APIM-173: Check whether user can exit from the define scope window by clicking on the 'x' sign on the window

APIM-176: Check whether scopes get loaded successfully into the scope selection drop down in front of resource

APIM-174: Check whether user can exit from the define scope window by clicking

1.6.1.edit

APIM-178: Check whether user can edit and change scope already assigned to the resource into

APIM-177: Check whether user can edit and change scope already assigned to the resource into a different scope

1.7.API URI templates

APIM-73: Ability to define the templates in publisher

APIM-74: user should be able to define complex uri - templates having many parameters

APIM-75: User should be able to define complex uri-templates with each HTTP verb

APIM-76: Test boundary values in template definition

APIM-77: Test with requests having large number of query parameters

APIM-78: Test with lengthy URL parameters

APIM-79: Update URI template given a valid URI pattern-URI should be able to invoke the resource successfully

APIM-80: Update the template in synapse config in file system, change should reflect on API resource details

1.8.Search functionality

1.8.1.publisher

1.8.2.Store

1.9.API Docs

1.9.1.Add/Verify Documents

APIM-116: Check whether table which contains added documents get updated properly

APIM-105: Create a document given 'other' as the type and 'file' as the source.

APIM-104: Create a document given 'other' as the type and 'URL' as the source

APIM-103: Create a document given 'other' as the type and in-line as the source.

APIM-102: Create a document given 'support forum' as the type and 'file' as the source.

APIM-101: Create a document given 'support forums' as the type and 'URL' as the source

APIM-100: System should not allow to select 'in-line' source type given 'support forum' as the type

APIM-99: Create a document given 'public forum' as the type and 'file' as the source.

APIM-98: Create a document given 'public forums' as the type and 'URL' as the source

APIM-96: Create a document given 'Samples and SDK' as the type and 'file' as the source.

APIM-95: Create a document given 'Samples & SDK' as the type and 'URL' as the source

APIM-90: Check whether system validates for empty fields in add new document page

APIM-91: Successfully add new in-line, 'how to' type document

APIM-92: Create a document given 'How to' as the type and 'URL' as the source

APIM-93: Create a document given 'How To' as the type and 'file' as the source.

APIM-94: Create a document given 'samples & sdk' as the type and in-line as the source.

APIM-97: System should not allow to select 'in-line' source type given 'public forum' as the type

APIM-347: User should not be able to Create a document with the same name that already added

APIM-348: User added a document with summary

1.9.2.Update documents

APIM-165: Check whether system save changes when user adds special characters to the content of inline docs

APIM-113: Check whether system doesn't allow user to change source as 'in-line' for forum type documents.

APIM-112: Check whether system allows user to edit content of already created in-line documents.

APIM-114: Check whether user can update documents created already.

APIM-115: User should get validation message when trying to update document having an empty source URL

1.9.3.Delete Documents

APIM-121: Check whether documents of type other get deleted successfully.

APIM-120: Check whether documents of type 'support forum' get deleted successfully.

APIM-119: Check whether documents of type 'public forum' get deleted successfully.

APIM-118: Check whether documents of type 'sample SDK' gets deleted successfully.

APIM-117: Check whether documents of type how to get deleted successfully.

1.10.Sequences

1.11.Resource Level Authentication Types

APIM-422: A user should be able to set different types of resource level authentication types for the resource

2.API Store

2.1.Application management

APIM-196: Check where system redirects logged in user to my applications page successfully by clicking on my applications menu items.

APIM-195: Check whether user cannot access application page without signing into the API manager store.

2.1.1.Delete Application

APIM-203: Check whether application gets deleted properly and not allowing users to use application keys any longer.

APIM-202: Check whether user can delete created applications successfully

APIM-204: Check whether application consumer key and consumer secret is getting deleted once user delete appli

2.1.2.Add Application

APIM-201: Check whether user can see the usage details by clicking on the '?' sing of the callback URL.

APIM-198: Check whether compulsory fields of the add application form are marked with an asterisk.

APIM-197: Check whether system allows user to add a new application successfully.

APIM-199: Check whether compulsory fields of the 'add new application' form is being validated.

APIM-200: Check where name field of the 'add new application' form get validated for special characters.

2.1.3.Edit Application

APIM-206: Check whether the cancel button of the edit view works correctly.

APIM-205: Check whether user can edit created Applications.

2.2.Subscription management

2.3.Forums

2.4.Token generation

2.4.1.application token

APIM-207: Check whether user can successfully generate application token using api store UI.

APIM-208: Check whether user can use generated application token to invoke API.

APIM-209: Check whether application tokens get successfully expired once validity time is exceeded.

APIM-229: Check whether Application token revocation is successful when user regenerates the access token before expiry time.

APIM-230: Check whether sandbox and production access tokens are generated once user gives different validity times.

APIM-231: Check whether sandbox and production tokens get expired correctly when they are given 2 different validity periods.

APIM-242: Check whether User interface reflect changes done to validity time in identity.xml

APIM-243: Check whether changes made for ApplicationAccessTokenDefaultValidityPeriod in identity.xml applies correctly.

2.4.2.password grant type

APIM-241: Check whether validity period configure using identity.xml applies correctly.

APIM-221: whether token api returns an error when invoking it with invalid user credentials.

APIM-210: Check whether token api can be used to generate user tokens of type password grant successfully without specifying a scope.

APIM-211: Check whether user token get expired when validity period exceeds.

APIM-220: Check whether token api returns an error "invalid_grant" when it is invoke with a registered user who doesn't have subscriber permission.

APIM-222: Check whether token api returns an error when user trying to invoke with an encoded string of a deleted application.

2.4.3.Authorization code

APIM-237: Check whether system removes first token when same user generates access token twise using same application's consumer key and secret.

APIM-234: Check whether access token generated can no longer use once access permission of the application is denied for the user whose credentials were used to generate authorisation code

APIM-224: Check whether Token API returns an error when it is invoked given an valid authorisation code generated for app2 with consumer key and secret of app1.

APIM-213: Check whether access token can be generated successfully using authorisation grant type.

APIM-214: Check whether access token generated using authorisation grant type can be used to invoke API successfully.

APIM-235: Check whether default validity period configure in identity.xml applies correctly when generating authorisation code.

APIM-236: Check whether system gives an error when trying generate access token using expired authorisation code.

2.4.4.SAML extension grant type

APIM-228: Check whether user gets an error when invoking token api with a SAML2 assertion generated for a user who doesn't have subscribe permission.

APIM-226: Check whether user gets an error message once token API is invoked with a invalid consumer key:consumer secret pair.

APIM-218: Check whether system successfully return OAuth2.0 access token when a valid SAML2 bearer token is being passed.

APIM-225: Check whether user gets an error message once token API is invoked with a invalid "SAML2_Encoded_Assertion_Token".

APIM-238: Check whether validity time configure using identity.xml get applied to the token generated using SAML grant type correctly.

APIM-239: Check whether access token generated using saml2 bearer token grant type expires when exceeding the given validity time.

2.4.5.renewing access token

APIM-212: Check whether user can successfully renew token using refresh token once the user token is expired.

APIM-223: Check whether API store > my subscription page can be use to regenerate access token.

2.4.6.revoking access token

APIM-219: Check whether user can successfully make the generated token invalid by invoking revoke end point.

2.4.7.Implicit grant type

APIM-216: Check whether access token generated using implicit authorisation grant type can be used to invoke API successfully

APIM-215: Check whether access token can be generated successfully using Implicit grant type.

2.4.8.Other

APIM-244: Check whether validity period configured using identity.xml applies correctly when regenerating new

2.4.9.Generate application token for scopes

2.4.9.1.Token generation for scopes

APIM-407: A user should be able to generate a token via CURL for a paticular scope of an API when there are multiple scopes assigned to an API via CURL

APIM-405: A user should be able to assign a scope to a newly creating API and subscribe by an application and generate a access token to the paticular scope

APIM-364: A user should be able to generate a token via CURL for a paticular scope of an API

APIM-362: A user should be able to assign a scope to a subscribed application which the key is already generated and generate the token again for a scope

APIM-361: A user should be able to assign multiple scopes to a subscribed application and generate token

APIM-360: A user should be able to assign a scope to a subscribed application and generate a new access token to a paticular scope

2.5.API visibility and subscriptions

APIM-68: API with subscription available only to current tenant, not allow other tenant users to subscribe.

APIM-66: Publish an API given 'visible to my domain' -users belong to same tenant should be able to view.

APIM-60: Published APIs with public visibility can be viewed by all users.

APIM-61: publish an API given 'visible to my domain' -users belong to other tenantsshould not be able to view

APIM-62: published api with visibility restricted to a role should visible to users with the particular role.

APIM-63: publishes APIs with visibility restricted to a role should be visible to user with publisher role.

APIM-64: API with visibility restricted to a role should not be visible to users belong to other tenants

APIM-65: publish an api with visibility restricted to multiple roles-only users with allowed roles can view.

APIM-67: API with subscription available only to current tenant, allows current tenant user to subscribe.

APIM-71: APIs with subscription available to all tenants should allow users in other tenants to subscribe.

APIM-72: API with subscription available to specific tenants should allow subscription to specified tenants

2.6.Application token

2.7.Search functionality

2.7.1.publisher

APIM-350: User Search added API by name

APIM-351: User Search added API by provider name

APIM-352: User Search added API by api version

APIM-353: User should be able to search added API by context

APIM-354: User search for API with upper case name

APIM-355: User search for API with lower case name

APIM-356: Successfully logged in user should be able search API by status

APIM-357: Successfully logged in user should be able search API by description

APIM-358: Successfully logged in user should be able search API by subcontext

APIM-359: Successfully logged in user should be able search API by documentation content

APIM-367: Successfully logged in user should be able search with keywords that contains only part of the provider name

APIM-368: Successfully logged in user should be able search with keywords that contains only part of the conte

APIM-384: User should be able to see all API's in all stage on browse page

2.7.2.Store

APIM-404: (Tenant specific) User Search added API by api version

APIM-403: (Tenant specific) User Search added API by name

APIM-402: (Tenant specific) User should be able to search added API by context

APIM-401: (Tenant specific) User search for API with upper case name

APIM-400: (Tenant specific) User search for API with lower case name

APIM-399: (Tenant specific) Successfully logged in user should be able search API by description

APIM-398: (Tenant specific) Successfully logged in user should be able search with keywords that contains only part of the provider name

APIM-393: (Tenant specific)Successfully logged in user should be able search with keywords that contains only part of the context name

APIM-392: Successfully logged in user should be able search with keywords that contains only part of the context name

APIM-391: Successfully logged in user should be able search with keywords that contains only part of the provider name

APIM-390: Successfully logged in user should be able search API by description

APIM-389: User search for API with lower case name

APIM-388: User search for API with upper case name

APIM-387: User should be able to search added API by context

APIM-386: User Search added API by name

APIM-385: User Search added API by api version

2.8.API Docs

2.8.1.documentation visibility

APIM-111: Check document visibility when it's given as 'Visible to my domain'.

APIM-110: Check document visibility when it's given as private.

APIM-109: Check doc visibility when it's given same as api visibility; api visibility = restricted by role

APIM-108: Check doc visibility when it's given same as api visibility; api visibility = visible to my domain.

APIM-107: When user clicks on the '?' icon of the visibility field tip box should open.

APIM-106: Check document visibility when it's given same as api visibility when api visibility = public

2.9.API Console

2.9.1.Verify functionality without logging in to Store

2.9.1.1.Request Header

APIM-383: Invoke the API without specifying a value for the request header

2.9.2.Verify functionality while logged in to Store

2.10.Workflows

3.Server Profiles

3.1.api-publisher

APIM-1: Check whether the publisher app gets loaded properly

APIM-2: Check for other start up errors

APIM-3: check whether the GW and KM functionality are not working

APIM-4: Check whether the statistics are displayed (when configured with BAM)

3.2.api-store

APIM-5: Check whether the store functionality works

APIM-6: Check for other start up errors

APIM-7: Verify that GW and KM functionality are not working

APIM-8: Check whether the statistics are displayed (when configured with BAM)

APIM-9: Check the workflow integration

3.3.api-key-manager

APIM-10: Check whether the key manager functionality works

APIM-11: Check for startup errors

APIM-12: Verify that the Publisher, store, Gateway functionality are blocked

3.4.gateway-manager

APIM-13: Check whether the GW functionality work with the profile

APIM-14: Check for other start up errors

APIM-15: Verify that the key manager, publisher, store functionality are blocked

3.5.gateway-worker

APIM-16: Check whether Gateway worker functionality work with this profile

APIM-17: Verify that the key manager, publisher, store functionality are blocked

APIM-18: Check for other start up errors

APIM-19: Check whether the stat publishing to BAM works properly

4.APIM user roles

APIM-427: Try to subscribe by a user who has only login permission

5.Throttling

5.1.API Level Throttling

APIM-156: Check whether Unlimited Resource level throttling tier works.

5.2.Application Level Throttling

APIM-155: Check whether Bronze Resource level throttling tier works.

5.3.Resource Level Throttling

APIM-154: Check whether Silver Resource level throttling tier works.

5.4.IP Level Throttling

APIM-153: Check whether GOLD Resource level throttling tier works.

5.5.Other Throttling Scenarios

APIM-152: Check whether Unlimited Application level throttling tier works.

APIM-151: Check whether GOLD API level throttling tier works.

APIM-150: Check whether Bronze Application level throttling tier works.

APIM-149: Check whether Silver Application level throttling tier works.

APIM-148: Check whether Unlimited API level throttling tier works.

APIM-147: Check whether Bronze API level throttling tier works.

APIM-146: Check whether SILVER API level throttling tier works.

APIM-144: Check whether tier of api loads correctly in api store according to the tier configure in publisher.

APIM-145: Check whether Gold Application level throttling tier works.

APIM-157: Check whether throttling works correctly given different throttling tiers for different resources.

APIM-162: Check whether system behaves correctly when each resource have different throttling tiers.

APIM-164: Check the behaviou when user subscribe to same API to 2 applications with different throttling tiers

6.Caching

APIM-428: Verify JWT caching

7.API invocation

7.1.Resources - Auth types

APIM-194: Check whether changes applies when user change auth type of a resource 'application & application user' to 'application user'

APIM-193: Check whether changes applies when user change auth type of a resource 'application & application user' to application

APIM-192: Check whether changes applies when user change auth type of a resource 'application & application user' to none

APIM-191: Check whether changes applies when user change auth type of a resource 'application user' to application & application user'

APIM-190: Check whether changes applies when user change auth type of a resource 'application user' to 'application'

APIM-189: Check whether changes applies when user change auth type of a resource 'application' to 'none'

APIM-188: Check whether changes applies when user change auth type of a resource 'none' to 'application user'

APIM-187: Check whether changes applies when user change auth type of a resource 'application' to 'application & application user'

APIM-186: Check whether changes applies when user change auth type of a resource 'none' to 'application & application user

APIM-185: Check whether changes applies when user change auth type of a resource 'application' to 'application user'

APIM-182: Check whether resources with auth type non can be access with no access token.

APIM-181: Check wether resources given auth type as 'application user' can be invoked using user access tokens.

APIM-180: Check whether resources secured using 'application' auth type cannot be invoked using user access tokens.

APIM-179: Check whether resources secured using 'application' auth type can be accessed using application token generated for the application.

APIM-183: Check whether each resource show correct behaviour according to the auth type assigned to the resource.

APIM-184: Check whether changes applies when user change auth type of a resource 'none' to 'application'.

7.2.lifecycle

7.3.scopes

APIM-406: A user should not be able to invoke an API method which is not bound to a paticular scope that is assigned to a different role once the inovkable scope is changed to a scope which is mapped to the different role

APIM-365: A user should be able to generate a token via CURL for a paticular scope of an API when there are multiple scopes defined for an API and invoke

8.Statistics

8.1.API Store Statistics

8.1.1.Configure with BAM (1.8.0 >)

8.1.1.1.permission based

APIM-246: Check whether users logged into publisher who only has creator role can view both All Statistics and 'My APIs' > Statistics tabs on the side menu.

APIM-245: Check whether users logged into publisher who only has publisher role cannot view "My APIs > statis

8.1.1.2.Subscription statistics

APIM-248: Check whether system depicts correct subscription statistics to user who has creator permission when system has APIs created two different creators.

APIM-247: Check whether system depicts correct subscription statistics to user who has creator permission when system only has APIs created by himself.

APIM-249: Check whether Statistics get update properly once subscriber unsubscribe from the API.

APIM-250: Check whether user can successfully view subscription statistics relevant to a given time duration.

8.1.1.3.Usage statistics

APIM-255: Check whether User can successfully view API usage statistics relevant to a particular time duration.

APIM-253: Check whether API manager statistics are updated accurately when another version of the same API is

APIM-252: Check whether API creator can successfully view API usage statistics of API created by him as well as Usage statistics available for all APIs.

APIM-254: Check whether Usage statistics gets updated accurately when a particular API is deleted from API man

8.1.1.4.response time

APIM-256: Check whether the API response time statistics are accurate.

8.1.1.5.Last access time

APIM-257: Check whether user can successfully view accurate last access times of each APIs.

APIM-272: Check whether user can view statistics related to a particular date range successfully.

8.1.1.6.API Usage by Resource Path

APIM-269: Check whether user can view API usage by resource paths took place during a particular date range

APIM-258: Check whether user can view API requests sent to different resource paths of APIs accurately.

8.1.1.7.Usage by destination

APIM-270: Check whether user can view API usage by destination relevant to a particular date range

APIM-267: Check whether user can view API usage by destination statistics successfully.

8.1.1.8.Usage by user

APIM-259: Check whether user can successfully view API usage of each API by user accurately.

APIM-271: Check whether user can view API usage by user related to a particular date range.

8.1.1.9.Faulty Invocation

APIM-260: Check whether user can successfully view accurate faulty invocation statistics.

APIM-273: Check whether user can view correct faulty API related statistics for a selected date range.

8.1.1.10.API store statistics

APIM-261: Check whether API usage statistics per application gets updated accurately.

APIM-262: Check whether Faulty invocation per application gets updated correctly .

APIM-263: Check whether user can successfully view api usage from resource path per application.

APIM-268: Check whether user can successfully view top users per applications.

8.1.2.Configure with with BAM from UI (1.9.0 <)

8.1.2.1.permission based

APIM-317: Check whether users logged into publisher who only has creator role can view both All Statistics and

APIM-318: Check whether users logged into publisher who only has publisher role cannot view "My APIs > statis

8.1.2.2.Subscription statistics

APIM-319: Check whether system depicts correct subscription statistics to user who has creator permission when

APIM-320: Check whether system depicts correct subscription statistics to user who has creator permission when

APIM-321: Check whether Statistics get update properly once subscriber unsubscribe from the API.

APIM-322: Check whether user can successfully view subscription statistics relevant to a given time duration.

8.1.2.3.Usage statistics

APIM-323: Check whether User can successfully view API usage statistics relevant to a particular time duration

APIM-324: Check whether API manager statistics are updated accurately when another version of the same API is

APIM-325: Check whether API creator can successfully view API usage statistics of API created by him as well a

APIM-326: Check whether Usage statistics gets updated accurately when a particular API is deleted from API man

8.1.2.4.response time

APIM-327: Check whether the API response time statistics are accurate.

8.1.2.5.Last access time

APIM-328: Check whether user can successfully view accurate last access times of each APIs.

APIM-329: Check whether user can view statistics related to a particular date range successfully.

8.1.2.6.API Usage by Resource Path

APIM-330: Check whether user can view API usage by resource paths took place during a particular date range

APIM-331: Check whether user can view API requests sent to different resource paths of APIs accurately.

8.1.2.7.Usage by destination

APIM-332: Check whether user can view API usage by destination relevant to a particular date range

APIM-333: Check whether user can view API usage by destination statistics successfully.

8.1.2.8.Usage by user

APIM-334: Check whether user can successfully view API usage of each API by user accurately.

APIM-335: Check whether user can view API usage by user related to a particular date range.

8.1.2.9.Faulty Invocation

APIM-338: Check whether user can successfully view accurate faulty invocation statistics.

APIM-339: Check whether user can view correct faulty API related statistics for a selected date range.

8.1.2.10.API store statistics

APIM-340: Check whether API usage statistics per application gets updated accurately.

APIM-341: Check whether Faulty invocation per application gets updated correctly .

APIM-342: Check whether user can successfully view api usage from resource path per application.

APIM-343: Check whether user can successfully view top users per applications.

8.1.2.11.Stats available for publisher

APIM-344: Check whether publisher can view API usage of the particular API by versions.

APIM-345: Check whether user can successfully view statistics of API subscriptions by versions.

APIM-346: Check whether user can successfully view statistics of API Usage by Current Subscribers.

8.1.2.12.Configure through admin dashboard

APIM-408: Successfully logged user should be configure to BAM through admin dashboard

APIM-409: User saved without adding URL group configure to BAM through admin dashboard

APIM-410: User should be able to view already configuration after logged into admin dashboard again

8.2.API Publisher Statitics

APIM-266: Check whether user can successfully view statistics of API Usage by Current Subscribers.

APIM-265: Check whether user can successfully view statistics of API subscriptions by versions.

APIM-264: Check whether publisher can view API usage of the particular API by versions.

8.3.Southbound Stat Publishing

9.API Gateway

9.1.CORS

9.1.1.CORS configuration enabled

9.1.1.1.Access-Control-Allow Parameter verification

9.1.1.1.1.Access-Control-Allow-Origin

APIM-369: API invocation with Access-Control-Allow-Origin=specific valid Store address

APIM-375: API invocation with Access-Control-Allow-Origin=an invalid Store address

APIM-382: API invocation when connected to an external API Store

9.1.1.1.2.Access-Control-Allow-Methods

APIM-377: API invocation with Access-Control-Allow-Methods=All methods

APIM-378: API invocation with Access-Control-Allow-Methods=Selected # of methods

9.1.1.1.3.Access-Control-Allow-Headers

APIM-376: API invocation with Access-Control-Allow-Headers=different types of headers

9.1.1.2.Verify CORS for different API Types

APIM-380: Verify CORS behaviour for versioned APIs

APIM-381: Verify CORS behaviour for prototyped APIs

9.1.2.CORS configuration disabled

APIM-379: Invoking APIs when CORS configuration is disabled

10.Application sharing capabilities

APIM-425: Test whether applications do not share between different tenant domain.

APIM-423: Test whether application sharing capabilities work accurately based on user's organisation.

APIM-424: Test whether users can subscribe to APIs using shared applications, generate keys and invoke API

11.External API Store

APIM-426: Publish an API created in to an external API store

Scope

WSO2 API Manager

1.Test Suite : API Publisher

Test Artifact
Mind Map

1.1.Test Suite : API creation

Test Artifact
Mind Map

Test Case APIM-275: Create an api with version in different places in the context publish it and invoke it [Version : 1]				
Author:	Ushani Balasooriya			
Summary: Sucessfully logged in user should be able to create and publish an api with a context which has {version} tag in every place and which will be replaced by the version number provided in the Version and invoke it				
Preconditions: 1. A user should be sucessfully logged in to the system 2. A user should have provided other relevant information to create an api				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Provide Context as api1/{version}/test and version as 1.0.0 and other information and save the api	API context should be created with the context as api1/1.0.0/test and the endpoint with https://<host>:<port>/api1/1.0.0/test		
2	Provide Context as {version}/api1 and version as 1.0.0 and other information and save the api	API context should be created as 1.0.0/api1 and endpoint as https://<host>:<port>/1.0.0/api1		
3	Provide Context as api1/test/{version} and Version as 1.0.0 and other information and save it	API context should be created as api1/test/1.0.0 and endpoint as https://<host>:<port>/api1/test/1.0.0		
4	Login to store, subscribe to the api, generate a token and then Inovoke the aboe created apis which are with 3 different contexts	It should be sucessfully invoked		
5	Provide Context as api1/{version}/test and version as 1.0.0 and other information and save the api	API context should be created with the context as api1/1.0.0/test and the endpoint with https://<host>:<port>/api1/1.0.0/test		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.1.1.Test Suite : API Design stage

Test Artifact
Mind Map

Test Case APIM-22: Editing an API [Version : 1]				
Author:	Nirodha Gallage			
Summary: Edit and save an already created API				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on a saved API and and try to change the http verbs (add/remove) for a resource and save it.	Should be saved without errors.		
2	Give a resource (say /customer) with GET and POST methods and save. Now again add another resource with the same name (/customer) with POST and DELETE methods. Error "Resource already exist for URL Pattern / customer and Verb POST" Add DELETE method only. Then the existing /customer resource should be updated with DELETE with out adding a new resource.	Should work		
3	Change visibility level and check the store whether it is reflected	should work		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			

Last Result	**Not Run**

Test Case APIM-21: Try to change the lifecycle of an API in design stge [Version : 1]				
Author:	Nirodha Gallage			
Last edit by:	Waruna Perera			
Summary:				
Click on the created API and go to lifecycles page and check wether it can be promoted to PROTOTYPED or PUBLISHED - it should not				
Preconditions:				
1) User should be log into the system. 2) User should have created API in design stage				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User selected the created API	API should be successfully selected		
2	User clicks lifecycle tab	Lifecycle stage should be successfully selected		
3	Pressed the dropdown list	Published should not be on the list		
Execution type:	Automated			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-20: Create API in design stage [Version : 1]				
Author:	Nirodha Gallage			
Summary:				
Create an API and Save it in the Design stage and browse the APIs and check the created one is visible. Check whether it is visible in API store, it should not.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Try to save with out API name	Not allowed		
2	try to save with out API context	Not Allowed		
3	Try to save with out API version	Not Allowed		
4	Add API with a thumbnail image	API should be visible with the thumbnail when saved		
5	Try to give a non-image file when selecting the thumbnail	Not Allowed		
6	Save the API with 15 different tags	Should be allowed		
7	Try to save without adding any resource	it should ask to create the default resource		
8	Change visibility level to Restricted to Role and try to give an non existing role and save	Should be validated		
9	Try to save the API with URL pattern and resource name, but with out specifying a verb	it should not be allowed		
10	Create API with a context containing multiple slashes (/products /orders/)	Should be allowed and should work		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-276: Change the visibilty levels of the API to public and save it [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to change the API visibility levels of the API to public and save it				
Preconditions:				
1) User should be successfully logged into the system 2) Should be a created API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User selectes the created API from the list	API should be successfully selected.		

2	User press edit button and select public (it should be the default one)	That API should be visible to all users		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-278: Change the visibilty levels of the API to 'reststircted by role' and save it [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to change the API visibility levels of the API to 'reststircted by role' and save it				
Preconditions:				
1) User should be successfully logged into the system 2) Should be a created API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User selectes the created API from the list	API should be successfully selected.		
2	User press edit button and select reststircted by role	'visible to role' text field shoud be appeared		
3	User gives roles to visible	That API's should be visible to the roles		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-277: Change the visibility levels of the API to 'visible to my domain' and save it (this feature is only be visible if there are multiple tenants) [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to change the API visibility levels of the API to 'visible to my domain' and save it. (this feature is only be visible if there are multiple tenants)				
Preconditions:				
1) User should be successfully logged into the system 2) Should be a created API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User selectes the created API from the list	API should be successfully selected.		
2	User press edit button and select visible to my domain	API should be only visible on that domain only.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-279: Change visibility level to 'Restricted to Role' and try to give an non existing role and save [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Sucessfully logged in user change visibility level to 'Restricted to Role' and try to give an non existing role and save				
Preconditions:				
1) User should be successfully logged into the system 2) Should be a created API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User selectes the created API from the list	API should be successfully selected.		
2	User press edit button and select 'restricted by role'	'visible to role' text field shoud be appeared		

3	User gives invalid role	Error message should be appeared message text - Invalid role nam		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-280: Try to save without adding any resource [Version : 1]

Author:	Shamin Goonetileke

Summary:

Successfully logged in user try to save without adding any resource - it should ask to create the default resource

Preconditions:

1) User should be successfully logged into the system.

2) User should be on API creation page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User gives valid API name	Name should be added.		
2	User gives valid context	Context should be added correctly.		
3	User gives a valid version	version should be added		
4	User clicks save	Message should be prompt Message text - 'At least one resource should be specified. Do you wa[illegible] a wildcard resource (/*).'		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.1.2.Test Suite : API implementation stage

Test Artifact
Mind Map

Test Case APIM-23: Create API from scratch and give necessary details and deploy as prototype [Version : 1]

Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-24: Open an saved API in design stage and give the necessary details and deploy it as prototype [Version : 1]

Author:	Nirodha Gallage			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Open an saved API in design stage, and go to 'Implement' page.			
2	Select 'Backend Endpoint' for Implementation method.			
3	Select Endpoint Type as HTTP Endpoint and give a backend URL. Eg: http://ws.cdyne.com/phoneverify/phoneverify.asmx			
4	Click on Deploy Prototype. And go back to API browse page by clicking cancel button.	When this step is done, when you go to store the API should be visible in Prototype AP[illegible] section.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-25: Prototype APIs should not be visible in general store [Version : 1]

Author:	Nirodha Gallage
Last edit by:	Shamin Goonetileke

Summary:

Successfully logged in user should not be visible prototype API's in general store

Preconditions:

1) User should be logged into the system

2) There should be a created prototype API .

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User logged into store with valid username and password	User should be successfully logged into the store		
2	User views API list	Created prototype API should not be available in the general store		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-26: Prototype APIs should not be visible in the recently added list [Version : 1]

Author:	Nirodha Gallage
Last edit by:	Shamin Goonetileke

Summary:

Successfully logged in user should not be able to view prototype API's recently added list.

Preconditions:

1) User should be logged into the system

2) There should be a created prototype API .

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User logged into store with valid username and password	User should be successfully logged into the store		
2	User views recently API list	Prototype API's should not be visible on that list		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-27: Tags of Prototype APIs should not be visible in the tags list [Version : 1]

Author:	Nirodha Gallage
Last edit by:	Shamin Goonetileke

Summary:

Successfully logged in user should not be able to view prototype API's tags in the tags list.

Preconditions:

1) User should be logged into the system

2) There should be a created prototype API .

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:

1	User logged into store with valid username and password	User should be successfully logged into the store		
2	User should be on API store home page	Product APIs tags should not be visible in the tags list on right corner		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.1.2.1.Test Suite : Endpoint Types

Test Artifact
Mind Map

1.1.2.1.1.Test Suite : HTTP Endpoint

This endpoint type is mostly used for defining RESTful endpoints

Test Artifact
Mind Map

Test Case APIM-28: Production/Sandbox endpoint URLs should be validated at UI for correct URL pattern [Version : 1]	
Author:	Nirodha Gallage
Summary:	This should be validated at UI level before saving the API.
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-29: Check the functionality of endpont "Test" button [Version : 1]				
Author:	Nirodha Gallage			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Give a valid endpoint and Click the test button	It should show as valid		
2	Give an invalid endpoint and click the test button	It should show as invalid		
3	Test the functionality of Test button same as above for Sandbox endpoint			
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-31: Adding a WADL of the endpoint [Version : 1]				
Author:	Nirodha Gallage			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Give a valid WADL endpoint and check the Test URI button	Should indicate as valid		
2	Give an invlaid WADL endpoint and check the Test URI button	Should indicate as invalid		
3	Add a valid WADL and save and publish the API. Go to store and check the WADL appears for the API.	WADL should be viewable		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				

Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-32: Apply Endpoint Security configuration to the API [Version : 1]				
Author:	Nirodha Gallage			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Select the endpoint as secured and try to save the API with username and password blank	should not be allowed		
2	Give valid username and password and save and publish the API. Endpoint should be security applied. Invoke the API in usual way.	API operation should be invoked		
3	Give invalid username or password for the secured configuration and save and publish the API. Try to invoke with the usual way.	It should not be allowed since the credentials are wrong		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-30: Adding a WSDL of the endpoint [Version : 1]				
Author:	Nirodha Gallage			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Give a valid WSDL url and check the Test URI button	Should be validated		
2	Give an invalid WSDL URL and check the Test URI button	should be indicated as not valid		
3	After giving a valid WSDL URI and save it and publish the API. And go to the store and chcek the WSDL Link of that API created. In store when you download the wsdl and open it all the enpoint urls need to be re written with gateway URLs.			
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-33: Give a valid Production endpoint URL and publish the API and invoke it [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-34: Give a valid sandbox endpoint URL and publish the API and invoke the API with sandbox token [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	

Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.1.2.1.2.Test Suite : WSDL Endpoint

Test Artifact
Mind Map

Test Case APIM-35: Select the WSDL Endpoint type and check the Test button for production and sandbox endpoints [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-36: Give the service and port details as specified in the wsdl and save the API [Version : 1]				
Author:	Nirodha Gallage			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	save the API after correct details	Should save properly		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-37: Give incorrect details for Service and Port fields, something different to what is specified in WSDL [Version : 1]				
Author:	Nirodha Gallage			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Save the API after giving incorrect details.	Should pop up and error message		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-38: Give the WADL URL for the API and check whether it is downloadable in API store [Version : 1]				
Author:	Nirodha Gallage			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Check with a valid WADL URL	Should be able to save the API and retrieve the WADL in Store properly		
2	Invalid WADL url	Should not be able to save		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.1.2.1.3.Test Suite : Address Endpoint

This endpoint type is used for providing SOAP endpoints

Test Artifact
Mind Map

Test Case APIM-39: Set valid SOAP service endpoints for Production and Sandbox and check the "Test" button [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-40: Production/Sandbox URLs should be validated for correct URL patterns at UI level before saving [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-41: Add a WSDL for the endpoint and save [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-42: Add a WADL for the endpoint and save [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-43: Apply Security to the Production endpoint [Version : 1]				
Author:	Nirodha Gallage			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Set the correct username and password and try to invoke the API with production token	API should be invoked		
2	Set the incorrect username or password and try to invoke the API	Should not be allowed		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				

Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.1.2.1.4.Test Suite : Failover Endpoints

Test Artifact
Mind Map

Test Case APIM-44: Add at least 5 valid production endpoints and save the API [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-45: Invoke the API multiple times and verify only the first endpoint is invoked all the times [Version : 1]	
Author:	Nirodha Gallage
Summary:	
At this point all the endpoints should be up and running and whent the API is invoked multiple times it should invoke only the first endpoint all the time.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-46: Shutdown the first endpoint and validate the subsequent invocations are routed to the second endpoin [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-47: Apply Endpoint Security and verify that security headers are passed to all the failover endpoints [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-48: Add a WSDL for the endpoint and save and check whether it is retrievable at the store [Version : 1]

Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-49: Add a WADL for the endpoint and save and check whether it is retrievable at store [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.1.2.1.5.Test Suite : Load Balanced Endpoint

Test Artifact
Mind Map

Test Case APIM-50: Add at least 5 Load balanced endpoints and save the API [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-51: Set the algorithm "Round Robin" and invoke the API and check the requests are load balanced [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.1.2.1.6.Test Suite : Advanced Endpoint configuration

Test Artifact
Mind Map

1.1.2.1.7.Test Suite : Specify Inline Response

Test Artifact
Mind Map

Test Case APIM-52: Create an API from the scratch and give necessary details for design and implementation and save [Version : 1]	
Author:	Sewmini Jayaweera

Summary:	
Create an API from the scratch and give necessary details for design and implementation and save - Given design and implementation details should be saved properly and when you go back to the page again the changes should be persisted.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-281: Prototype API's should be visible on the API store [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to view added prototype API's in the Protoyped API's section.				
Preconditions:				
1) User should be logged in to the system. 2) There should be a created prototype API				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User logged into store with valid username and password	User should be successfully logged into the store		
2	User clicks prototyped API section	Prototype API's should be available on the list		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.1.3.Test Suite : API manage stage

Test Artifact	
Mind Map	

Test Case APIM-70: When user select subscription available to specific tenants visibility should always set to public [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
User adds an api given visibility as 'visible to my domain' then in the manage page set subscription to 'available to specific tenants' when user selects subscription avalable to specific tenants visibility should automatically chage to public.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-53: Add throttling tires and other details to a prototyped API and save [Version : 1]				
Author:	Sewmini Jayaweera			
Last edit by:	Nirodha Gallage			
Summary:				
Click on an API in Prototyped state and Edit it and go to manage page and add the throttling tiers and other details etc. And click on "save". Now check on the store whether the saved API is visible in store. And check the API state in lifecycles tab, it still should be prototyped.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to 'Manage' page and tick 'Unlimited', 'Gold', 'Bronze' from 'Tier Availability' drop down.	They should be ticked and persisted.		
2	Tick both 'HTTPS' and 'HTTP' to select Transports.	Both Should be selected and persisted.		
3	Keep the other values as default and click on 'SAVE' button.	API should be saved and a confirmation message should be shown saying API is saved.		
Execution type:	Manual			

Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-55: Save an API in manage stage. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user with permission give configurations in manage stage and save api.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click save without giving compulsory fields	System should give validation messages for 'tier availability' and transport fields.		
2	Add the throttling tiers and other details etc. And click on "save".	API lifecycle should selected as 'created'. API should not be visible on the store.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.1.3.1.Test Suite : Transport

Test Artifact
Mind Map

Test Case APIM-56: Create an API with a WSDL , enable only the http transport and save [Version : 1]
Author: Sewmini Jayaweera
Summary:
Create an API with a WSDL , enable only the http transport and save. Check in the WSDL whether only the endpoints for enabled type of transport are exposed
Execution type: Manual
Estimated exec. duration (min):
Importance: Medium
Artifacts:
Artifacts:
Artifacts:
Automation Test Case:
Requirements None
Keywords: None
Last Result **Not Run**

Test Case APIM-57: Create an API with a WSDL , enable only the https transport and save [Version : 1]
Author: Sewmini Jayaweera
Summary:
Create an API with a WSDL , enable only the https transport and save. Check in the WSDL whether only the endpoints for enabled type of transport are exposed.
Execution type: Manual
Estimated exec. duration (min):
Importance: Medium
Artifacts:
Artifacts:
Artifacts:
Automation Test Case:
Requirements None
Keywords: None
Last Result **Not Run**

Test Case APIM-58: Create an API with a WSDL , enable both https and http transports and save [Version : 1]
Author: Sewmini Jayaweera

Summary:	
Create an API with a WSDL , enable both https and http transports and save. Check in the WSDL whether the endpoints for both enabled type of transports are exposed	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-59: Create an API from the scratch and complete the API until you save and publish it in manage page [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Create an API from the scratch and complete the API until you save and publish it in manage page - API should visible under APIs in API store.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-69: When user select subscription available to all tenants visibility should always set to public [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
User adds an api given visibility as 'visible to my domain' then in the manage page set subscription to 'available to all tenants' when user selects subscription avalable to all tenants visibility should automatically chage to public.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-81: When user selects available to current tenants only, visibility can be selected as users choice [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
User with permission should be able to select any visibility option given subscription as available to current tenat only.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-142: When having many copies of the API publish one of the copies given require re subscription option [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
When there are more than one API versions and only one of the versions are given required re subcription only that version should show require re subscription behavior.	

Preconditions:

1. There should be two copies of the api having verion2 and version3 in created stage.

2. Original API should have a subscription.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Publish API copy version2 without giving require re subscription.	• API version2 should be visible on api store > my subscriptions page of Subscriber who is already subscribed to original under my subscriptions, and new version should show same API throttling level as the original API throttling level. • subscriber should be able to invoke new auto subscribed version. • New version of the API should be visible on the store for new subscriptions. • New version should be visible on API publisher > browse page marked published.		
2	Publish API copy version3 giving require re subscription.	• API verion3 should not be visible on api store > my subscriptions > subscribed APIs page of Subscriber who is already subscribed to original. • New version of the API should be visible on the store for new subscriptions. • New version should be visible on API publisher > browse page marked published.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.2.Test Suite : Edit API

Test Artifact	
Mind Map	

Test Case APIM-217: Check whether user can successfully edit icon. **[Version : 1]**

Author:	Sewmini Jayaweera

Summary:

User who has successfully logged into API publisher should be able to edit API properties define in the design stage and changes should get saved and reflect successfully.

Preconditions:

1. There should be an API (api1) which is created and published already.

2. API publisher user should be successfully logged into API publisher

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to API publisher > browse > api1 > edit page and click on edit icon link and choose a text file and save.	user should get a validation message.		
2	Go to API publisher > browse > api1 > edit page and click on edit icon link and choose a icon, click on 'save' and check whether change made is successfully updated	The icon should get saved and display in both API store and API publisher.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.3.Test Suite : Delete API

Test Artifact	
Mind Map	

1.4.Test Suite : API Versioning/Copy API

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_Publisher/API-Versioning_Copy-API/MindMaps [^]

1.4.1.Test Suite : default version

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_Publisher/API-Versioning_Copy-API/MindMaps [^]

1.4.1.1.Test Suite : With default version

Preconditions

1. There should be a user who is logged into API publisher.

2. User must have a copy of an existing API which is set as the default version.

3. There should be a subscription for the original API.

4. '<DisplayMultipleVersions>' option should set to true in <AM_HOME>/repository/conf/api-manager.xml

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_Publisher/API-Versioning_Copy-API/MindMaps [^]

Test Case APIM-421: Copy an api with {version} strategy in context as the default api and invoke it sucesfully with its new endpoint [Version : 1]				
Author:	Ushani Balasooriya			
Summary:				
Copy an api with {version} strategy in context as the default api and invoke it sucesfully with its new endpoint (E.g., {version} should be replaced by the copied API's version)				
Preconditions:				
1. Preconditions and the steps mentioned in APIM-275 test case should have been sucessfully performed.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Copy the API providing the version as 2.0.0 and publish it	API should be created with the replaced version context and endpoint. It should replaced the 1st 3 steps given in test case APIM-275 and the default api url should not have the version E.g., api1/{version}/test -> api1/2.0.0/test and api1/test for the default {version}/api1 -> 2.0.0/api1 and api1 for the default api1/test/{version} -> api1/test/1.0.0} and api1/test for the default		
2	Invoke the new default endpoint	API should be sucessfully invoked		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-366: Check whether user can publish new copy given deprecate old versions option. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User with publish permission should be able to publish new copy given deprecate old version.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to lifecyle tab of the new copy.	User should reside on the lifecycle tab of the new copy.		
2	Select published, check deprecate old versions and click 'update' button.	New api version should get updated successfully. • In API publisher browse view satatus of old version should marked as deprecated.		
3	Go to API store	API store should view only the new version of the api.		
4	Login to API using subscriber user who has a subscription to th original API and go to my subscription page.	• Original API should marked as deprecate. • New API version should be visible under subscribed APIs.		
5	Invoke old version of the API	Old version of the api should be able to invoke successfully.		
6	Invoke Old version of the API without giving api version.	Invocation should fail with a 404 error (response not found)		
7	Invoke new version of the API using url which has version number.	User should be able to invoke API successfully.		
8	Invoke new version of the API using the URL with no version number.	User should be able to invoke API successfully.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-370: Check whether user can publish new copy given require re-subscription . [Version : 1]	
Author:	Sewmini Jayaweera

Summary:				
User who has publish permission should be able to published new copy given require re-subscription.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to lifecycle tab of the new copy	User should reside on the lifecycle tab of the new copy.		
2	Tick require re-subscription option and click 'update'	New API version should get published successfully with a success message.		
3	Go to API store go to apis tab.	• New copy should list unser recently added side panel. • New copy should list under APIs having correct name, verion and provider details.		
4	Login to API store with the subscriber who has a subscription to the original API and go to my subscription page.	New version should not be visible under subscribed APIs		
5	Invole original API.	Original API should get invoked successfully.		
6	Subscribe to the same application user subscribed to the original api.	User should be able to subscribe to the new version successfully and should be able to see it under subscribed APIs.		
7	Invoke new version with URL without version.	User should be able to invoke API successfully.		
8	Invoke new version with URL with version.	User should be able to invoke api successfully.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-371: Check whether user can publish new copy with both deprecate old version and require re-subscription [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User who has publish permission should be able to publish new copy given both deprecate old version and require re-subscriptions options.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to lifecycle tab of the new api version.	User should successfully redirect to lifecucle tab		
2	Give • Deprecate old version • requre re-subscription options and click update.	• New API copy should get published successfully. • In API publisher browse view satatus of old version should marked as deprecated.		
3	Go to API store	• New version should be visible under APIs as well as recently added section. • Old version should not be visible in the store		
4	Login to API using subscriber user who has a subscription to the original API and go to my subscription page.	• Old version of the API should marked as deprecated • New version should not available under subscribed APIs		
5	Subscribe to the new version using same application user subscribed to the original api.	User should be able to subscribe to the new version successfully and should be able to see it under subscribed APIs.		
6	Invoke deprecated API	User should be able to invoke successfully.		
7	Invoke new api version using url with version	user should be able to invoke new api version successfully.		
8	Invoke new api version using url without version	user should be able to invoke new api version successfully.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.4.1.2.Test Suite : without default version

Preconditions

1. There should be a user who is logged into API publisher.

2. User must have a copy of an existing API which is not set as the default version.

3. There should be a subscription for the original API.

4. '<DisplayMultipleVersions>' option should set to true in <AM_HOME>/repository/conf/api-manager.xml

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_Publisher/API-Versioning_Copy-API/MindMaps [^]

Test Case APIM-363: Check whether user can set the new copy as the default version. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: user who is successfully logged in to API publisher with publisher and creator permission should be able to set new copy as the default version.				
Preconditions: 1. User should be successfully logged into API publisher with creator and publisher permission. 2. There should be an API which is already created. 3. User should reside on overview page of the API				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	click on 'create new version' button.	Sub section which has • a field to specify version, • check box to make it default verion • 'done' and 'cancel' buttons should expand.		
2	specify version number, check default version check box and click 'done'	User should redirect to browse page and new copy should list having • created status • given version number • creator's name • number of users (0)		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-374: Check whether user can publish new copy with both deprecate old version and require re-subscription [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: User who has publish permission should be able to publish new copy given both deprecate old version and require re-subscriptions options.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to lifecycle tab of the new api version.	User should successfully redirect to lifecucle tab		
2	Give • Deprecate old version • requre re-subscription options and click update.	• New API copy should get published successfully. • In API publisher browse view satatus of old version should marked as deprecated.		
3	Go to API store	• New version should be visible under APIs as well as recently added section. • Old version should not be visible in the store		
4	Login to API using subscriber user who has a subscription to the original API and go to my subscription page.	• Old version of the API should marked as deprecated • New version should not available under subscribed APIs		
5	Subscribe to the new version using same application user subscribed to the original api.	User should be able to subscribe to the new version successfully and should be able to see it under subscribed APIs.		
6	Invoke deprecated API	User should be able to invoke successfully.		
7	Invoke new api version using url with version	user should be able to invoke new api version successfully.		
8	Invoke new api version using url without version	Invocation should fail with a error code 404		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-373: Check whether user can publish new copy given require re-subscription . [Version : 1]

Author:	Sewmini Jayaweera
Summary:	
User who has publish permission should be able to published new copy given require re-subscription.	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to lifecycle tab of the new copy	User should reside on the lifecycle tab of the new copy.		
2	Tick require re-subscription option and click 'update'	New API version should get published successfully with a success message.		
3	Go to API store go to apis tab.	• New copy should list unser recently added side panel. • New copy should list under APIs having correct name, verion and provider details.		
4	Login to API store with the subscriber who has a subscription to the original API and go to my subscription page.	New version should not be visible under subscribed APIs		
5	Invole original API.	Original API should get invoked successfully.		
6	Subscribe to the same application user subscribed to the original api.	User should be able to subscribe to the new version successfully and should be able to see it under subscribed APIs.		
7	Invoke new version with URL without version.	Invocation should fail with a error code 404.		
8	Invoke new version with URL with version.	User should be able to invoke api successfully.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-372: Check whether user can publish new copy given deprecate old versions option. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	
User with publish permission should be able to publish new copy given deprecate old version.	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to lifecyle tab of the new copy.	User should reside on the lifecycle tab of the new copy.		
2	Select published, check deprecate old versions and click 'update' button.	New api version should get updated successfully. • In API publisher browse view satatus of old version should marked as deprecated.		
3	Go to API store	API store should view only the new version of the api.		
4	Login to API using subscriber user who has a subscription to th original API and go to my subscription page.	• Original API should marked as deprecate. • New API version should be visible under subscribed APIs.		
5	Invoke old version of the API	Old version of the api should be able to invoke successfully.		
6	Invoke Old version of the API without giving api version.	Invocation should fail with a 404 error (response not found)		
7	Invoke new version of the API using url which has version number.	User should be able to invoke API successfully.		
8	Invoke new version of the API using the URL with no version number.	Invocation should fail with a 404 error (response not found)		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-420: Copy an api with {version} strategy in context and invoke it sucesfully with its endpoint [Version : 1]

Author:	Ushani Balasooriya
Summary:	
Copy an api with {version} strategy in context and invoke it sucesfully with its new endpoint endpoint (E.g., {version} should be replaced by the copied API's version)	
Preconditions:	
1. Preconditions and the steps mentioned in APIM-275 test case should have been sucessfully performed.	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Copy the API providing the version as 2.0.0 and publish it	API should be created with the replaced version context and endpoint. It should replaced the 1st 3 steps given in test case APIM-275. E.g., api1/{version}/test -> api1/2.0.0/test {version}/api1 -> 2.0.0/api1 api1/test/{version} -> api1/test/1.0.0}		
2	Invoke the new endpoint of version 2.0.0	API should be sucessfully invoked		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.4.2.Test Suite : lifecycle

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_Publisher/API-Versioning_Copy-API/MindMaps [^]

Test Case APIM-127: Check whether system allows to get a copy of an api which is in blocked state. **[Version : 1]**

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to copy an API in blocked state.

Preconditions:

1. User with permission should be successfully logged into the API publisher.
2. There should be an API already prototyped in blocked state.
3. User should reside on browse > api > overview page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'copy' button	Section to specify version will expand.		
2	Give a valid version and click done	User should redirected to browse page and new copy should be listed on the page having below details • API name - same as original • API version - specified version when copying • creator - same as original • users - 0 • status - created		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-122: Check whether system allows to get a copy of an api which is in created state. **[Version : 1]**

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to copy an API in created state.

Preconditions:

1. User with permission should be successfully logged into the API publisher.
2. There should be an API already created in created state.
3. User should reside on browse > api > overview page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'copy' button	Section to specify version will expand.		
2	Give a valid version and click done	User should redirected to browse page and new copy should be created having same API details as the original api other than api version. (API version should be the version specified)		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium

Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-123: Check whether system allows to get a copy of an api which is in published state. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to copy an API in published state.

Preconditions:

1. User with permission should be successfully logged into the API publisher.

2. There should be an API already published in created state.

3. User should reside on browse > api > overview page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'copy' button	Section to specify version will expand.		
2	Give a valid version and click done	User should redirected to browse page and new copy should be listed on the page having below details • API name - same as original • API version - specified version when copying • creator - same as original • users - 0 • status - created		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-124: Check whether system allows to get a copy of an api which is in prototyped state. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to copy an API in prototyped state.

Preconditions:

1. User with permission should be successfully logged into the API publisher.

2. There should be an API already prototyped in created state.

3. User should reside on browse > api > overview page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'copy' button	Section to specify version will expand.		
2	Give a valid version and click done	User should redirected to browse page and new copy should be listed on the page having below details • API name - same as original • API version - specified version when copying • creator - same as original • users - 0 • status - created		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-125: Check whether system allows to get a copy of an api which is in deprecated state. [Version : 1]

Author:	Sewmini Jayaweera

Summary:				
Successfully logged in user with permission should be able to copy an API in deprecated state.				
Preconditions:				
1. User with permission should be successfully logged into the API publisher. 2. There should be an API already prototyped in deprecated state. 3. User should reside on browse > api > overview page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'copy' button	Section to specify version will expand.		
2	Give a valid version and click done	User should redirected to browse page and new copy should be listed on the page having below details • API name - same as original • API version - specified version when copying • creator - same as original • users - 0 • status - created		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-126: Check whether system allows to get a copy of an api which is in retired state. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user with permission should be able to copy an API in retired state.				
Preconditions:				
1. User with permission should be successfully logged into the API publisher. 2. There should be an API already prototyped in retired state. 3. User should reside on browse > api > overview page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'copy' button	Section to specify version will expand.		
2	Give a valid version and click done	User should redirected to browse page and new copy should be listed on the page having below details • API name - same as original • API version - specified version when copying • creator - same as original • users - 0 • status - created		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.4.3.Test Suite : Docs

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_Publisher/API-Versioning_Copy-API/MindMaps [^]

1.4.3.1.Test Suite : Existing docs

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_Publisher/API-Versioning_Copy-API/MindMaps [^]

1.4.3.1.1.Test Suite : Edit existing doc content

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_Publisher/API-Versioning_Copy-API/MindMaps [^]

Test Case APIM-419: Check whether user can edit and change content type of an existing hot to type document. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User with permission should be able to edit source of the content in a how to type document.				
Preconditions:				
1. There should be a published new copy taken from an API which has; 'How to' type document given content using a File. 2. User with permission should successfully logged into publisher and reside on docs tab of the new API copy.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on update button of other type document.	Document summery, type and source should open in an editable view.		
2	Edit summary and give URL as the source and update	Changes should get updated successfully and reflect, publisher side as well as in the store side.		
3	Edit how to type document and change source to inline and updae	Changes should get updated and reflect on both publisher and store side.		
4	Edit and change doc type to samples & SDK, source to file and update.	Changes should get updated successfully and reflect on both publisher and store sides.		
5	Edit and change doc type : public forum source : URL and update.	Changes should get updated successfully and reflect on both publisher and store sides.		
6	Edit and change, doc type : other source to inline and update.	Changes should get updated successfully and reflect on both publisher and store sides.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-418: Check whether user can edit existing 'other' type docs which has URL content type. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User with permission should be able to edit existing 'Other' type document content of the new API copy which is specified using a URL.				
Preconditions:				
1. There should be a published new copy taken from an API which has; 'Other' type document given content using a URL. 2. User with permission should successfully logged into publisher and reside on docs tab of the new API copy.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on update button of 'other' type document.	Document summery, type and source should open in an editable view.		
2	Edit summary and give a different URL and click update.	Changes should get updated successfully and reflect, publisher side as well as in the store side.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-417: Check whether user can edit existing 'Support forum' type docs which has URL content type. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User with permission should be able to edit existing 'Support forum' type document content of the new API copy which is specified using a URL.				
Preconditions:				
1. There should be a published new copy taken from an API which has; 'Support forum' type document given content using a URL. 2. User with permission should successfully logged into publisher and reside on docs tab of the new API copy.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on update button of 'support forum' type document.	Document summery, type and source should open in an editable view.		
2	Edit summary and give a different URL and click update.	Changes should get updated successfully and reflect, publisher side as well as in the store side.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-416: Check whether user can edit existing 'Public forum' type docs which has URL content type. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User with permission should be able to edit existing 'Public forum' type document content of the new API copy which is specified using a URL.

Preconditions:

1. There should be a published new copy taken from an API which has; 'Public forum' type document given content using a URL.

2. User with permission should successfully logged into publisher and reside on docs tab of the new API copy.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on update button of 'public forum' type document.	Document summery, type and source should open in an editable view.		
2	Edit summary and give a different URL and click update.	Changes should get updated successfully and reflect, publisher side as well as in the store side.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-415: Check whether user can edit existing 'Samples & SDK' type docs which has URL content type. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User with permission should be able to edit existing 'Samples & SDK' type document content of the new API copy which is specified using a URL.

Preconditions:

1. There should be a published new copy taken from an API which has; 'Samples & SDK' type document given content using a URL.

2. User with permission should successfully logged into publisher and reside on docs tab of the new API copy.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on update button of 'Samples & SDK' type document.	Document summery, type and source should open in an editable view.		
2	Edit summary and give a different URL and click update.	Changes should successfully reflect, publisher side as well as in the store.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-413: Check whether user can edit existing 'other' type docs which has in-line content type. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User with permission should be able to edit existing 'other' type document content of the new API copy which is specified as in-line

Preconditions:

1. There should be a new copy taken from an API which has; 'other' type document given inline content type.

2. New copy should be published

3. User with permission should successfully logged into publisher and reside on docs tab of the new API copy.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on edit content link of the how other document	• User should get redirected to an edit content page. • Page should contain document content opened in a text editor, document name and API version.		
2	Edit and change content and click on 'save & close'	• Changes should get successfully updated. • Chages should reflect in both publisher ans store side.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-412: Check whether user can edit existing 'samples & SDK' type docs which has in-line content type. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User with permission should be able to edit existing 'samples & SDK' type document content of the new API copy which is specified as in-line

Preconditions:

1. There should be a published new copy taken from an API which has; 'Samples & SDK' type document given inline content type.

2. User with permission should successfully logged into publisher and reside on docs tab of the new API copy.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on edit content link of the Samples & SDK type document	• User should get redirected to an edit content page. • Page should contain document content opened in a text editor, document name and API version.		
2	Edit and change content and click on 'save & close'	• Changes should get successfully updated. • Chages should reflect in both publisher ans store side.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-411: Check whether user can edit existing how to type docs which has in-line content type. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User with permission should be able to edit existing 'how to' type document content of the new API copy which is specified as in-line

Preconditions:

1. There should be a published new copy taken from an API which has; 'how to' type document given inline content type.

2. User with permission should successfully logged into publisher and reside on docs tab of the new API copy.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on edit content link of the how to type document	• User should get redirected to an edit content page. • Page should contain document content opened in a text editor, document name and API version.		
2	Edit and change content and click on 'save & close'	• Changes should get successfully updated. • Chages should reflect in both publisher ans store side.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-414: Check whether user can edit existing 'how to' type docs which has URL content type. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User with permission should be able to edit existing 'How to' type document content of the new API copy which is specified using a URL.				
Preconditions:				
1. There should be a published new copy taken from an API which has; 'How to' type document given content using a URL.				
2. User with permission should successfully logged into publisher and reside on docs tab of the new API copy.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on update button of how to type document.	Document summery, type and source should open in an editable view.		
2	Edit summary and give a different URL and click update.	Changes should successfully reflect, publisher side as well as in the store.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.4.3.2.Test Suite : New docs

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_Publisher/API-Versioning_Copy-API/MindMaps [^]

1.4.4.Test Suite : Other

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_Publisher/API-Versioning_Copy-API/MindMaps [^]

Test Case APIM-128: Check whether system validate for existing versions of the same api [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user with permission should get a validation message when trying to copy an api given an existing version.				
Preconditions:				
1. User with permission should be successfully logged into the API publisher.				
2. There should be an API created alredy.				
3. User should reside on browse > api > overview page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'copy' button	Section to specify version will expand.		
2	Give an existing version and click 'done'	User should get a validation message saying there is an api having the same version or similar.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-131: Check whether cancel button on add new version section is working. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user who has permission should be able to exit from add new version section by clicking on cancel button.				
Preconditions:				
1. User with permission should be successfully logged into the API publisher.				
2. There should be an API already created.				
3. User should reside on browse > api > overview page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'copy'	add new version section should expand.		
2	Click on cancel	User should exit from add new api version section.		
Execution type:	Manual			

Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.5.Test Suite : API lifecycle management

Test Artifact	
Mind Map	

Test Case APIM-161: Check whether auto subscription works for new versions of Blocked. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
When Publisher publishes a new version of a blocked API without given require re-subscription option. Subscriber of the API should get auto subscribed to the new version of the API and subscriber should be able to invoke new API versions successfully.				
Preconditions:				
1. An API (api-v1) who has a subscriber should be in blocked stage.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Get a copy of the API(api-v2) and publish the api without giving 'require re-subscription' option.	• New version of the API (api-v2) should get auto subscribed to the subscriber of api-v1 and visible in API store > my subscriptions > subscribed APIs section. • New version should contain published lifecycle status on API and visible on API publisher > browse page. • Subscriber should be able to invoke new api version and api-v1 should not be able to get invoked.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-159: Check whether "DisplayMultipleVersions" functionality is working properly [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
When user set "DisplayMultipleVersions" to true in <am_home>/repository/conf/api-manager.xml, Store should view all the versions of the same API.				
Preconditions:				
1. "DisplayMultipleVersions" should be set to true in <am_home>/repository/conf/api-manager.xml 2. There should be three published versions as version1, version2 and version3 of the same API. 3. API visibility should be public.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API store and check whether all three versions are visible in API store.	All three versions should be visible in the store/		
2	Edit <am_home>/repository/conf/api-manager.xml and set DisplayMultipleVersions to false and restart server and check the store.	Only the latest version of the API should be visible in the store.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-158: Publish an API copy after editing the throttling tier and without re subscription option. [Version : 1]	
Author:	Sewmini Jayaweera

Summary:	
Create a copy of an API which already has a unlimited subscription, edit the throttling tier by disabling gold tier and publish it wihout giving the option 'require re-subscription" - User should get a warning message	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-135: check whether user can change api lifecycle status of a published API to created. [Version : 1]

Author:	Sewmini Jayaweera			
Summary:				
successfully logged in user who has permission should be able to chage lifecycle status of a published API to created state.				
Preconditions:				
1. User with permission should be successfully logged into the API publisher. 2. There should be an published api which can be viewed by public. 3. A subscriber should have subscribed to the API 4. User should reside on browse > api >lifecycle page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Try to change the lifecycle status of the API from published to created.	User should get a warning saying "This API has users subscribed to it" or similar.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-141: Publish a copy of an API with a new version, giving require re subscription & deprecate old version. [Version : 1]

Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user who has permission should be able to publish a new version of an API giving require re subscription and deprecate old versions option at the same time.				
Preconditions:				
1. There should be a copy of API in create lifecycle stage. 2. There should be a subscriber (subscriber1) already subscribed to the original API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Publish new version (v2) of the API giving both require re-subscriptions and deprecate old version options.	• API v2 should be visible on api store > my subscriptions page of Subscriber who is already subscribed to original under my subscriptions, and new version should show same API throttling level as the original API throttling level. • Old version of the api should marked as deprecated in both API publisher > browse and API store > my subscriptions > subscribed APIs pages. • subscriber should be able to invoke both API versions. • New version of the API should be visible on the store for new subscriptions. • New version should be visible on API publisher > browse page marked published.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-140: Publish a copy of an API with a new version without giving require re subscription option. [Version : 1]

Author:	Sewmini Jayaweera

Summary:				
Successfully logged in user who has permission should be able to publish a new version of an API without giving require re subscription option.				
Preconditions:				
1. There should be a copy of API in create lifecycle stage. 2. There should be a subscriber (subscriber1) already subscribed to the original API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Publish new version (v2) of the API without giving require re-subscriptions option.	• API v2 should be visible on api store > my subscriptions page of Subscriber who is already subscribed to original under my subscriptions, and new version should show same API throttling level as the original API throttling level. • subscriber should be able to invoke new auto subscribed version. • New version of the API should be visible on the store for new subscriptions. • New version should be visible on API publisher > browse page marked published.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-139: Publish a copy of an API with a new version given require re subscription option. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user who has permission should be able to publish a new version of an API given require re subscription option.				
Preconditions:				
1. There should be a copy of API in create lifecycle stage. 2. There should be a subscriber (subscriber1) already subscribed to the original API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Publish new version (v2) of the API given require re-subscriptions option.	• API v2 should not be visible on api store > my subscriptions > subscribed APIs page of Subscriber who is already subscribed to original. • New version of the API should be visible on the store for new subscriptions. • New version should be visible on API publisher > browse page marked published.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-138: check behaviour when lifecycle is changed to created from published when there are subscriptions. [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
System should give a warning message when user trying to change API lifecycle status to created from published when there are subscribers already subscribed to the API.	
Preconditions:	
1. User with permission should be successfully logged into the API publisher. 2. There should be an API created having created status. 3. User should reside on browse > api >lifecycle page	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-137: Check whether API behave right when status is changed to blocked to published. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Successfully logged in user who has permission should be able to change life cycle status of an API blocked to published and API should behave correctly in each stage.				
Preconditions: 1. User with permission should be successfully logged into the API publisher. 2. There should be an API created having blocked status.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Check whether API in published stage show correct bahavior.	• API should be available in api store for new subscriptions. • Api publisher > browse page should contain API, marked as published. • API should be visible on API store > mysubscriptions > subscribed APIs page of users who are already subscribed to the API. • Subscribed user should be able to successfully invoke API.		
2	Login to publisher with publish permission and change the lifecycle status from block to published.	• In both API publisher > browse and API store > my subscriptions >subscribed APIs of subscriber1 pages API should visible marked as blocked. • API should not be visible on the store for new subscriptions. • Subscriber should not be able to invoke API.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-136: Check whether life API behave right when lifecycle changes to publish>depre;>blocked>depre;>retired [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Successfully logged in user who has permission should be able to change life cycle status of a created API to published and in each stage API should show correct behaviour.				
Preconditions: 1. User with permission should be successfully logged into the API publisher. 2. There should be an API created having published status. 3. User should reside on browse > api >lifecycle page 4. Published API (API1) should have a subscription (subscriber1).				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Change the lifecycle of API1 to deprecate	• API publisher > browse > API status on API should change as deprecated. • Subscriber1 should view API having marked as deprecated on api in API store > my subscriptions > Subscribed APIs section. • API should not be visible on the stored for new subscribers. • Subscriber1 should be able to invoke API.		
2	Login to the publisher with publisher permission and change the lifecycle from deprecated to Blocked.	• In both API publisher > browse and API store > my subscriptions >subscribed APIs of subscriber1 pages API should visible marked as blocked. • API should not be visible on the store for new subscriptions. • Subscriber should not be able to invoke API.		
3	Login to publisher with publish permission and change lifecycle from Blocked to deprecated.	• API publisher > browse > API status on API should change as deprecated. • Subscriber1 should view API having marked as deprecated on api in API store > my subscriptions > Subscribed APIs section. • API should not be visible on the stored for new subscriptions. • Subscriber1 should be able to invoke API.		
4	Login to publisher with publish permission and change lifecycle from blocked to retired.	• API should not be visible in the store for new subscroptions. • already subscribed users should see api marked retired on API store > my subscriptions > subscribed APIs section. • In API publisher > browse page API should visible marked as retired. • Subscriber should no longer be able to invoke this API.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				

Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-134: check whether user can change API life cycle, create > blocked [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user who has permission should be able to change life cycle status of a created API to published.				
Preconditions:				
1. User with permission should be successfully logged into the API publisher. 2. There should be an API created having created status. 3. User should reside on browse > api >lifecycle page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Select status as blocked and click update.	User should get a success massage saying successfully updated API.		
2	Check whether APi status has changed to blocked on api publisher > browse > particular api details.	Status of the particular API appear as blocked on api publisher>browse page.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-133: check whether user can change API life cycle, create > deprecated [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user who has permission should be able to change life cycle status of a created API to published.				
Preconditions:				
1. User with permission should be successfully logged into the API publisher. 2. There should be an API created having created status. 3. User should reside on browse > api >lifecycle page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Select status as deprecated and click update.	User should get a success massage saying successfully updated API.		
2	Check whether APi status has changed to deprecated on api publisher > browse > particular api details.	Status of the particular API appear as deprecated on api publisher>browse page.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-132: check whether user can change API life cycle, create > retired [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user who has permission should be able to change life cycle status of a created API to published.				
Preconditions:				
1. User with permission should be successfully logged into the API publisher. 2. There should be an API created having created status. 3. User should reside on browse > api >lifecycle page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1				

2	Select status as retired and click update.	User should get a success massage saying successfully updated API.		
3	Check whether APi status has changed to retired on api publisher > browse > particular api details.	Status of the particular API appear as retired on api publisher>browse page.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-84: copy an published API and create a new version and publish it without deprecating the old version [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Change the lifecycle of a copy to published
-don't deprecated old version
-Check whether newest version is listed in store
-Check whether old version is still listed under more apis from same creator
-Whether users can still subscribe to old version
-Invoke new api and previous api

Preconditions:

1.User with permission should be successfully logged into the system.

2. There should be an existing API which is published already.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Create new version give a valid version number.	New version should get list on the browse page of API Publisher		
2	Publish new version without deprecating the old version and Check whether newest version is listed in store	newest version should be listed in the api store.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-85: copy and published API given a new version and publish it given deprecate old versions option. [Version : 1]

Author:	Sewmini Jayaweera
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-86: Publish newest version of an api which has many versions given deprecate old versions option [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user should be able to see prior versions marked as deprecared when user publish newst version given deprecate old versions.

Preconditions:

1. User with permission should be successfully logged into the API publisher.

2. There should be an API version 1.0.0 and 1.2.0 published and version 1.1.0 in created lifecycle state.

3. A user should be subscribed to API version 1.0.0 and 1.2.0 already.

4. User should reside on browse > api > overview page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:

2	Publish API 1.1.0 given deprecate old versions option.	• API version 1.0.0 's status should change to deprecate in api publisher > browse view on the api and In the store when logged in as user who has subscribed the api 1.0.0 and go to my subscriptions page, bottom of the page on api • API version 1.2.0 should remain as published in api publisher > browse > on the api which is version is 1.2.0		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-87: create two versions in created state; publish the new one given deprecate old then publish old one [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with create and publish permission copy an api which is in created state and first publish latest version given deprecate old versions and then publish the old version.

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-89: check whether user can change API life cycle, create > prototyped [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user who has permission should be able to change life cycle status of a created API to published.

Preconditions:

1. User with permission should be successfully logged into the API publisher.

2. There should be an API created having created status.

3. User should reside on browse > api >lifecycle page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1				
2	Select status as prototyped and click update.	User should get a success massage saying successfully updated API.		
3	Check whether APi status has changed to prototyped on api publisher > browse > particular api details.	Status of the particular API appear as retired on api publisher>browse page.		
4	Check whether API list on the prototyped APIs page of the API store.	User should be able to see Api listed on 'prototyped API' page of the API store.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-160: Check whether auto subscription works for new versions of deprecated. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

When Publisher publishes a new version of a deprecated API without given require re-subscription option. Subscriber of the API should get auto subscribed to the new version of the API and subscriber should be able to invoke both API versions successfully.

Preconditions:

1. An API (api-v1) who has a subscriber should be in deprecated stage.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:

1	Get a copy of the API(api-v2) and publish the api without giving 'require re-subscription' option.	• New version of the API (api-v2) should get auto subscribed to the subscriber of api-v1 and visible in API store > my subscriptions > subscribed APIs section. • New version should contain published lifecycle status on API and visible on API publisher > browse page. • Subscriber should be able to invoke both apis		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-349: User should not be create to add a document without document name [Version : 1]

Author:	Shamin Goonetileke

Summary:

Successfully logged in user to API publisher who has permission to add documents should not be able to Create a document without document name

Preconditions:

1. User with permission to add documents should be successfully logged in to the API publisher.

2. There should be an api already published.

3. User who has permission to add documents should reside on docs page of the particular api.

4. A user who has permission to view API in the API store should be successfully logged into the API store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User click on add new document button	Text fields should be appeared		
2	User press add document button without document name	Error message should be appeared message text - " This field is required."		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.6.Test Suite : Scopes

Test Artifact
Mind Map

Test Case APIM-175: Check whether user can add different scopes for different resources. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to assign different scopes for different resources.

Preconditions:

1. User with permission should be logged into the api publisher.

2. There should be an API having be at least 4 scopes (scope1,scope2,scope3,scope4) already created; which has created given jaxrs_basic (service on wso2 application server) as the endpoint in implementation stage and specified separate resource URI for each resource.

3. User should reside on the manage stage of the API.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Assign scope1, scope2, scope3 and scope4 for resource1, resource2, resource3 and resource4 respectively and save and publish Apl.	Assigned scopes should get successfully saved.		
2	Login as a different user who has publisher permission go to edit view and check whether assigned scope can be seen.	Assigned scopes can be seen by the user.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None

Last Result	**Not Run**

Test Case APIM-172: Check whether user gets a validation message when trying to add scopes given invalid roles. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	Successfully logged in user who has permission should get a validation message when trying to save a scope given an invalid role.
Preconditions:	1. User with permission should successfully logged into the API publisher. 2. User should reside on define scope pop up window by clicking on 'add scope' button on manage stage of the API.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Give a valid scope name and key, specify an non-existing role and click 'add scope' button.	User should get a validation message saying role does not exist or similar		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-170: User should get validation message for special characters. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	Successfully logged in user with permission should get validation message when trying to add a scope given special characters.
Preconditions:	1. User with permission should be logged into API Publisher. 2. User should reside 'define scope' pop up window by clicking on the 'add scopes' button on the manage stage of the API.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	add "`or ""="" AND Pass ="" " as the scope key and give a valid name, role description and click 'add scope'`	User should get validation masage for special characte.		
2	Add "& scope1" as the scope key give valid information for other fields and click 'add scope'.			
3	Add as the key give valid information to other fields and click 'add scope'	User should get validation message for the special characters entered.		
4	Give <script> as the scope name give valid details to other fields and click on 'add scope' button	User should get a validation error message.		
5	Give '& name1' as the scope name fill other fields with valid details and click 'add scope'	User should get a validation error mesage.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-168: Check whether compulsory fields are marked with a red star. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	Successfully logged in use who resides on define new scope window should be able to see compulsory fields marked with red asterisks.
Preconditions:	1. User who has permission should successfully logged into the API publisher. 2. User should reside on manage stage of the API.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on "Add Scopes" button.	User should get 'define scope' pop up window.		
2	Check whether scope key, scope name and roles fields are marked as compulsory fields.	scope key, scope name and roles fields should be marked as compulsory fields.		

Execution type:	Manual
Estimated exec. duration (min):	

Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-166: User should be able to add scopes in the manage stage when creating an API [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user should be able to add scopes in the manage stage, when creating a new api.

Preconditions:

1. User with permission should be successfully logged into API publisher.

2. User should complete design and plementation levels by giving valid information in design and implementation stages and come to manage level.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on the 'add scopes' button reside under resources section of the page.	A pop up window should open to define the scope.		
2	Give valid information for scope key (should be a unique name), scope name, role and description and click on 'add scope' button.	Added scope details should be visible above ''add scopes' button. scope name, scope key, roles and description.		
3	save API and again check manage page using edit view.	Added scope should be visible successfully.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-167: Check whether user can add scope to a created API using edit view. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user should be able to add scopes by editing an existing API.

Preconditions:

1. User with permission should be successfully logged into API publisher.

2. There should be an API which is already created.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on the 'add scopes' button reside under resources section of the page.	A pop up window should open to define the scope.		
2	Give valid information for scope key (should be a unique name), scope name, role and description and click on 'add scope' button.	Added scope details should be visible above ''add scopes' button. scope name, scope key, roles and description.		
3	save API and again check manage page using edit view.	Added scope should be visible successfully.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-169: Check whether user gets a validation message for empty compulsory fields. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in use who has permission should get a validation message when user tries to add a scope without defining compulsory fields.

Preconditions:

1. User who has permission should successfully logged into the API publisher.

2. User should reside 'define scope' pop up window by clicking on the 'add scopes' button on the manage stage of the API.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Keep all fields empty and click on 'add scope' button.	User should get validation errors for compulsory fields.		
2	give only scope key and click on 'add scope' button.	User should get compulsory field validation message for scope name and roles.		
3	Give scope key, scope name and click 'add scope' button.	User should get a compulsory field validation error for roles field.		
4	Give scope ey, scope name and roles and click 'add scope' button	User should be able to successfully add scope and added scope with given information should be listed above 'add scopes' button.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-171: Check whether user gets an validation message when trying to add scopes with duplicate key. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should get validation message when trying to create a scope given buplicate scope key and scope name.

Preconditions:

1. User with permission should be successfully logged into API publisher.

2. User should reside on 'define scope' pop up window by clicking on 'add scope' button on the manage stage of the API.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
3	Give duplicate scope key and scope name, specify rolse and description and click 'add scope'	User should get a validation message saying scope already exists or similar.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-173: Check whether user can exit from the define scope window by clicking on the 'x' sign on the window [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to exit from the window by cicking on the 'x' sign on the top right corner of the window.

Preconditions:

1. User with permission should be successfully logged into API publisher.

2. User should reside on 'define scope' pop up window by clicking on 'add scope' button on the manage stage of the API.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
3	Click on 'x' sign on the top right corner of the window.	User should exit fro the define scope window and get redirected to manage stage of the API.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-176: Check whether scopes get loaded successfully into the scope selection drop down in front of resource [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to see all the scopes added in the drop down list in front of the each resource.

Preconditions:				
1. User with permission should be logged into the api publisher. 2. There must be at least 6 scopes already created. 3. User should reside on the manage stage of the API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on scope links on each resource and check whether all existing scopes load into the drop down list.	For each resource existing scopes should lists by it's name.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-174: Check whether user can exit from the define scope window by clicking [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Successfully logged in user with permission should be able to exit from the window by cicking on 'close' button.				
Preconditions: 1. User with permission should be successfully logged into API publisher. 2. User should reside on 'define scope' pop up window by clicking on 'add scope' button on the manage stage of the API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
3	Click on 'close' button.	User should exit fro the define scope window and get redirected to manage stage of the API.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.6.1.Test Suite : edit

Test Artifact	
Mind Map	

Test Case APIM-178: Check whether user can edit and change scope already assigned to the resource into [Version : 1]				
Author:	Sewmini Jayaweera			
Last edit by:	Ushani Balasooriya			
Summary: Successfully logged in user who has permission should be able to edit and change the already existing scope of a resource to another scope.				
Preconditions: 1. User who has permission should be logged into API publisher. 2. There should be an API having more than one scope and resource (resource1) which is assigned a scope(scope2). 2. User should open the API using edit view and reside in the manage stage of the API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Change scope of the resource one from scope1 to a different sope, save API.	Changes should successfully get saved.		
2	invoke the api using old token and check whether user who has specified user role in the previous scope cannot access the resource 1.	User should not be able to access the resource 1		
3	Invoke api using generated user token for given the new scope and check whether user with the specified role in the new scope can access api resource 1.	User should be able to access resource1		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			

Keywords:	None
Last Result	**Not Run**

Test Case APIM-177: Check whether user can edit and change scope already assigned to the resource into a different scope [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user who has permission should be able to edit and change the already existing scope of a resource to another scope.				
Preconditions:				
1. User who has permission should be logged into API publisher. 2. There should be an API having more than one scope and resource (resource1) which is assigned a scope(scope2). 2. User should open the API using edit view and reside in the manage stage of the API.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Change scope of the resource one from scope1 to a different sope, save API.	Changes should successfully get saved.		
2	invoke the api using old token and check whether user who has specified user role in the previous scope cannot access the resource 1.	User should not be able to access the resource 1		
3	Invoke api using generated user token for given the new scope and check whether user with the specified role in the new scope can access api resource 1.	User should be able to access resource1		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.7.Test Suite : API URI templates

Test Artifact
Mind Map

Test Case APIM-73: Ability to define the templates in publisher [Version : 1]	
Author:	Sewmini Jayaweera
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-74: user should be able to define complex uri - templates having many parameters [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Define complex uri-templates; sample uri pattern : /view/{symbol,symbol1}/test/{symbol3,symbol4}/test1	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-75: User should be able to define complex uri-templates with each HTTP verb [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Define complex uri-templates with each HTTP verb - PUT, POST, GET, DELETE	

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-76: Test boundary values in template definition [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Test boundary values in template definition - e.g: trailing space, /, without /, etc.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-77: Test with requests having large number of query parameters [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Test with requests having large number of query parameters (?a=b&c=d&e=f&g=h&i=j&k=l&m=n etc…)	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-78: Test with lengthy URL parameters [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Test with lengthy URL parameters (/a/b/c/d/e/f/g/h/i/j/k)	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-79: Update URI template given a valid URI pattern-URI should be able to invoke the resource successfully [Version : 1]	
Author:	Sewmini Jayaweera
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	

Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-80: Update the template in synapse config in file system, change should reflect on API resource details [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Changes made in the synapse configuration should update the API resource details shown in publisher or the Swagger API definition	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.8.Test Suite : Search functionality

Test Artifact
Mind Map

1.8.1.Test Suite : publisher

Test Artifact
Mind Map

1.8.2.Test Suite : Store

Test Artifact
Mind Map

1.9.Test Suite : API Docs

Test Artifact
Mind Map

1.9.1.Test Suite : Add/Verify Documents

@Sewmini. Please move all test cases under API Publisher/API Docs under this collection

Test Artifact
Mind Map

Test Case APIM-116: Check whether table which contains added documents get updated properly [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user who has permission should be able to see table in api > docs page gets updated with correct information when user is adding documents.				
Preconditions: 1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an already created api 3. User who has permission to add documents should reside on docs page of the particular api. 4. set api doc visibility level in <api_home>/repository/conf/api-manager.xml to true				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Add a new document How to type document and check whether table has updated it's name, type, visibility, modified on correctly.	Table should update document name, type, visibility, modified on, correctly.		
2	Add a new document Samples and SDK type document and check whether table has updated it's name, type, visibility, modified on correctly.	Table should update document name, type, visibility, modified on, correctly.		
3	Add a new document Public forum type document and check whether table has updated it's name, type, visibility, modified on correctly.	Table should update document name, type, visibility, modified on, correctly.		
4	Add a new document support forum type document and check whether table has updated it's name, type, visibility, modified on correctly.	Table should update document name, type, visibility, modified on, correctly.		
5	Add a new document other type document and check whether table has updated it's name, type, visibility, modified on correctly.	Table should update document name, type, visibility, modified on, correctly.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			

Keywords:	None
Last Result	**Not Run**

Test Case APIM-105: Create a document given 'other' as the type and 'file' as the source. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user to API publisher who has permission to add documents should be able to add a new 'other' type document, specifying 'file' as the source.				
Preconditions:				
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api. 4. A user who has permission to view API in the API store should be successfully logged into the API store.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Give a valid name, summery, 'other' as the type and specify type, 'file' as the source and click 'Add document' without selecting a file.	System should give a validation message asking to select a file or similar.		
2	Give a valid name, summery, 'other' as the type and specify the type, 'file' as the source select a valid file and click 'Add document'	Newly added doument should successfully list in the table below add new document section.		
3	Click on open link of the document listed under actions column.	Attached file should get downloaded.		
4	Check whether API sore displays the added document under documentation of the particular API.	Added document should list under other section in the documentation tab of the API.		
5	Click on the download link of the added document (from the documentation section of the api in the api store)	Attached file when creating the document should successfully get downloaded.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-104: Create a document given 'other' as the type and 'URL' as the source [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Succesfully logged in user to API publisher who has permission to add documents should be be able to add a new 'other' type document specifying 'URL' as the source.				
Preconditions:				
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api. 4. A user who has permission to view API in th API store should be successfully logged into the API store.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	A new section will be expanded and visible which has the options to add new dacouments.		
2	Give doc@!23 as the document name, select 'other' as the type and specify type, 'URL' as the source and click 'Add document'	System should prompt a validation message saying special charactes cannot be used in the name or similar.		
3	Give a different name with special characters, select 'other' as the type, 'URL'as the source and click 'Add document'	System should give a validation message for invalid name or similar.		
4	Give a valid name & summery, select 'How To' as the type, 'URL'as the source and click 'Add document' without giving a URL and specifying other type.	Required fied validation should be prompted for both url and other fields.		
5	Give a valid name & summery, select 'other' as the type, specify type, 'URL'as the source, give a valid URL and click 'Add document'	The document should get added to documents table successfully.		
6	CLick on 'view' link of the added document listed under documents table.	User should get redirected to the page of the specified URL in the previous step.		
7	In the API store go to documentation tab of the particular API and check whether newly added document is there under 'other'	Newly added document should be visible under 'other' category.		
8	Click on the 'view content' link of the document.	User should get redirected to the page of the given URL when adding the document.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			

Last Result	**Not Run**

Test Case APIM-103: Create a document given 'other' as the type and in-line as the source. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user to API publisher who has permission to add documents should be be able to add a new 'other' type document specifying 'in-line' as the source.				
Preconditions:				
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api. 4. A user who has permission to view API in the API store should be successfully logged into the API store.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on add new document	Add new document section should expand and available.		
2	Give a name which has spaces, summary, select 'other' as type, specify other type, 'in-line' as the source and click on 'add document'	Newly added document should list under the add new document section in a table.		
3	Click on edit content link	user should get redirected to a page having document name, version and a text editor which user can use to add content.		
4	Add content and click 'cancel'	User should get redirected to the document listing table on the docs page of the api.		
5	Again got to edit content page add content and click 'save and close'	Added content should successfully get saved and user should get redirected to the docs page where it has the document listing table.		
6	Go to documentation page of the particular API in API store and check whether added document is visible under 'other' section.	Added document should be visible under 'other' section of the documentation tab of the particular API.		
7	Click on 'view content' link and see whether added content is visible to user who is logged into the sore who can view the API.	User should get redirected to a page which contains added content.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-102: Create a document given 'support forum' as the type and 'file' as the source. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user to API publisher who has permission to add documents should be able to add a new 'support forum' type document specifying 'file' as the source.				
Preconditions:				
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api. 4. A user who has permission to view API in the API store should be successfully logged into the API store.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Give a valid name, summery, 'support forum' as the type, 'file' as the source and click 'Add document' without selecting a file.	System should give a validation message asking to select a file or similar.		
2	Give a valid name, summery, 'support forum' as the type, 'file' as the source select a valid file and click 'Add document'	Newly added doument should successfully list in the table below add new document section.		
3	Click on open link of the document listed under actions column.	Attched file should get downloaded.		
4	Check whether API sore displays the added document under documentation of the particular API.	Added document should list under support forum section in the documentation tab of the API.		
5	Click on the download link of the added document (from the documentation section of the api in the api store)	Attached file when creating the document should successfully get downloaded.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-101: Create a document given 'support forums' as the type and 'URL' as the source [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Succesfully logged in user to API publisher who has permission to add documents should be be able to add a new 'public forums' type document specifying 'URL' as the source.				
Preconditions:				
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api. 4. A user who has permission to view API in th API store should be successfully logged into the API store.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	A new section will be expanded and visible which has the options to add new dacouments.		
2	Give a valid name & summery, select 'support forum' as the type, 'URL'as the source, give a valid URL and click 'Add document'	The document should get added to documents table successfully.		
3	CLick on 'view' link of the added document listed under documents table.	User should get redirected to the page of the specified URL in the previous step.		
4	In the API store go to documentation tab of the particular API and check whether newly added document is there under 'support forum'	Newly added document should be visible under 'support forum' section.		
5	Click on the 'view content' link of the document.	User should get redirected to the page of the given URL when adding the document.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-100: System should not allow to select 'in-line' source type given 'support forum' as the type [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user to API publisher who has permission to add documents should not be able to select source as in-line once the user selects type as 'support forum'				
Preconditions:				
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	Add new document section should expand.		
2	Give a name, summery, select type as 'public forum' and try to select source as 'in-line'	user should not be able to select 'in-line' radio button.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-99: Create a document given 'public forum' as the type and 'file' as the source. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user to API publisher who has permission to add documents should be able to add a new 'public forum' type document specifying 'file' as the source.				
Preconditions:				
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api. 4. A user who has permission to view API in the API store should be successfully logged into the API store.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Give a valid name, summery, 'public forum' as the type, 'file' as the source and click 'Add document' without selecting a file.	System should give a validation message asking to select a file or similar.		

2	Give a valid name, summery, 'public forum' as the type, 'file' as the source select a valid file and click 'Add document'	Newly added doument should successfully list in the table below add new document section.		
3	Click on open link of the document listed under actions column.	Attched file should get downloaded.		
4	Check whether API sore displays the added document under documentation of the particular API.	Added document should list under public forum section in the documentation tab of the API.		
5	Click on the download link of the added document (from the documentation section of the api in the api store)	Attached file when creating the document should successfully get downloaded.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-98: Create a document given 'public forums' as the type and 'URL' as the source [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Succesfully logged in user to API publisher who has permission to add documents should be be able to add a new 'public forums' type document specifying 'URL' as the source.

Preconditions:

1. User with permission to add documents should be successfully logged in to the API publisher.

2. There should be an api already published.

3. User who has permission to add documents should reside on docs page of the particular api.

4. A user who has permission to view API in th API store should be successfully logged into the API store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	A new section will be expanded and visible which has the options to add new dacouments.		
2	Give a valid name & summery, select 'public forum' as the type, 'URL'as the source, give a valid URL and click 'Add document'	The document should get added to documents table successfully.		
3	CLick on 'view' link of the added document listed under documents table.	User should get redirected to the page of the specified URL in the previous step.		
4	In the API store go to documentation tab of the particular API and check whether newly added document is there under 'public forum'	Newly added document should be visible under 'public forum' category.		
5	Click on the 'view content' link of the document.	User should get redirected to the page of the given URL when adding the document.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-96: Create a document given 'Samples and SDK' as the type and 'file' as the source. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user to API publisher who has permission to add documents should be able to add a new 'Samples & SDK' type document, specifying 'file' as the source.

Preconditions:

1. User with permission to add documents should be successfully logged in to the API publisher.

2. There should be an api already published.

3. User who has permission to add documents should reside on docs page of the particular api.

4. A user who has permission to view API in the API store should be successfully logged into the API store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Give a valid name, summery, 'samples & SDK' as the type, 'file' as the source and click 'Add document' without selecting a file.	System should give a validation message asking to select a file or similar.		
2	Give a valid name, summery, 'Samples & SDK' as the type, 'file' as the source select a valid file and click 'Add document'	Newly added doument should successfully list in the table below add new document section.		
3	Click on open link of the document listed under actions column.	Attched file should get downloaded.		
4	Check whether API sore displays the added document under documentation of the particular API.	Added document should list under samples section in the documentation tab of the API.		
5	Click on the download link of the added document (from the documentation section of the api in the api store)	Attached file when creating the document should successfully get downloaded.		

Execution type:	Manual
Estimated exec. duration (min):	

Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-95: Create a document given 'Samples & SDK' as the type and 'URL' as the source [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Succesfully logged in user to API publisher who has permission to add documents should be able to add a new 'SAMPLES & SDK' type document specifying 'URL' as the source.

Preconditions:

1. User with permission to add documents should be successfully logged in to the API publisher.

2. There should be an api already published.

3. User who has permission to add documents should reside on docs page of the particular api.

4. A user who has permission to view API in th API store should be successfully logged into the API store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	A new section will be expanded and visible which has the options to add new dacouments.		
2	Give doc@!23 as the document name, select 'samples & SDK' as the type, 'URL' as the source and click 'Add document'	System should prompt a validation message saying special charactes cannot be used in the name or similar.		
3	Give a valid name & summery, select 'Samples & SDK' as the type, 'URL'as the source and click 'Add document' without giving a URL.	Required fied validation should be prompted to url field.		
4	Give a valid name & summery, select 'Samples & SDK' as the type, 'URL'as the source, give a valid URL and click 'Add document'	The document should get added to documents table successfully. Type column of the added document should have samples.		
5	CLick on 'view' link of the added document listed under documents table.	User should get redirected to the page of the specified URL in the previous step.		
6	In the API store go to documentation tab of the particular API and check whether newly added document is there under 'Samples'	Newly added document should be visible under 'Samples' category.		
7	Click on the 'view content' link of the document.	User should get redirected to the page of the given URL when adding the document.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-90: Check whether system validates for empty fields in add new document page [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in use who has permission should get a validation messages when user tries to save a document leaving compulsory fields empty

Preconditions:

1. User who has permission to add documents should successfully log in to the publisher.

2. There should be an API created already.

3. User should reside on docs tab on the partucular API's overview page.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Keep 'Name' field empty and click 'Add Document'	System should give a validation message.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-91: Successfully add new in-line, 'how to' type document [Version : 1]

Author:	Sewmini Jayaweera
Summary:	
Successfully logged in user who has permission should be able to add a document given 'Hot To' as the type and 'in-line' as the source.	
Preconditions:	
1. User who has permission to add documents should be successfully logged in to the publisher. 2. There should be an API already created and published. 3. User should reside on the 'Docs' tab of the overview page of the particular API. 4. A user who has permission to view API in th API store should be successfully logged into the API store.	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Add a document given a vaid name and summary of type 'how to' source 'in-line'	Added document should list in a table with name, type, modified on, actions. Action field sould contain edit content, update, delete links		
2	Check whether added document is visible under documentation of the particular API in the API store.	Newly added document should list under 'how to' category.		
3	Check whether 'view content' link under the added document opens the document. (should be visible under documentation of the particular API in the API store)	Once user clicks on the 'view content' link of the document user should get redirected to the document page having "No document content have been added yet." as the content body.		
4	In API publisher, Go to edit content page of the document added in previous step.	A page should open which has a text editor enabling user to add content. Page should also contain document name and version.		
5	Add some content click on 'cancel'	User should get redirected docs page and added document should be still visible on the table.		
6	Again go to 'edit content' page add content and click on 'save'	content added should get saved.		
7	Append more content to the previously added content and click 'save and close'	Added content should get saved, user should exit from the current page and come back to the document listing table .		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-92: Create a document given 'How to' as the type and 'URL' as the source [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Succesfully logged in user to API publisher who has permission to add documents should be be able to add a new 'How To' type document specifying 'URL' as the source.	
Preconditions:	
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api. 4. A user who has permission to view API in th API store should be successfully logged into the API store.	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	A new section will be expanded and visible which has the options to add new dacouments.		
2	Give doc@!23 as the document name, select 'How To' as the type, 'URL' as the source and click 'Add document'	System should prompt a validation message saying special charactes cannot be used in the name or similar.		
3	Give a different name with special characters, select 'How To' as the type, 'URL'as the source and click 'Add document'	System should give a validation message for invalid name or similar.		
4	Give a valid name & summery, select 'How To' as the type, 'URL'as the source and click 'Add document' without giving a URL.	Required fied validation should be prompted to url field.		
5	Give a valid name & summery, select 'How To' as the type, 'URL'as the source, give a valid URL and click 'Add document'	The document should get added to documents table successfully.		
6	CLick on 'view' link of the added document listed under documents table.	User should get redirected to the page of the specified URL in the previous step.		
7	In the API store go to documentation tab of the particular API and check whether newly added document is there under 'How To'	Newly added document should be visible under 'How to' category.		
8	Click on the 'view content' link of the document.	User should get redirected to the page of the given URL when adding the document.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	

Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-93: Create a document given 'How To' as the type and 'file' as the source. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user to API publisher who has permission to add documents should be able to add a new 'How To' type document specifying 'file' as the source.

Preconditions:

1. User with permission to add documents should be successfully logged in to the API publisher.
2. There should be an api already published.
3. User who has permission to add documents should reside on docs page of the particular api.
4. A user who has permission to view API in the API store should be successfully logged into the API store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Give a valid name, summery, 'How to' as the type, 'file' as the source and click 'Add document' without selecting a file.	System should give a validation message asking to select a file or similar.		
2	Give a valid name, summery, 'How to' as the type, 'file' as the source select a valid file and click 'Add document'	Newly added doument should successfully list in the table below add new document section.		
3	Click on open link of the document listed under actions column.	Attched file should get downloaded.		
4	Check whether API sore displays the added document under documentation of the particular API.	Added document should list under how to category in the documentation tab of the API.		
5	Click on the download link of the added document (from the documentation section of the api in the api store)	Attached file when creating the document should successfully get downloaded.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-94: Create a document given 'samples & sdk' as the type and in-line as the source. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user to API publisher who has permission to add documents should be be able to add a new 'samples & SDK' type document specifying 'in-line' as the source.

Preconditions:

1. User with permission to add documents should be successfully logged in to the API publisher.
2. There should be an api already published.
3. User who has permission to add documents should reside on docs page of the particular api.
4. A user who has permission to view API in the API store should be successfully logged into the API store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on add new document	Add new document section should expand and available.		
2	Give a name which has spaces, summary, select 'Samples & SDK' as type and 'in-line' as the source and click on 'add document'	Newly added document should list under the add new document section in a table.		
3	Click on edit content link	user should get redirected to a page having document name, version and a text editor which user can use to add content.		
4	Add content and click 'cancel'	User should get redirected to the document listing table on the docs page of the api.		
5	Again got to edit content page add content and click 'save and close'	Added content should successfully get saved and user should get redirected to the docs page where it has the document listing table.		
6	Go to documentation page of the particular API in API store and check whether added document is visible under samples.	Added document should be visible under samples section of the documentation tab of the particular API.		
7	Click on 'view content' link and see whether added content is visible to user who is logged into the sore who can view the API.	User should get redirected to a page which contains added content.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None

Keywords:	None
Last Result	**Not Run**

Test Case APIM-97: System should not allow to select 'in-line' source type given 'public forum' as the type [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user to API publisher who has permission to add documents should not be able to select source as in-line once the user selects type as 'public forum"				
Preconditions:				
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	Add new document section should expand.		
2	Give a name, summery, select type as 'public forum' and try to select source as 'in-line'	user should not be able to select 'in-line' radio button.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-347: User should not be able to Create a document with the same name that already added [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user to API publisher who has permission to add documents should not be able to Create a document with the same name that already added				
Preconditions:				
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api. 4. A user who has permission to view API in the API store should be successfully logged into the API store.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User click on add new document button	Text fields should be appeared		
2	User add a already existing name to the document name and press add document	Error message should be appeared message text - "Duplicate Document Name."		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-348: User added a document with summary [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user to API publisher who has permission to add documents should not be able to Create a document with a summary				
Preconditions:				
1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an api already published. 3. User who has permission to add documents should reside on docs page of the particular api. 4. A user who has permission to view API in the API store should be successfully logged into the API store.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User click on add new document button	Text fields should be appeared		

2	User gives doc name	name should be succesfully added		
3	User give a summary	Summary should be successfully added		
4	User selects doc type and source	Both should be successfully seletced		
5	User press add document button	Document should be successfully saved with the summary and it should be visible on the store		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

1.9.2.Test Suite : Update documents

Test Artifact
Mind Map

Test Case APIM-165: Check whether system save changes when user adds special characters to the content of inline docs [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Successfully logged in user who has permission to add documents should be able to update the content of the documents given special characters such as @, ;, &, % for 'in-line' documents.				
Preconditions: 1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an published api which is source is given as in-line. 3. User who has permission to add documents should reside on docs page of the particular api.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on edit content link	User will get redirected to edit content page.		
2	Edit content adding & , @, %, * symbols and save.	Changes should get successfully updated.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-113: Check whether system doesn't allow user to change source as 'in-line' for forum type documents. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Successfully logged in user who has permission to add documents should not be able to update source of the documents to 'in-line' for both public or support forums when updating the documents.				
Preconditions: 1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an published api with documents of type 'public forum' and 'support forum'. 3. User who has permission to add documents should reside on docs page of the particular api.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on update link listed under actions column of the 'public forum' type document.	User should get redirected to an edit document page. Document name should be visible in a non-editabe field. Summary, visibility type and source should be editable/changable		
2	Try to change source of the document to in-line	In-line radio button should be non selectable.		
3	Click on update link listed under actions column of the 'support forum' type document.	User should get redirected to an edit document page. Document name should be visible in a non-editabe field. Summary, visibility type and source should be editable/changable		
4	Try to change source of the document to in-line	In-line radio button should be non selectable.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-112: Check whether system allows user to edit content of already created in-line documents. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user who has permission to add documents should be able to edit document content of documents already created given the source as 'in-line'.

Preconditions:
1. User with permission to add documents should be successfully logged in to the API publisher.

2. There should be an published api with a document with content which is given source as 'in-line'

3. User who has permission to add documents should reside on docs page of the particular api.

4. User with permission should be able to view document from the store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Check whether document details are listed on the docs page of the API.	Document should be listed in a table which has name, type, visibility, modified on and action columns.		
2	Click on 'edit content' link in order to go to edit content, make changes to existing content and click on 'save'	Cheges should successfully get saved.		
3	Append more content and click 'save and close'	User should exit from the edit content view and redirect to the document list.		
4	Try accessing the content of the document from the API store and check whether chages made reflects successfully.	When user view the content the changed made should be visible to the user successfully.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-114: Check whether user can update documents created already. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to update summary, visibility, source and type of the document which is already created.

Preconditions:
1. User with permission to add documents should be successfully logged in to the API publisher.

2. There should be an published api with a 'How to' type document given source as 'in-line'

3. User who has permission to add documents should reside on docs page of the particular api.

4. User with permission should be able to view document from the store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to edit page of the document by clicking on update link of the document.	Other than name field all the other fields and options should be editable.		
2	Go to edit page and change, • summary • visibility to same as api • type to sample sdk • source to url and click on update. Then Go to API store and view updated document	• check whether user can view changes from the api store.		
3	Go to edit page and change • summary • visibility to private • keep type as sample sdk • source to in-line and click on update. Then Go to API store and view updated document	• Only the users who have permission to create or/and publish should be able to view the updated document. • document should list as a smalpe type document. • user should be able to see view content link • Summary should reflect the changes made		

4	Go to edit page and change • visibility to mydomain • keep type as sampe sdk • keep summary as it is • source to file, select a valid file and click on update, Then Go to API store and view updated document	• Only the users who are in the same tenant domain should be able to view the updated document. • document should list as a smalpe type document. • user should be able to see download link and should be able to download file attached		
5	Go to edit page and change • summary • type to public forum • visibility to 'same as api visibility' • keep source as file and attach a different file and click on update. Then Go to API store and view updated document	• Only the users who can view API should be able to view updated document. • Document should list as a public forum • user should be able to download the attached document.		
6	Go to edit page and change • summary • type to support forum • visibility to private • source to url, spacify a valid url and click on update, then go to API store and view updated document.	• Only the users who have permission to create or/and publish should be able to view the updated document. • Updated document should list as a support forum • user should be able to access the specified link. • Summary should reflect the changes made.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-115: User should get validation message when trying to update document having an empty source URL [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user who has permission should get a validation message when trying to update document after selecting source as URL and keep URL field empty.

Preconditions:
1. User with permission to add documents should be successfully logged in to the API publisher.

2. There should be an published api with a document

3. User who has permission to add documents should reside on docs page of the particular api.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click 'update' link of the document	User should get redirected to edit document page.		
2	Select URL as the source, keep URL field empty and click update	User should get a validation message saying url is a required field or similar.		
3	Select file as the source and click Update without selecting a file.	User should get a validtion message saying file cannot be empty or similar.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

1.9.3.Test Suite : Delete Documents

Test Artifact
Mind Map

Test Case APIM-121: Check whether documents of type other get deleted successfully. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to delete other type documents successfully.

Preconditions:

1. user with permission shold be successfully logged in to the system

2. There should be a published API with three other type documents having in-line, url, file sources as doc1, doc2, doc3 successfully.

3. User should reside on browse > api > docs page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on delete link listed under actions column of doc1	User should get a delete confirmation message.		
2	Click 'yes' on delete confirmation button.	Doc1 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
3	Click on delete link listed under actions column of doc2	user should get a delete confirmation message.		
4	Click 'yes' on delete confirmation button	Doc2 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
5	Click on delete link listed under actions column of doc3	User should get a delete confirmation message.		
6	Click 'yes' on delete confirmation button	Doc3 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-120: Check whether documents of type 'support forum' get deleted successfully. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	
Successfully logged in user with permission should be able to delete support forum type documents successfully.	
Preconditions:	
1. user with permission shold be successfully logged in to the system 2. There should be a published API with three 'support forum' documents having url, file sources as doc1, doc2 successfully. 3. User should reside on browse > api > docs page	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on delete link listed under actions column of doc1	User should get a delete confirmation message.		
2	Click 'yes' on delete confirmation button.	Doc1 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
3	Click on delete link listed under actions column of doc2	user should get a delete confirmation message.		
4	Click 'yes' on delete confirmation button	Doc2 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-119: Check whether documents of type 'public forum' get deleted successfully. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	
Successfully logged in user with permission should be able to delete public forum type documents successfully.	
Preconditions:	
1. user with permission shold be successfully logged in to the system 2. There should be a published API with three 'public forum' documents having url, file sources as doc1, doc2 successfully. 3. User should reside on browse > api > docs page	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on delete link listed under actions column of doc1	User should get a delete confirmation message.		
2	Click 'yes' on delete confirmation button.	Doc1 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
3	Click on delete link listed under actions column of doc2	user should get a delete confirmation message.		
4	Click 'yes' on delete confirmation button	Doc2 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
Execution type:	Manual			

Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-118: Check whether documents of type 'sample SDK' gets deleted successfully. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to delete sample sdk type documents successfully.

Preconditions:

1. user with permission shold be successfully logged in to the system

2. There should be a published API with three sample type documents having in-line, url, file sources as doc1, doc2, doc3 successfully.

3. User should reside on browse > api > docs page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on delete link listed under actions column of doc1	User should get a delete confirmation message.		
2	Click 'yes' on delete confirmation button.	Doc1 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
3	Click on delete link listed under actions column of doc2	user should get a delete confirmation message.		
4	Click 'yes' on delete confirmation button	Doc2 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
5	Click on delete link listed under actions column of doc3	User should get a delete confirmation message.		
6	Click 'yes' on delete confirmation button	Doc3 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-117: Check whether documents of type how to get deleted successfully. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to delete how to type documents successfully.

Preconditions:

1. user with permission shold be successfully logged in to the system

2. There should be a published API with three how to type documents having in-line, url, file sources as doc1, doc2, doc3 successfully.

3. User should reside on browse > api > docs page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on delete link listed under actions column of doc1	User should get a delete confirmation message.		
2	Click 'yes' on delete confirmation button.	Doc1 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
3	Click on delete link listed under actions column of doc2	user should get a delete confirmation message.		
4	Click 'yes' on delete confirmation button	Doc2 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		
5	Click on delete link listed under actions column of doc3	User should get a delete confirmation message.		
6	Click 'yes' on delete confirmation button	Doc3 should delete successfully and no longer visible in the table nor under documentation of the particular api in the store.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	

Requirements	None
Keywords:	None
Last Result	**Not Run**

1.10.Test Suite : Sequences

Test Artifact
Mind Map

1.11.Test Suite : Resource Level Authentication Types

Test Artifact
Mind Map

Test Case APIM-422: A user should be able to set different types of resource level authentication types for the resource [Version : 1]

Author:	Ushani Balasooriya			
Summary: A sucessfully logged in user to the store should be able to set different types of resource level authentication types for the resources				
Preconditions: 1. A publisher user should be logged in to the store and create an API with GET, POST, PUT and OPTIONS recources types and go to the Manage stage				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Set Resource level type as "Application" for GET resource	Resource level authentication type should be set as Application only for GET		
2	Set Resource level type as "Application User" for POST resource	Resource level authentication type should be set as Application User only for POST		
3	Set Resource level type as "Application and Application User" for PUT resource	Resource level authentication type should be set as Application and Application User only for PUT		
4	Set Resource level type as "None" for OPTIONS resource	Resource level authentication type should be set as None only for OPTIONS		
5	Then update the resource level type in to "Application User" for OPTIONS resource	It should be sucessfully updated and set the new value		
6	Now publish the paticular api and make a copy as the version 2.0.0	Copied API should have the same resource values set in above steps		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

2.Test Suite : API Store

Test Artifact
Mind Map

2.1.Test Suite : Application management

Test Artifact
Mind Map

Test Case APIM-196: Check where system redirects logged in user to my applications page successfully by clicking on my applications menu items. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: User who is successfully logged into the API store should be able to access my applications page by clicking on the my applications menu item.				
Preconditions: 1. System should be up and running. 2. User with permission should be logged into api store.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on api store > my applications	• User should get redirected to my applications page. • My application page should contain add new application form and list applications which are already		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-195: Check whether user cannot access application page without signing into the API manager store. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: User who is not logged in to the the system should not be able to access api store > applications page.				
Preconditions: 1. System should be up and running.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to API manager store and click on 'my applications' tab of the menu bar.	User should be prompted a login screen.		
2	Give valid user name and password and click on login button.	User should successfully get logged into the system and reside on my applications page of the user.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

2.1.1.Test Suite : Delete Application

Test Artifact
Mind Map

Test Case APIM-203: Check whether application gets deleted properly and not allowing users to use application keys any longer. [Version : 1]	
Author:	Sewmini Jayaweera
Summary: Successfully logged in user to API store should not be able to invoke api using already generated application token after application is being deleted.	
Preconditions: 1. User should have created an application (app1), subacribed to an api (api1) using application, generated application keys using UI given Validity Time as -1 and then delete the application.	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Try to invoke api1 using appliation token of deleted applciation (app1)	User should not be able to invoke api using deleted application tokens.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-202: Check whether user can delete created applications successfully [Version : 1]

Author: Sewmini Jayaweera

Summary:

Successfully logged in user to API store should be able to delete created applications successfully.

Preconditions:

1. User should be successfully logged into the API store.

2. User should reside on API store > my applicartion > add new application page.

3. There should be an application created already.

4. Application user must have already generated application keys for a subscribed api (api1).

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on the delete link under actions column in the of the relevant application listed, which needs to get deleted.	System should prompt user an confirmation pop-up massage saying "This will cancel all the existing subscriptions and keys associated with the application." or similar.		
2	Click on 'yes' button on the pop up massage.	• Application should get successfully deleted. • Deleted application should no longer list on the bottom of the my applications page.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-204: Check whether application consumer key and consumer secret is getting deleted once user delete appli [Version : 1]

Author: Sewmini Jayaweera

Summary:

When Application user deletes an application which has already generated consumenr key and consumer secret, access token, user should no longer use them to generate user tokens.

Preconditions:

1. There should be a deleted application (app 1) which had a subscribed api (api 1) under it and which had generated application keys.

2. Cache should be cleaned.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke token API using consumer key, consumer secret and valid user credentials in order to generate a user token.	User should get an error and no user token should get generated.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

2.1.2.Test Suite : Add Application

Test Artifact	
Mind Map	

Test Case APIM-201: Check whether user can see the usage details by clicking on the '?' sing of the callback URL. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	Successfully logged in user to API store should be able to view callback url usage by clicking on the '?' icon infront of the callback RUL field on API store > my applicartion > add new application page.
Preconditions:	1. User should be successfully logged into the API store. 2. User should reside on API store > my applicartion > add new application page.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on the '?' icon in front of the callback RUL field.	A usage description of the callback URL should appear below the callback URL field.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-198: Check whether compulsory fields of the add application form are marked with an asterisk. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	User who is successfully logged into API store, should be able to see compulsory fields of 'add new application' form are marcked with a red asterisks
Preconditions:	1. User should be successfully logged into API manager store and resice on API store > my applications page.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Check whether name and throttling tier is marked with red asterisks.	name and throttling tier should be marked with red asterisks.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-197: Check whether system allows user to add a new application successfully. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	Successfully logged in user to API store, should be able to add a new application successfully.
Preconditions:	1. System should be up and running. 2. There should be a user registered in API store. 3. User should be successfully logged into the API store. 4. User should reside on API store > my applications page.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Enter name, throttling tier, callback URL, description and click on add button.	• User should get a massage saying application is being successfully added or similar. • User should be able to view newly added application on the bottom of the my applications page. • Newly added application should list correct information under each column of the table.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	

Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-199: Check whether compulsory fields of the 'add new application' form is being validated. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User who is successfully logged into the API store should get a validation message when trying to add an application without specifying compulsory fields.				
Preconditions:				
1. User should be successfully logged into API manager store and resice on API store > my applications page.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Keep all fields empty and click 'save'	User should get a validation message for name field.		
2	Keep name field empty select the throttling tier, give a decritption and save.	User should get a validation message for name field.		
3	Give a valid name select throottling tier and save.	Aplication should get saved successfully.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-200: Check where name field of the 'add new application' form get validated for special characters. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User who is successfully logged into the API store should get a validation message when trying to add an application given special characters in the name field.				
Preconditions:				
1. User should be successfully logged into API manager store and resice on API store > my applications page.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	specify a name with an special character fill other fields with information and click 'save'	User should get a validation message for special characters used in name field.		
2	Give a name with & on the name field, valid information for other fields and save.	User should be able to add the application successfully.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

2.1.3.Test Suite : Edit Application

Test Artifact
Mind Map

Test Case APIM-206: Check whether the cancel button of the edit view works correctly. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User who is successfully logged into the api store should be able to click on cancel button on the edit view and exit from the edit view.				
Preconditions:				
1. User with permission should be successfully logged into api store.				
2. There should be ab existing applicatioon (app1) created already.				
3. User should reside on api store > my applications page.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on edit link listed under the actions column of the table at the bottom of the page	app 1 information fields should open as editable fields.		
2	Click on cancel button after changing the name.	User should exit from the edit view. and changes should NOT get saved.		
Execution type:	Manual			

Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-205: Check whether user can edit created Applications. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User who is successfully logged into the api store should be able to edit existing appliation name, throttling tier, callback url, description and save changes successfully.

Preconditions:

1. User with permission should be successfully logged into api store.

2. There should be ab existing applicatioon (app1) created already.

3. User should reside on api store > my applications page.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on edit link listed under the actions column of the table at the bottom of the page	app 1 information fields should open as editable fields.		
2	Edit name and click save	New chages should get saved.		
3	Open edit view change throtting tier and save.	throttling tier chages should get saved successfully.		
4	Open edit view change callback url and save	changes made to callback url shold get changed successfully.		
5	Open edit view update edit view with a content having & and few other special characters and save.	Changes should gert saved successfully.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

2.2.Test Suite : Subscription management

Test Artifact
Mind Map

2.3.Test Suite : Forums

Test Artifact
Mind Map

2.4.Test Suite : Token generation

Test Artifact
Mind Map

2.4.1.Test Suite : application token

Test Artifact
Mind Map

Test Case APIM-207: Check whether user can successfully generate application token using api store UI. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Use who has successfully logged into API store should be able to generate application tokens by clicking on the 'generate' button on the my subscriptions page.

Preconditions:

1. User should be successfully logged into API store

2. API (api1) should contain resources given auth type of the resources to application.

3. User should be subscribed to an API (api1) which has a production URL and a sandbox URL using an application (app1) which is throttling is set to unlimited.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Select application as app1 from the 'Application with subscription' list.	App1 should get successfully selected.		
2	Expand 'keys - production' section, give 1800 as validity time and click generate.	Consumer key, consumer secret and access token (token-p1) should generate successfully.		

3	Expand 'Keys - sandbox' , give 1800 (seconds) as validity time and click 'generate' button.	Consumet key, consumer secret and a access token (token-s1) should get generated successfully.		
4	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated properly with generated access tokens and relavent validity times.	• Table IDN_OAUTH2_ACCESS_TOKEN should contain both sandbox and production access tokens in active state. • VALIDITY_PERIODs od both records should recorded in milliseconds correcly.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-208: Check whether user can use generated application token to invoke API. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User who is successfully logged into API store should be able to invoke API (api1) which contains resources which are auth type is set to application.

Preconditions:

1. User should be successfully logged into the system

2. User should have executed test case no : 207 and have application tokens generated.

3. access tokens should not have expired. (generated access tokens are valid for 1800 seconds)

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke production endpoint resources using access token (token -p1) listed in the production section.	User should be able to invoke production endpoint resources successfully.		
2	Invoke sandbox endpoint resources using access token (token-s1) listed in the sandbox section.	User should be able to invoke sandbox endpoint resources successfully.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-209: Check whether application tokens get successfully expired once validity time is exceeded. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Access token generated should get successfully expired when the validity period is being exceeded and should not be able to use for invoke api invocations.

Preconditions:

1. User should be accessfully logged into the system.

2. User should have run test case number 207 and 208 respectively.

3. User should reside on API store > my subscriptions page.

4. Gateway cache should be disabled.

5. <TimestampSkew> should be set to 0 in identity.xml

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
8	Invoke Production end point using (token -p1) once token gets expired due to validity period.	Invocation should get fialed due to invalid access token. (Response message: Unauthorized)		
9	Invoke Sandbox end point using (token -s1) once token gets expired due to validity period.	Invocation should get fialed due to invalid access token. (Response message: Unauthorized)		
10	Check database table IDN_OAUTH2_ACCESS_TOKEN of WSO2AM_DB after executing step 8 and 9.	In the database tables token_state of the tokens (p2 and s2) should be marked as EXPIRED.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	

Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-229: Check whether Application token revocation is successful when user regenerates the access token before expiry time. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully loged in user who has subscriber permission should be able to regenerate access token before eceeding validty time of the already generate token and once regenerated new token previous token should marked as revoked and clear from gateway caching as well.

Preconditions:
1. User should be accessfully logged into the system.

2. User should have run test case number 207.

3. User should reside on API store > my subscriptions page.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on regenerate button on 'key - Production' section before expiring tokens due to validity period, given new validity period as 600	New consumer key, consumer secret and access token (token - p2) should get generated.		
2	Click on regenerate button on 'key - Sandbox' section before expiring tokens due to validity period, given new validity period as 600.	New consumer key, consumer secret and access token (token - s2) should get genetated successfully.		
3	Check database table IDN_OAUTH2_ACCESS_TOKEN of WSO2AM_DB.	• TOKEN_STATE of previously generated tokens (p1 and s1) shold be marked as REVOKED. • New tokens should display correct information and TOKEN_STATE should be marked active in both records.		
4	Try to invoke Poduction endpoint using old application token (token - p1).	Invocation should get failed due to invalid access token.		
5	Try to invoke Sandbox endpoint using old application token (token s1).	Invocation should get failed due to invalid access token.		
6	Try to invoke Production end point using (token -p2) before the token gets expired.	User should be able to invoke production endpoint successfully.		
7	Try to invoke Sandbox end point using (token -s2), before the token gets expired after 600 seconds.	User should be able to invoke sandbox endpoint successfully.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-230: Check whether sandbox and production access tokens are generated once user gives different validity times. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user who has subscriber permission should be able to generate Access tokens given two different validity periods for production and sandbox.

Preconditions:
1. User should be successfully logged into API store

2. API (api1) should contain resources given auth type of the resources to application.

3. User should be subscribed to an API (api1) which has a production URL and a sandbox URL using an application (app1) which is throttling is set to unlimited

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Select application as app1 from the 'Application with subscription' list.	App1 should get selected successfully.		
2	Expand 'keys - production' section, give 1800 as validity time and click generate.	Consumer key, consumer secret and access token (token-p1) should generate successfully.		
3	Expand 'Keys - sandbox' , give 600 (seconds) as validity time and click 'generate' button.	Consumet key, consumer secret and a access token (token-s1) should get generated successfully.		
4	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated properly with generated access tokens and relavent validity times.	• Table IDN_OAUTH2_ACCESS_TOKEN should contain both sandbox and production access tokens in active state. • VALIDITY_PERIODs od both records should recorded in milliseconds correcly.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None

Last Result	**Not Run**

Test Case APIM-231: Check whether sandbox and production tokens get expired correctly when they are given 2 different validity periods. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

When tokens for sandbox and production are generated given different validity periods each token should get expired correctly based on the validity periods.

Preconditions:

1. user should be successfully logged into the api store.

2. Test case 230 should be executed.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke sandbox endpoint using generated token (token-s1) after 800 seconds (after exceeding validity time before exceeding validity time of production token.)	Invocation should fail.		
2	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated properly with generated access tokens and relavent validity times.	• Token generated for sandbox should be marked as EXPIRED • Token generated for production should be still in ACTIVE state.		
3	Regenerate sandbox token given validity time as 2700 seconds.	A token should generate successfully.		
4	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated properly with generated access tokens and relavent validity times.	• Newly generated token (sandbox) should be stored having correct validity period and other information.		
5	Invoke production endpoint using (token-p1) after exceeding validity time.	Invocation should fail.		
6	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated properly with generated access tokens and relavent validity times.	• Token generated for production (token - p1) should be marked as EXPIRED		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-242: Check whether User interface reflect changes done to validity time in identity.xml [Version : 1]

Author:	Sewmini Jayaweera

Summary:

When user changes default value of <ApplicationAccessTokenDefaultValidityPeriod> in <AM_HOME> /repository/conf/identity.xml file and restart the server api store > my subscription page should reflect the default validity time changed to the specified time.

Preconditions:

1. User who has subscriber permission should be logged into the API manager store.

2. User should reside on API store > my subscriptions page and selected an application.

3. There should be an application (app1) which has an api subscribed to it and generated access tokens already.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	change <ApplicationAccessTokenDefaultValidityPeriod> in <AM_HOME> /repository /conf/identity.xml to 4000 save it and restart the server.	App1's validity period and default app's validity period should have changed to 4000.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-243: Check whether changes made for ApplicationAccessTokenDefaultValidityPeriod in identity.xml applies correctly. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

When user regenerating access token having the default value (should be the value configured as ApplicationAccessTokenDefaultValidityPeriod in identity.xml) it should get successfully stored in the database as the validity perriod of the access token.

Preconditions:

1. Execute test case 242

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Check record belongs to generated token in database table IDN_OAUTH2_ACCESS_TOKEN in WSO2AM_DB database.	validity period of the token should have 4000000		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

2.4.2.Test Suite : password grant type

Test Artifact	
Mind Map	

Test Case APIM-241: Check whether validity period configure using identity.xml applies correctly. **[Version : 1]**

Author: Sewmini Jayaweera

Summary:
When user changes the default User Access Token Default Validity Period in identity.xml to a different value and generate an access token using password grant type the database record relavant to the access token should contain configured validity period.

Preconditions:

1. User (app owner) should have executed test cases 207 after editing api1 resource auth type to application user.

2. Generated access token (token - p1 and token -s1) should NOT be expired.

3. There should be a registered user with subscriber permission (user 2).

4. App owner should have combined the consumer key and consumer secret keys in the format **consumer-key:consumer-secret** and encode the combined string using base64 for both production and sandbox. base64 URL: http://base64encode.org

5. User should change User Access Token Default Validity Period as 180 seconds in <am_home>/repository/conf>identity.xml file.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	In the command line run bellow cURL command after replacing `Authorization: Basic with encoded string using production consmer key secret pair,` username and password using user2's user credentials. `curl -k -d "grant_type=password&username=<username>&password=<password>" -H "Authorization: Basic SVpzSWk2SERiQjVlOFZLZFpBblVpX2ZaM2Y4YTpHbTBiSjZvV1Y4ZkM1T1FMTGxDNmpzbEFDVzhh, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token`	User should get a json object having token_type, Bearer, expires_in, refresh_token and access_token.		
2	In the command line run bellow cURL command after replacing `Authorization: Basic with encoded string using sandbox consmer key secret pair,` username and password using user2's user credentials. `curl -k -d "grant_type=password&username=<username>&password=<password>" -H "Authorization: Basic SVpzSWk2SERiQjVlOFZLZFpBblVpX2ZaM2Y4YTpHbTBiSjZvV1Y4ZkM1T1FMTGxDNmpzbEFDVzhh, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token`	User should get a json object having token_type, Bearer, expires_in, refresh_token and access_token.		
5	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated with generated tokens.	• Tokens generated should be successfully updated and in active state. and the validity period column of the records should contain 1800000. • Tokens generated before changing the configuration should still contains the default validity time.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None

Last Result	**Not Run**

Test Case APIM-221: whether token api returns an error when invoking it with invalid user credentials. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	Check whether token API returns an error when user credentials used to invoke api are not registerd with api manager.
Preconditions:	1. There should be an API (api1) subscribed under an application (app1) 2. Application keys should be generated for both production and sandbox given validity time -1 3. App owner should have combined the consumer key and consumer secret keys in the format consumer-key:consumer-secret and encode the combined string using base64 for both production and sandbox. base64 URL: http://base64encode.org

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Replace username and password with rash and asdf1 respectively. `curl -k -d "grant_type=password&username=<username>&password=<password>" -H "Authorization: Basic <base64 encoded consumer-key:consumer-secret of PRODUCTION>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token`	Token API should return a json object response having an error saying invalid_grant. see example below. {"error":"invalid_grant","error_description":"Provided Authorization Grant is invalid."}		
2	Replace username and password with rash and asdf1 respectively. `curl -k -d "grant_type=password&username=<username>&password=<password>" -H "Authorization: Basic <base64 encoded consumer-key:consumer-secret of SANDBOX>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token`	Token API should return a json object response having an error saying invalid_grant. see example below. {"error":"invalid_grant","error_description":"Provided Authorization Grant is invalid."}		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-210: Check whether token api can be used to generate user tokens of type password grant successfully without specifying a scope. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	User who is successfully logged into API store and generated application tokens using store UI should be able to invoke token api and generate a user token successfully.
Preconditions:	1. User (app owner) should have executed test cases 207 after editing api1 resource auth type to application user. 2. Generated access token (token - p1 and token -s1) should NOT be expired. 3. There should be a registered user with subscriber permission (user 2). 4. App owner should have combined the consumer key and consumer secret keys in the format **consumer-key:consumer-secret** and encode the combined string using base64 for both production and sandbox. base64 URL: http://base64encode.org

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	In the command line run bellow cURL command after replacing `Authorization: Basic with encoded string using production consmer key secret pair,` username and password using user2's user credentials. `curl -k -d "grant_type=password&username=<username>&password=<password>" -H "Authorization: Basic SVpzSWk2SERiQjVlOFZLZFpBblVpX2ZaM2Y4YTpHbTBiSjZvV1Y4ZkM1T1FMTGxDNmpzbEFDVzhh, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token`	User should get a json object having token_type, Bearer, expires_in, refresh_token and access_token.		
2	In the command line run bellow cURL command after replacing `Authorization: Basic with encoded string using sandbox consmer key secret pair,` username and password using user2's user credentials. `curl -k -d "grant_type=password&username=<username>&password=<password>" -H "Authorization: Basic SVpzSWk2SERiQjVlOFZLZFpBblVpX2ZaM2Y4YTpHbTBiSjZvV1Y4ZkM1T1FMTGxDNmpzbEFDVzhh, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token`	User should get a json object having token_type, Bearer, expires_in, refresh_token and access_token.		

3	Invoke production endpoint using user access token generated for production.	User should be able to successfully invoke production endpoint.		
4	Invoke sandbox endpoint using user access token genearated for sandbox.	User should be able to successfully invoke sandbox endpoint		
5	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated with generated tokens.	Tokens generated should be successfully updated and in active state.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-211: Check whether user token get expired when validity period exceeds. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User should not be able to invoke API when the token exceeds the validity period of the user token.

Preconditions:

1. User should execute test case 210.

2. <TimestampSkew> should be set to 0 in identity.xml

3. server should be restarted after changing the configuration.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	After exceeding validity period (default is 3600) of the user token generated for production, invoke production endpoint using that user access token.	Invocatio should fail due to invalid access token.		
2	After exceeding validity period of the user token generated for sandbox, invoke sandbox endpoint using that user access token.	Invocation should fail.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-220: Check whether token api returns an error "invalid_grant" when it is invoke with a registered user who doesn't have subscriber permission. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check whether token API returns an error when user credentials used to invoke is not of a user who has subscriber permission.

Preconditions:

1. There should be a user (user1) registered with API manager who doesn't have subscriber permission.

2. There should be an API (api1) subscribed under an application (app1)

3. Application keys should be generated for both production and sandbox given validity time -1

4. App owner should have combined the consumer key and consumer secret keys in the format consumer-key:consumer-secret and encode the combined string using base64 for both production and sandbox. base64 URL: http://base64encode.org

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Replace username and password with user1's username and password `curl -k -d "grant_type=password&username=<username>&password=<password>" -H "Authorization: Basic <base64 encoded consumer-key:consumer-secret of PRODUCTION>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token`	Token API should return a json object response having an error saying invalid_grant. see example below. {"error":"invalid_grant","error_description":"Provided Authorization Grant is invalid."}		

2	Replace username and password with user1's username and password `curl -k -d "grant_type=password&username=<username>&password=<password>" -H "Authorization: Basic <base64 encoded consumer-key:consumer-secret of SANDBOX>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token`	Token API should return a json object response having an error saying invalid_grant. see example below. {"error":"invalid_grant","error_description":"Provided Authorization Grant is invalid."}		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-222: Check whether token api returns an error when user trying to invoke with an encoded string of a deleted application. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check whether token api returns an error when user invokes token api using base64 encoded consumer key:consumer secret of an application which is being deleted.

Preconditions:

1. There should be consumer key and consumer secret of an application which is deleted.

2. App owner should have combined the consumer key and consumer secret keys in the format consumer-key:consumer-secret and encode the combined string using base64 for both production and sandbox. base64 URL: http://base64encode.org

3. There should be a user who is successfully registered with the API store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Replace username and password with the registered user's crdentials and generated `base64 encoded consumer-key:consumer-secret of the below curl command and run it.` `curl -k -d "grant_type=password&username=<username>&password=<password>" -H "Authorization: Basic <base64 encoded consumer-key:consumer-secret of PRODUCTION>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token`	Token API should return a json object response having an error saying invalid_grant. see example below. {"error":"invalid_grant","error_description":"Provided Authorization Grant is invalid."}		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

2.4.3.Test Suite : Authorization code

Test Artifact	
Mind Map	

Test Case APIM-237: Check whether system removes first token when same user generates access token twise using same application's consumer key and secret. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

When same user genertes access token twise using same authorization code and application consumer key and consumer secret, first access token generated should get overriden in the database.

Preconditions:

1. 'Play ground' thired party application should be hosted in tomcat as explained in [1]

2. There should be an application (app1) created given ' http://localhost:8080/playground2.0/oauth2client ' as the callback URL.

3. User should be subscribed to an API (api 1) using application (app1) and generated consumer key, consumer secret and access token given validity time as -1.

4. User should reside on play ground application page.

[1] http://charithaka.blogspot.com/2013/07/oauth-20-grant-types-with-wso2-api.html

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-234: Check whether access token generated can no longer use once access permission of the application is denied for the user whose credentials were used to generate authorisation code **[Version : 1]**

Author:	Sewmini Jayaweera			
Summary:	Check whether access token generated can no longer use once access permission of the application is denied for the user whose credentials were used to generate authorisation code			
Preconditions:	1. Application admin should be logged into the system. 2. Test case 213 should be executed successfully.			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
4	Remove access permission from the user whose credentails were used when generating authorization code.	The user should no loger be able to access the application.		
5	Use Access token generated and try to access API.	API invocation should fail.		
6	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated properly with generated access token.			
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-224: Check whether Token API returns an error when it is invoked given an valid authorisation code generated for app2 with consumer key and secret of app1. **[Version : 1]**

Author:	Sewmini Jayaweera			
Summary:	System should return an error message when user gives a valid authorization code geneared for app2 with consumer key and consumer secret of app1.			
Preconditions:	1. 'Play ground' thired party application should be hosted in tomcat as explained in [1] 2. There should be an applications (app1, and app2) created given ' http://localhost:8080/playground2.0/oauth2client ' as the callback URL. 3. User should be subscribed to an API (api 1) using application (app1) and generated consumer key, consumer secret and access token given validity time as -1. 4. User should reside on play ground application page. [1] http://charithaka.blogspot.com/2013/07/oauth-20-grant-types-with-wso2-api.html			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on "Import Photos" icon.	User will be landed in a page where he will find a form with various options such as Authorization Grant Type, Client Id etc..		
2	• Select Authorization Code as the Authorization Grant Type • Copy the consumer key of app2 and enter it in Client Id text box. • Enter some name for scope • Enter http://localhost:8080/playground2.0/oauth2client (same as given when creating app1) as callback URL. • Enter as http://localhost:8280/authorize (when apim is not running on an offset) and finally Click on 'Authorize' button.	User should get redirected to the login page of the authorization server.		
3	Enter user credentials of a valid user who is registered with APIM store and click on 'sign in' button.	User should get redirected to a new form with an Authorization Code.		
4	Specify callback URL, same as specified in step 2, end point URL of the token api (https://localhost:8243/token), consumer secret generated for app1 and click on 'get access token'	User should get get an error message and no valid access token should get generated.		
5	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated.	No accesstoken should get stored relavant to this invocation.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				

Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-213: Check whether access token can be generated successfully using authorisation grant type. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

System should support authorization grant type in order to generate access tokens.

Preconditions:

1. execute test case 213

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated properly with generated access token.	• Generated access token should stored successfully in active state having correct information. • Validity period should get updated with default validity periods specified in identity.xml.		
2	Re-execute test case 213 and check IDN_OAUTH2_ACCESS_TOKEN table again.	record relavant to previously generated access token should getupdate with the new information.		
3	Try to invoke API using previously generated access token.	Invocation should fail.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-214: Check whether access token generated using authorisation grant type can be used to invoke API successfully. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User should be able to use access token and invoke API successfully.

Preconditions:

1. Execute test case 213 and obtain the access token generated.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Try to invoke api 1 using access token generated.	User should be able to invoke api 1 successfully.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-235: Check whether default validity period configure in identity.xml applies correctly when generating authorisation code. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Default validity period of the authorization code configured in identity.xml should be updated in the 'IDN_OAUTH2_AUTHORIZATION_CODE' table in database table of the WSO2AM_DB.

Preconditions:

1. 'Play ground' thired party application should be hosted in tomcat as explained in [1]

2. There should be an application (app1) created given ' http://localhost:8080/playground2.0/oauth2client ' as the callback URL.

3. User should be subscribed to an API (api 1) using application (app1) and generated consumer key, consumer secret and access token given validity time as -1.

4. User should reside on play ground application page.

5. <AuthorizationCodeDefaultValidityPeriod> should contain default value (300 s) in <product_home>/repository/conf/identity.xml.

[1] http://charithaka.blogspot.com/2013/07/oauth-20-grant-types-with-wso2-api.html

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:

1	Click on "Import Photos" icon.	User will be landed in a page where he will find a form with various options such as Authorization Grant Type, Client Id etc..		
2	• Select Authorization Code as the Authorization Grant Type • Copy the consumer key of app1 and enter it in Client Id text box. • Enter some name for scope • Enter http://localhost:8080/playground2.0/oauth2client (same as given when creating app1) as callback URL. • Enter as http://localhost:8280/authorize (when apim is not running on an offset) and finally Click on 'Authorize' button.	User should get redirected to the login page of the authorization server.		
3	Enter user credentials of a valid user who is registered with APIM and click on 'sign in' button.	User should get redirected to a new form with an Authorization Code.		
4	Check whether IDN_OAUTH2_AUTHORIZATION_CODE table of the WSO2AM_DB is being updated with the generated authorization code.	'IDN_OAUTH2_AUTHORIZATION_CODE' table should be updated having correct details. Note : Validity period should be equivalent to the configuered <AuthorizationCodeDefaultValidityPeriod> time. The authorization codes generated before changing the default validity period should contains the default validity period.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-236: Check whether system gives an error when trying generate access token using expired authorisation code. [Version : 1]

Author:	Sewmini Jayaweera			
Summary:	System should give an error message saying 'invalid_grant' or similar when user tries to get an access token using expired authorisation code.			
Preconditions:	1. execute 235.			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Specify callback URL, same as specified in app1, end point URL of the token api (https://localhost:8243/token), consumer secret generated for app1 and click on 'get access token'	User should get an error message saying invalid grant or similar.		
2	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table is not updated due to this token api invocation.	No new record should be there.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

2.4.4.Test Suite : SAML extension grant type

Test Artifact
Mind Map

Test Case APIM-228: Check whether user gets an error when invoking token api with a SAML2 assertion generated for a user who doesn't have subscribe permission. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	User should get an json object with an error message when trying to generate auth2 token sending a SAML2 token generated using a user who doesn't have subscribe permission.
Preconditions:	1. There should be a registered user (user1) in api manager who doesn't have subscriber permission. 2. User must have successfully subscribed to an API (api1) and generated valid consumer key, consumer secret through API store > my subscription page. 3. Start API manager without port offsets. 4. Give port offset of IS as 1 and start the server. 5. Refer to [1] and do the relevant configurations in api manager, IS nad generate SAML2 token for user1.

6. API manager and IS user store and permission db should be shared.

[1] https://docs.wso2.com/display/AM180/Token+API#TokenAPI-ExchangingSAML2bearertokenswithOAuth2(SAMLextensiongranttype)

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
2	execute following command after replacing saml2 token and base 64 encoded consumer key:consumer secret. curl -k -d "grant_type=urn:ietf:params:oauth:grant-type:saml2-bearer&assertion= <SAML2_Encoded_Assertion_Token>" -H "Authorization: Basic <consumer key:consumer secret encoded using base 64> Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token	Token API should return an error.		
3	Check database table IDN_OAUTH2_ACCESS_TOKEN of WSO2AM_DB	No token should get saved for this invocation.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-226: Check whether user gets an error message once token API is invoked with a invalid consumer key:consumer secret pair. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User should get a json massage with an error message, when user invokes token API given an invalid authorization header.

Preconditions:

1. There should be a registered user in api store.

2. User must have successfully subscribed to an API (api1) and generated valid consumer key, consumer secret through API store > my subscription page.

3. Start API manager without port offsets.

4. Give port offset of IS as 1 and start the server.

5. Refer to [1] and do the relevant configurations and get a valid SAML token generated.

6. API manager and IS user store should be shared.

[1] https://docs.wso2.com/display/AM180/Token+API#TokenAPI-ExchangingSAML2bearertokenswithOAuth2(SAMLextensiongranttype)

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
2	execute following command after replacing samal token and in valid authorization header. curl -k -d "grant_type=urn:ietf:params:oauth:grant-type:saml2-bearer&assertion= <SAML2_Encoded_Assertion_Token>" -H "Authorization: Basic <consumer key:consumer secret encoded using base 64> Content-Type: application/x-www-form-urlencoded" https://localhost:8243 /token	Token API should return an error.		
3	Check database table IDN_OAUTH2_ACCESS_TOKEN of WSO2AM_DB	No new token should get stored relavant to this invocation.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-218: Check whether system successfully return OAuth2.0 access token when a valid SAML2 bearer token is being passed. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Enterprise application which use SAML 2 sso for authorization should be able to pass SAML 2 bearer token and get an OAuth2.0 access token which can be used for API invocations.

Preconditions:

1. There should be a registered user in api store.

2. User must have successfully subscribed to an API (api1) and generated valid consumer key, consumer secret through API store > my subscription page.

3. Start API manager without port offsets.

4. Give port offset of IS as 1 and start the server.

5. Refer to [1] and do the relevant configurations and get the SAML token generated.

6. API manager and IS user store should be shared.

[1] https://docs.wso2.com/display/AM180/Token+API#TokenAPI-ExchangingSAML2bearertokenswithOAuth2(SAMLextensiongranttype)

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Combine the consumer key and consumer secret keys as `consumer-key:consumer-secret`. Encode the combined string using base64 (http://base64encode.org)	User should be able to get the encoded string.		
2	Use below curl command to generate auth2.0 access token url -k -d "grant_type=urn:ietf:params:oauth:grant-type:saml2-bearer&assertion= <SAML2_Encoded_Assertion_Token>" -H "Authorization: Basic <base64 encoded keys>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token	Auth 2.0 access token should be generated .		
3	Use the generated token and invoke api1	user should be able to invoke api1		
4	Check database table IDN_OAUTH2_ACCESS_TOKEN of WSO2AM_DB	Access token generated should be saved and token state should be active.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-225: Check whether user gets an error message once token API is invoked with a invalid "SAML2_Encoded_Assertion_Token". [Version : 1]

Author: Sewmini Jayaweera

Summary:

User should get a json massage with an error message, when user invokes token API end point given an invalid SAML assertion token and valid application consumer key and consumer secret.

Preconditions:

1. There should be a registered user in api store.

2. User must have successfully subscribed to an API (api1) and generated valid consumer key, consumer secret through API store > my subscription page.

3. Start API manager without port offsets.

4. Give port offset of IS as 1 and start the server.

5. Refer to [1] and do the relevant configurations in api manager and IS download java client.

6. API manager and IS user store and permission db should be shared.

[1] https://docs.wso2.com/display/AM180/Token+API#TokenAPI-ExchangingSAML2bearertokenswithOAuth2(SAMLextensiongranttype)

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Go to cilent folder and execute given command after replacing relavant fields. when providing a user name use an invalid user name in order to generate invalid saml assertion. java -jar SAML2AssertionCreator.jar TestSP <registered username> <saml2_assertion_recipient> <saml2_asseertion_audience_restriction> <full_path_to_your_JKS_file> <your_JKS_password> <your_certificate_alias> <your_private_key_password>	An SAML2 encoded assertion token should be generated.		
2	execute following command after replacing samal token and encoded consumer key:consumer secret in order to get auth 2 token. curl -k -d "grant_type=urn:ietf:params:oauth:grant-type:saml2-bearer&assertion= <SAML2_Encoded_Assertion_Token>" -H "Authorization: Basic <consumer key:consumer secret encoded using base 64> Content-Type: application/x-www-form-urlencoded" https://localhost:8243 /token	Token API should return an error.		
3	Check database table IDN_OAUTH2_ACCESS_TOKEN of WSO2AM_DB	No token should get saved for this invocation.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-238: Check whether validity time configure using identity.xml get applied to the token generated using SAML grant type correctly. [Version : 1]

Author: Sewmini Jayaweera

Summary:

When user changes the default value of <UserAccessTokenDefaultValidityPeriod> in identity.xml when access token is generated using SAML2 bearer grant type, correct validity period should be shown in the database table 'IDN_OAUTH2_ACCESS_TOKEN' .

Preconditions:

1. There should be a registered user in api store.

2. User must have successfully subscribed to an API (api1) and generated valid consumer key, consumer secret through API store > my subscription page.

3. Start API manager without port offsets.

4. Give port offset of IS as 1 and start the server.

5. Refer to [1] and do the relevant configurations and get the SAML token generated.

6. API manager and IS user store should be shared.

7. Change <UserAccessTokenDefaultValidityPeriod> to 1800 in <apim_home> /repository/conf/identity.xml file save and restart apim.

[1] https://docs.wso2.com/display/AM180/Token+API#TokenAPI-ExchangingSAML2bearertokenswithOAuth2(SAMLextensiongranttype)

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Combine the consumer key and consumer secret keys as `consumer-key:consumer-secret`. Encode the combined string using base64 (http://base64encode.org)	User should be able to get the encoded string.		
2	Use below curl command to generate auth2.0 access token url -k -d "grant_type=urn:ietf:params:oauth:grant-type:saml2-bearer&assertion=<SAML2_Encoded_Assertion_Token>" -H "Authorization: Basic <base64 encoded keys>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token	Auth 2.0 access token should be generated .		
3	Check database table IDN_OAUTH2_ACCESS_TOKEN of WSO2AM_DB to see generated access token is there with valid information.	The access token should be having token state as active and validity period as 1800000 miliseconds. Access tokens which were generated prior to changing the time should contain default validity period which was configured in identity.xml		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-239: Check whether access token generated using saml2 bearer token grant type expires when exceeding the given validity time. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Once access token exceeds given validity time token should get expired and no invocation should be able to do using the particualar token.

Preconditions:

1. Execute test case 238

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Wait until validity time exceeds (1800 seconds) and then invoke api1 using the access token.	Token invocation should fail.		
2	Check table IDN_OAUTH2_ACCESS_TOKEN of WSO2AM_DB.	The token status should be marked as EXPIRED.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

2.4.5.Test Suite : renewing access token

Test Artifact
Mind Map

Test Case APIM-212: Check whether user can successfully renew token using refresh token once the user token is expired. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User should be able to renew access token by invoking token api with the refresh token.

Preconditions:

1. User must have executed test case 211.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Make an api call to token api given grant type as `refresh_token` and the refresh token generated when generating user token in test case 210. `curl -k -d "grant_type=refresh_token&refresh_token=<retoken>&scope=PRODUCTION" -H "Authorization: Basic <base64 encoded string>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token`	A jason resoponse should be returned having new access token and refresh token with other information.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-223: Check whether API store > my subscription page can be use to regenerate access token. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User who is successfully logged into the API store should be able to regenerate application token by clicking on the regenerate button on the API store > my subscriptions page.

Preconditions:

1. There should be an API (api1) subscribed under an application (app1) which caontains resources given application auth type.

2. Application keys and tokens should be generated for both production and sandbox (tokenp1 and token s1 respectively) given validity time -1

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Use access token generated (token p1) for production and invoke production end point.	User should be able to invoke production end point successfully.		
2	Use access token generated for sandbox (token s1) and invoke sandbox end point.	User should be able to invoke sandbox endpoint successfully.		
3	Go to API store > my subscriptions > production and click on 'regenerate' button.	A new access token (token p2) should get generated while consumer key and concumer secret remains the same.		
4	Go to API store > my subscriptions > sandbox and click on 'regenerate' button.	Anew access token (token s2) should get generated while consumer key and consumer secret remails the same.		
5	Use new access token generated for production (token p2) and invoke production endpoint.	User should be able to successfully invoke production endpoint.		
6	Use old access token generated for production (token p1) and invoke production endpoint.	Invocation should fail.		
7	Use new access token generated for sandbox (token s2) and invoke sandbox endpoint.	User should be able to invoke sandbox endpoint successfully.		
8	use old access token generated for sandbox (token s1) and invoke sandbox endpoint.	Invocation should fail.		
9	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated.	• token state of token-p1 and token-s1 should be marked as revoked. • token p2 and tokens2 should be marked as active.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

2.4.6.Test Suite : revoking access token

Test Artifact
Mind Map

Test Case APIM-219: Check whether user can successfully make the generated token invalid by invoking revoke end point. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User should be able to revoke access token bu invoking revoke ed point using access token which needs to be revoked.

Preconditions:

1. There should be a valid access token generated for a particular application.

2. Application must have an API (api1) which can be invoked using the particular access token (token1).

3. API Gateway cache should be disabled.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Run following cURL command on the terminal in order to invoke revoke end point. `curl -k -d "token=<ACCESS_TOKEN_TO_BE_REVOKED>" -H "Authorization: Basic <Base64Encoded(Consumer key:consumersecret)>" http://localhost:8280/revoke`	End point should get revoked.		
2	Try to invoke api1 using access token1	User should not be able to invoke api1		
3	Check whether 'IDN_OAUTH2_ACCESS_TOKEN' table of 'WSO2AM_DB', is being updated	Status of the accesstoken should be marked as REVOKED.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

2.4.7.Test Suite : Implicit grant type

Test Artifact	
Mind Map	

Test Case APIM-216: Check whether access token generated using implicit authorisation grant type can be used to invoke API successfully [Version : 1]

Author:	Sewmini Jayaweera			
Summary:	User should be able to use access token generated using implicit grant type and invoke API successfully.			
Preconditions:	1. Execute test case 215 and obtain the access token generated.			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Try to invoke api 1 using access token generated.	User should be able to invoke api 1 successfully.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-215: Check whether access token can be generated successfully using Implicit grant type. [Version : 1]

Author:	Sewmini Jayaweera			
Summary:	System should support **Implicit** grant type in order to generate access tokens.			
Preconditions:	1. 'Play ground' thired party application should be hosted in tomcat as explained in [1] 2. There should be an application (app1) created given ' http://localhost:8080/playground2.0/oauth2client ' as the callback URL. 3. User should be subscribed to an API (api 1) using application (app1) and generated consumer key, consumer secret and access token given validity time as -1. 4. User should reside on play ground application page. [1] http://charithaka.blogspot.com/2013/07/oauth-20-grant-types-with-wso2-api.html			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on "Import Photos" icon.	User will be landed in a page where he will find a form with various options such as Authorization Grant Type, Client Id etc..		

2	• Select implicit as the Authorization Grant Type • Copy the consumer key of app1 and enter it in Client Id text box. • Enter some name for scope • Enter http://localhost:8080/playground2.0/oauth2client (same as given when creating app1) as callback URL. • Enter as http://localhost:8280/authorize (when apim is not running on an offset) as authorisation end point and finally Click on 'Authorize' button.	User should get redirected to the login page of the authorization server.		
3	Enter user credentials of a valid user who is registered with APIM and click on 'sign in' button.	User should get redirected to a page which has the access token.		
4	Check table IDN_OAUTH2_ACCESS_TOKEN of WSO2AM_DB contains generated access token.	The access token should get stored in the database having active as the token status and default validity period configured in identity.xml		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

2.4.8.Test Suite : Other

Test Artifact
Mind Map

Test Case APIM-244: Check whether validity period configured using identity.xml applies correctly when regenerating new [Version : 1]	
Author:	Sewmini Jayaweera
Summary: When user have configured different validity periods for AccessTokenDefaultValidityPeriod and ApplicationAccessTokenDefaultValidityPeriod in identity.xml all the new tokens generating after the change should reflect changes correctly.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

2.4.9.Test Suite : Generate application token for scopes

This includes the test cases written for token generation for a paticular scope and invoking them

Test Artifact
Mind Map

2.4.9.1.Test Suite : Token generation for scopes

Test Artifact
Mind Map

Test Case APIM-407: A user should be able to generate a token via CURL for a paticular scope of an API when there are multiple scopes assigned to an API via CURL [Version : 1]				
Author:	Ushani Balasooriya			
Summary: A user should be able to generate a token via CURL for a paticular scope of an API when there are multiple scopes assigned to an API via CURL ***Special Note :*** ***This test should be executed for a fresh pack login to store by a different role with admin priveledges and for a user with only subscriber priviledges.***				
Preconditions: 1. A user should be logged in to publisher. 2. Create an api and assign a scope which is bound to a paticular role. E.g., scope 1 - role 1 for GET resource of API1 and scope 2 - role2 for POST 3. API should have been published 4. User assigned to role1 should be logged in to the store 5. Subscribe API1 to an application (E.g., app1)				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:

1	Send the below command via curl curl -k -d "grant_type=password&username=<username>&password=<password>&scope=<scope1>" -H "Authorization: Basic <token>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token	Token should be generated for the paticular scope. E.g., Only for the scope1 since scope 2 is defined for another role2.		
2	Send the same command by login user (role1) for all the scopes defined for the paticular API curl -k -d "grant_type=password&username=<username>&password=<password>&scope=<scope1> <scope2>" -H "Authorization: Basic <token>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token	Then token should be generated for scope1 and not the scope 2.		
3	Send the same command by another user (role2) belongs to role2. curl -k -d "grant_type=password&username=<username>&password=<password>&scope=<scope1> <scope2>" -H "Authorization: Basic <token>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token	Then token should be generated for the scope2 and not the scope1		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-405: A user should be able to assign a scope to a newly creating API and subscribe by an application and generate a access token to the paticular scope [Version : 1]

Author:	Ushani Balasooriya

Summary:

A user should be able to assign a scope to a newly creating API and subscribe by an application and generate a access token to the paticular scope

Preconditions:

1. A user should be logged in to publisher.

2. Create an api and assign a scope which is bound to a paticular role. E.g., scope 1 - role 1 for GET resource of API1

3. API should have been published

4. User assigned to role1 should be logged in to the store

5. Subscribe API1 to an application (E.g., app1)

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Select scope from the popup (E.g., scope1)	scope 1 should be ticked		
2	Click on the regenerate button	A new access token should be generated and scope 1 should be displayed in the key section for the paticular environment (Production or sandbox)		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-364: A user should be able to generate a token via CURL for a paticular scope of an API [Version : 1]

Author:	Ushani Balasooriya

Summary:

A sucessfully logged in user should be able to assign a scope to a subscribed application and generate a token via CURL

Preconditions:

1. A user should be logged in to publisher.

2. Create an api and assign a scope which is bound to a paticular role. E.g., scope 1 - role 1 for GET resource of API1

3. API should have been published

4. User assigned to role1 should be logged in to the store

5. Subscribe API1 to an application (E.g., app1)

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send the below command via curl curl -k -d "grant_type=password&username=<username>&password=<password>&scope=<scope1>" -H "Authorization: Basic <token>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token	Token should be generated for the paticular scope.		
2	Send the same command by another login user who is assigned to a role which is not assigned to the paticular scope. E.g., user from role2	Then token should be generated for default scope not the scope defined for the step1 scope.		

	curl -k -d "grant_type=password&username=<username>&password=<password>& scope=<scope1>" -H "Authorization: Basic <token>, Content-Type: application/x-www-form-urlencoded" https://localhost:8243/token			
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-362: A user should be able to assign a scope to a subscribed application which the key is already generated and generate the token again for a scope [Version : 1]

Author:	Ushani Balasooriya			
Summary:				
A sucessfully logged in user should be able to assign a scope to a subscribed application which the key is already generated and once the scope is assigned, generate the token again				
Preconditions:				
1. A user should be logged in to publisher. 2. Create an api and publish it 3. API should have been published 4. User assigned to role1 should be logged in to the store 5. Create an application (app1) and generate the keys 6. Then Subscribe API1 to that application (E.g., app1)				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Edit the API1 and assign a scope which is bound to a paticular role. E.g., scope 1 - role 1 for GET resource of API1	scope1 shouls be assigned to GET resource		
2	Login to store and select scope from the popup (E.g., scope1)	scope 1 should be ticked		
3	Click on the regenerate button	A new access token should be generated and scope 1 should be displayed in the key section for the paticular environment (Production or sandbox)		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-361: A user should be able to assign multiple scopes to a subscribed application and generate token [Version : 1]

Author:	Ushani Balasooriya			
Summary:				
A sucessfully logged in user should be able to assign multiple scopes to an subscribed application				
Preconditions:				
1. A user should be logged in to publisher. 2. An API should have been created and assigned a scope which is bound to a paticular role. E.g., scope 1 - role 1 for GET resource of API1, scope 2 - role 1 for POST 3. API should have been published 4. User assigned to role1 should be logged in to the store 5. User is ubscribed to API1 to an application (E.g., app1)				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Select multiple scopes from the popup (E.g., scope1 and scope2)	scope 1 and scope 2 should be ticked		
2	Click on the regenerate button	A new access token should be generated and scope 1 and scope 2 should be displayed in the key section for the paticular environment (Production or sandbox)		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			

Keywords:	None
Last Result	**Not Run**

Test Case APIM-360: A user should be able to assign a scope to a subscribed application and generate a new access token to a paticular scope [Version : 1]				
Author:	Ushani Balasooriya			
Summary:				
A sucessfully logged in user should be able to assign a scope to a subscribed application and generate a new access token to a paticular scope				
Preconditions:				
1. A user should be logged in to publisher. 2. Create an api and assign a scope which is bound to a paticular role. E.g., scope 1 - role 1 for GET resource of API1 3. API should have been published 4. User assigned to role1 should be logged in to the store 5. Subscribe API1 to an application (E.g., app1) 6. A token should be already generated without a scope selected (Without selecting a scope)				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Select scope from the popup (E.g., scope1)	scope 1 should be ticked		
2	Click on the regenerate button	A new access token should be generated and scope 1 should be displayed in the key section for the paticular environment (Production or sandbox)		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

2.5.Test Suite : API visibility and subscriptions

Test Artifact	
Mind Map	

Test Case APIM-68: API with subscription available only to current tenant, not allow other tenant users to subscribe. [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Create and publish an API given subscription subscription available only to current tenant. Users of other tenant domains should not be able to subscribe to the API.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-66: Publish an API given 'visible to my domain' -users belong to same tenant should be able to view. [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Create an API from a tenant and restrict it to the tenant doman and verify that users of the same tenant can view created API.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-60: Published APIs with public visibility can be viewed by all users. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	
User who has creator and publisher permission publishes an API given public visibility and all users (registered / anonymous / users belogns to other tenants) should be able to view it.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-61: publish an API given 'visible to my domain' -users belong to other tenantsshould not be able to view [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Create an API from a tenant and restrict it to the tenant doman and verify that it is not visible to other tenants	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-62: published api with visibility restricted to a role should visible to users with the particular role. [Version : 1]

Author:	Sewmini Jayaweera			
Summary:				
Create an API with visibility restricted to a role and verify that it is only visible to users with the same role within that tenant who do not have publish permission.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Create an API with visibility restricted to a role and check whether users who are assigned with the particular role can view the API	users who are assigned with the particular role should be able to view the API		
2	Check whether users who are not assigned with the particular role cannot view the API.	Users who are not assigned with the particular role should not be able to view the API.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-63: publishes APIs with visibility restricted to a role should be visible to user with publisher role. [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Create an API with visibility restricted to a role and verify that it is visible to any user with publish permission within that tenant	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-64: API with visibility restricted to a role should not be visible to users belong to other tenants [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Create an API with visibility restricted to a role and verify that it's not visible to users with the same role belonging to other tenants	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-65: publish an api with visibility restricted to multiple roles-only users with allowed roles can view. [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Create an API restricted to multiple visibility roles and verify that its visibility is restricted users with the allowed roles within the tenant	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-67: API with subscription available only to current tenant, allows current tenant user to subscribe. [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Create and publish an API given subscription subscription available only to current tenant. Users of the current tenant domain should be able to subscribe to the API.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-71: APIs with subscription available to all tenants should allow users in other tenants to subscribe. [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Users who have subscribe permission in any tenant domain should be able to get subscribed into a Published API given subacription available to all tenants.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-72: API with subscription available to specific tenants should allow subscription to specified tenants [Version : 1]	
Author:	Sewmini Jayaweera
Summary:	
Create an API with public visibility and subscription restricted to selected tenants. Subscription should only be allowed for selected tenants	
Execution type:	Manual
Estimated exec. duration (min):	

Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

2.6.Test Suite : Application token

Test Artifact
Mind Map

2.7.Test Suite : Search functionality

Test Artifact
Mind Map

2.7.1.Test Suite : publisher

Test Artifact
Mind Map

Test Case APIM-350: User Search added API by name [Version : 1]

Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API by name on publisher				
Preconditions:				
1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API name on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-351: User Search added API by provider name [Version : 1]

Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API by provide name on publisher				
Preconditions:				
1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API provider name (ex:provider:user1) on search text field on browser page and press search button	Only seacrh API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-352: User Search added API by api version [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API by api version on publisher				
Preconditions:				
1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API version of the API (ex: version:1.0.0) on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-353: User should be able to search added API by context [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API by context on publisher				
Preconditions:				
1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API context (ex:context:api1) on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-354: User search for API with upper case name [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API with upper case name on publisher				
Preconditions:				
1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API with upper case name on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-355: User search for API with lower case name [Version : 1]				
Author:	Shamin Goonetileke			
Summary: Successfully logged in user should be able to search an API with lower case name on publisher				
Preconditions: 1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API with lower case name on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-356: Successfully logged in user should be able search API by status [Version : 1]				
Author:	Shamin Goonetileke			
Summary: Successfully logged in user should be able to search an API by API status				
Preconditions: 1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API context (ex: status:created) on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-357: Successfully logged in user should be able search API by description [Version : 1]				
Author:	Shamin Goonetileke			
Summary: Successfully logged in user should be able to search an API by API description				
Preconditions: 1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API context (ex: description:xxxx) on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			

Last Result	**Not Run**

Test Case APIM-358: Successfully logged in user should be able search API by subcontext [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API by API subcontext				
Preconditions:				
1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API context (ex: subcontext:xxx) on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-359: Successfully logged in user should be able search API by documentation content [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able search API by documentation content				
Preconditions:				
1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API context (ex: doc:xxxx) on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-367: Successfully logged in user should be able search with keywords that contains only part of the provider name [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able search with keywords that contains only part of the provider name.				
Preconditions:				
1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API provider name (ex:provider:us) on search text field on browser page and press search button	Only seacrh API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				

Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-368: Successfully logged in user should be able search with keywords that contains only part of the conte [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able search with keywords that contains only part of the context name				
Preconditions:				
1) User should be successfully logged into the publisher page. 2) There should be API's already created. 3) User should be on API browser page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API provider name (ex: context:ap) on search text field on browser page and press search button	Only seacrh API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-384: User should be able to see all API's in all stage on browse page [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to view all api's created (ex: created, published)				
Preconditions:				
1) User should be successfully logged into the publisher page. 2) There should be API's already created.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User clicks on browse button	User should be able to see all API's with Thubnail Api name Lifecycle state User count Version number Api creator Delete icon button		
2				
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

2.7.2.Test Suite : Store

Test Artifact	
Mind Map	

Test Case APIM-404: (Tenant specific) User Search added API by api version [Version : 1]	
Author:	Shamin Goonetileke
Summary:	
Successfully logged in user should be able to search an API by api version on store	

Preconditions:

1) User should be successfully logged into the store.

2) There should be API's already published.

3) User should be on API store home page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API version of the API (ex: version:1.0.0) on search text field on store page and press search button	Only searched API's should be able to view on store		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-403: (Tenant specific) User Search added API by name [Version : 1]

Author:	Shamin Goonetileke

Summary:

Successfully logged in user should be able to search an API by name on publisher

Preconditions:

1) User should be successfully logged into the store.

2) There should be API's already published.

3) User should be on API store home page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API name on search text field on store page and press search button	Only searched API's should be able to view on store		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-402: (Tenant specific) User should be able to search added API by context [Version : 1]

Author:	Shamin Goonetileke

Summary:

Successfully logged in user should be able to search an API by context on store

Preconditions:

1) User should be successfully logged into the store.

2) There should be API's already published.

3) User should be on API store home page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API context (ex:context:api1) on search text field on browser page and press search button	Only searched API's should be able to view on store		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-401: (Tenant specific) User search for API with upper case name [Version : 1]

Author:	Shamin Goonetileke

Summary:				
Successfully logged in user should be able to search an API with upper case name on store				
Preconditions:				
1) User should be successfully logged into the store. 2) There should be API's already published. 3) User should be on API store home page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API with upper case name on search text field on browser page and press search button	Only searched API's should be able to view on store		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-400: (Tenant specific) User search for API with lower case name [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API with lower case name on store				
Preconditions:				
1) User should be successfully logged into the store. 2) There should be API's already published. 3) User should be on API store home page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API with lower case name on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-399: (Tenant specific) Successfully logged in user should be able search API by description [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API by API description				
Preconditions:				
1) User should be successfully logged into the store. 2) There should be API's already published. 3) User should be on API store home page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API context (ex: description:xxxx) on search text field on store page and press search button	Only searched API's should be able to view on store		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-398: (Tenant specific) Successfully logged in user should be able search with keywords that contains only part of the provider name [Version : 1]

Author:	Shamin Goonetileke

Summary:

Successfully logged in user should be able search with keywords that contains only part of the provider name.

Preconditions:

1) User should be successfully logged into the store.

2) There should be API's already published.

3) User should be on API store home page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API provider name (ex:provider:us) on search text field on browser page and press search button	Only seacrh API's should be able to view on store		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-393: (Tenant specific)Successfully logged in user should be able search with keywords that contains only part of the context name [Version : 1]

Author:	Shamin Goonetileke

Summary:

Successfully logged in user should be able search with keywords that contains only part of the context name

Preconditions:

1) User should be successfully logged into the store.

2) There should be API's already published.

3) User should be on API store home page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API provider name (ex: context:ap) on search text field on browser page and press search button	Only seacrh API's should be able to view on browser page		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-392: Successfully logged in user should be able search with keywords that contains only part of the context name [Version : 1]

Author:	Shamin Goonetileke

Summary:

Successfully logged in user should be able search with keywords that contains only part of the context name

Preconditions:

1) User should be successfully logged into the store.

2) There should be API's already published.

3) User should be on API store home page

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API provider name (ex: context:ap) on search text field on browser page and press search button	Only seacrh API's should be able to view on browser page		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-391: Successfully logged in user should be able search with keywords that contains only part of the provider name [Version : 1]				
Author:	Shamin Goonetileke			
Summary: Successfully logged in user should be able search with keywords that contains only part of the provider name.				
Preconditions: 1) User should be successfully logged into the store. 2) There should be API's already published. 3) User should be on API store home page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API provider name (ex:provider:us) on search text field on browser page and press search button	Only seacrh API's should be able to view on store		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-390: Successfully logged in user should be able search API by description [Version : 1]				
Author:	Shamin Goonetileke			
Summary: Successfully logged in user should be able to search an API by API description				
Preconditions: 1) User should be successfully logged into the store. 2) There should be API's already published. 3) User should be on API store home page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API context (ex: description:xxxx) on search text field on store page and press search button	Only searched API's should be able to view on store		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-389: User search for API with lower case name [Version : 1]				
Author:	Shamin Goonetileke			
Summary: Successfully logged in user should be able to search an API with lower case name on store				
Preconditions: 1) User should be successfully logged into the store. 2) There should be API's already published. 3) User should be on API store home page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API with lower case name on search text field on browser page and press search button	Only searched API's should be able to view on browser page		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			

Keywords:	None
Last Result	**Not Run**

Test Case APIM-388: User search for API with upper case name [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API with upper case name on store				
Preconditions:				
1) User should be successfully logged into the store. 2) There should be API's already published. 3) User should be on API store home page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API with upper case name on search text field on browser page and press search button	Only searched API's should be able to view on store		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-387: User should be able to search added API by context [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API by context on store				
Preconditions:				
1) User should be successfully logged into the store. 2) There should be API's already published. 3) User should be on API store home page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API context (ex:context:api1) on search text field on browser page and press search button	Only searched API's should be able to view on store		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-386: User Search added API by name [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API by name on publisher				
Preconditions:				
1) User should be successfully logged into the store. 2) There should be API's already published. 3) User should be on API store home page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API name on search text field on store page and press search button	Only searched API's should be able to view on store		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				

Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-385: User Search added API by api version [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
Successfully logged in user should be able to search an API by api version on store				
Preconditions:				
1) User should be successfully logged into the store. 2) There should be API's already published. 3) User should be on API store home page				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User type the API version of the API (ex: version:1.0.0) on search text field on store page and press search button	Only searched API's should be able to view on store		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

2.8.Test Suite : API Docs

Test Artifact
Mind Map

2.8.1.Test Suite : documentation visibility

Test Artifact
Mind Map

Test Case APIM-111: Check document visibility when it's given as 'Visible to my domain'. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Only users who are registered in API tenant domain should be able to see the documents having visibility as 'Visible to my domain'.				
Preconditions: 1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an published api. 3. User who has permission to add documents should reside on docs page of the particular api. 4. <EnableAPIDocVisibilityLevels> should be set to true in <AM_HOME>/repository/conf/api-manager.xml				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	Add new document section should expand and visibility drop down also should be available.		
2	Add a 'How to' type in-line document given visibility as 'visible to my domain' and check whether only the users who are registered in API tenant domain can view the added document.	Only the users who are registered in API tenant domain should be able to view the document.		
3	Add a 'Samples & SDK' type document with a file given visibility as 'visible to my domain' and check whether only the users who are registered in API tenant domain can view the added document.	Only the users who are registered in API tenant domain should be able to access the added document.		
4	Add a 'Public forum' type document with a URL given visibility as 'visible to my domain' and check whether only users who are registered in API tenant domain can access the document.	Only the users who are registered in API tenant domain should be able to access the added document.		
5	Add a 'Support forum' type document with a URL given visibility as 'visible to my domain' and check whether only the users who are registered in API tenant domain can access the document.	Only the users who are registered in API tenant domain should be able to access the added document.		
6	Add a other type in-line document given visibility as 'visible to my domain' and check whether only the User users who are registered in API tenant domain can access the document.	Only the users who are registered in API tenant domain should be able to access the added document.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			

Keywords:	None
Last Result	**Not Run**

Test Case APIM-110: Check document visibility when it's given as private. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Only User who has permission to log in to the API publisher and create and/or publish APIs should be able to see the documents having visibility as 'Private'.				
Preconditions: 1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an published api. 3. User who has permission to add documents should reside on docs page of the particular api. 4. <EnableAPIDocVisibilityLevels> should be set to true in <AM_HOME>/repository/conf/api-manager.xml				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	Add new document section should expand and visibility drop down also should be available.		
2	Add a 'How to' type in-line document given visibility as private and check whether only the User who has permission to log in to the API publisher and create and/or publish APIs can view the added document.	Only the User who has permission to log in to the API publisher and create and/or publish APIs should be able to view the document.		
3	Add a 'Samples & SDK' type document with a file given visibility as private and check whether only the User who has permission to log in to the API publisher and create and/or publish APIs can view the added document.	Only the User who has permission to log in to the API publisher and create and/or publish APIs should be able to access the added document.		
4	Add a 'Public forum' type document with a URL given visibility as 'private and check whether only User who has permission to log in to the API publisher and create and/or publish APIs can access the document.	Only the User who has permission to log in to the API publisher and create and/or publish APIs should be able to access the added document.		
5	Add a 'Support forum' type document with a URL given visibility as 'private' and check whether only the User who has permission to log in to the API publisher and create and/or publish APIs can access the document.	Only the User who has permission to log in to the API publisher and create and/or publish APIs should be able to access the added document.		
6	Add a other type in-line document given visibility as 'private' and check whether only the User who has permission to log in to the API publisher and create and/or publish APIs can access the document.	Only the User who has permission to log in to the API publisher and create and/or publish APIs should be able to access the added document.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-109: Check doc visibility when it's given same as api visibility; api visibility = restricted by role [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Only the successfully logged in users who are in the same tenant domain as API creator and who has the role that is being sprcified should be able to see the documents having visibility as 'same as api visibility' of an api which has visibility set to 'restricted by roles' and specified roles.				
Preconditions: 1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an published api having api visibility level as 'restricted by roles' and role specified. 3. User who has permission to add documents should reside on docs page of the particular api. 5. <EnableAPIDocVisibilityLevels> should be set to true in <AM_HOME>/repository/conf/api-manager.xml				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	Add new document section should expand and visibility drop down also should be available.		
2	Add a 'How to' type in-line document given same as api visibility and check whether only the users who are in the same tenant domain as API creator and who has the role specified can view the added document.	Only the users who are in the same tenant domain as API creator and who has the role specified should be able to view the document.		
3	Add a 'Samples & SDK' type document with a file given same as api visibility and check whether only the users who are in the same tenant domain as API creator and who has the role specified can view the added document.	Only the users users who are in the same tenant domain as API creator and who has the role specified should be able to access the added document.		
4	Add a 'Public forum' type document with a URL given visibility as 'same as api visibility' and check whether only the users who are in the same tenant domain as API creator and who has the role specified can access the document.	Only the users who are in the same tenant domain as API creator and who has the role specified should be able to access the added document.		
5	Add a 'Support forum' type document with a URL given visibility as 'same as api visibility' and check whether users who are in the same tenant domain as API creator and who has the role specified can access the document.	Only the users who are in the same tenant domain as API creator and who has the role specified should be able to access the added document.		

6	Add a other type in-line document given visibility as 'same as api visibility' and check whether only the users who are in the same tenant domain as API creator and who has the role specified can access the document.	Only the users who are in the same tenant domain as API creator and who has the role specified should be able to access the added document.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-108: Check doc visibility when it's given same as api visibility; api visibility = visible to my domain. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Only the successfully logged in users who are in the same domain as API creator should be able to see the documents having visibility as 'same as api visibility' of an api which has visibility set to 'visible to my domain'.

Preconditions:
1. User with permission to add documents should be successfully logged in to the API publisher.

2. There should be an published api having api visibility level as visible to my domain.

3. User who has permission to add documents should reside on docs page of the particular api.

4. <EnableAPIDocVisibilityLevels> should be set to true in <AM_HOME>/repository/conf/api-manager.xml

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	Add new document section should expand and visibility drop down also should be available.		
2	Add a 'How to' type in-line document given same as api visibility and check whether only the users in the same domain as api creator can view the added document.	Only the users in the same domain as api creator should be able to view the document.		
3	Add a 'Samples & SDK' type document with a file given same as api visibility and check whether only the users in the same domain as api creator can view the added document.	Only the users in the same domain as api creator should be able to access the added document.		
4	Add a 'Public forum' type document with a URL given visibility as 'same as api visibility' and check whether only the users in the same domain as api creator can access the document.	Only the users in the same domain as api creator should be able to access the added document.		
5	Add a 'Support forum' type document with a URL given visibility as 'same as api visibility' and check whether only the users in the same domain as api creator can access the document.	Only the users in the same domain as api creator should be able to access the added document.		
6	Add a other type in-line document given visibility as 'same as api visibility' and check whether only the users in the same domain as api creator can access the document.	Only the users in the same domain as api creator should be able to access the added document.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-107: When user clicks on the '?' icon of the visibility field tip box should open. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user who has permission should be able to view the tip box which contains information about what is meant by each visibility option, by clicking on the '?' icon.

Preconditions:
1. User with permission to add documents should be successfully logged in to the API publisher.

2. There should be an api already published.

3. User who has permission to add documents should reside on docs page of the particular api.

4. <EnableAPIDocVisibilityLevels> should be set to true in <AM_HOME>/repository/conf/api-manager.xml

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document'	Add new document section should get expanded.		
2	Click on '?' icon of the visibility drop down.	Tip box which contains information about what is meant by each visibility option should open.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium

Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-106: Check document visibility when it's given same as api visibility when api visibility = public [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Create an api given public api visibility and add a 'How To' type document given visibility as 'same as api visibility'. Any user should be able to view the document added.				
Preconditions: 1. User with permission to add documents should be successfully logged in to the API publisher. 2. There should be an published api having api visibility level as public. 3. User who has permission to add documents should reside on docs page of the particular api. 4. <EnableAPIDocVisibilityLevels> should be set to true in <AM_HOME>/repository/conf/api-manager.xml				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on 'add new document' link	Add new document section should expand and visibility drop down also should be available.		
2	Add a 'How to' type in-line document given same as api visibility and check whether any one can view the added document.	Any user should be able to view the document.		
3	Add a 'Samples & SDK' type document with a file given same as api visibility and check whether any user can view the added document.	Any user should be able to access the added document.		
4	Add a 'Public forum' type document with a URL given visibility as 'same as api visibility' and check whether any user can access the document.	Any user should be able to access the added document.		
5	Add a 'Support forum' type document with a URL given visibility as 'same as api visibility' and check whether any user can access the document.	Any user should be able to access the added document.		
6	Add a other type in-line document given visibility as 'same as api visibility' and check whether any user can access the document.	Any user should be able to access the added document.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

2.9.Test Suite : API Console

This will contain all the test cases related to the API Console/Swagger

Test Artifact
Mind Map

2.9.1.Test Suite : Verify functionality without logging in to Store

This test scenario will be used to verify the scenarios of the API console without a subcriber login in to the API Console

Preconditions

1. Create a user with publish and create permissions
2. Login to the API Publisher and create an API allowing all/selected HTTP methods (there might be test cases where only a few methods should be selected)
3. Access the API Store

Test Artifact
Mind Map

2.9.1.1.Test Suite : Request Header

This scenario will have all the test cases related to request headers

Preconditions

1. Go to APIs and click on the API which you created
2. Go to the tab API Console

Test Artifact
Mind Map

Test Case APIM-383: Invoke the API without specifying a value for the request header [Version : 1]	
Author:	Evanthika Amarasiri

Summary:				
This test case will verify whether the correct error handling is done when invoking APIs without specifying the request headers				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke the GET/POST/PUT & DELETE methods of the API and verify the	1. All the requests should fail with the below error displayed under the 'Response Body' section <ams:fault xmlns:ams="http://wso2.org/apimanager/security"> <ams:code>900902</ams:code> <ams:message>Missing Credentials</ams:message> <ams:description>Required OAuth credentials not provided. Make sure your API invocation call has a header: "Authorization: Bearer ACCESS_TOKEN" </ams:description> </ams:fault>		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

2.9.2.Test Suite : Verify functionality while logged in to Store

This test scenario will be used to verify the scenarios of the API console when a subcriber is logged in to the API Console

Test Artifact	
Mind Map	

2.10.Test Suite : Workflows

Test Artifact	
Mind Map	

3.Test Suite : Server Profiles

Test Artifact
Mind Map

3.1.Test Suite : api-publisher

Test Artifact
Mind Map

Test Case APIM-1: Check whether the publisher app gets loaded properly [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-2: Check for other start up errors [Version : 1]	
Author:	Nirodha Gallage
Summary:	
Check whether there are any other errors or warnings you will get at the server startup log	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-3: check whether the GW and KM functionality are not working [Version : 1]	
Author:	Nirodha Gallage
Summary:	
When started with api-publisher profile you will be able to access the 'store' (due to limitations) but other than that GW and KM functionality should not work.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-4: Check whether the statistics are displayed (when configured with BAM) [Version : 1]	
Author:	Nirodha Gallage
Summary:	
Check whether the statistics are displayed when configured with BAM. It should show the stats properly without any errors.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None

Last Result	**Not Run**

3.2.Test Suite : api-store

Test Artifact
Mind Map

Test Case APIM-5: Check whether the store functionality works [Version : 1]	
Author:	Nirodha Gallage
Summary:	
Check whether the store gets loaded properly and the basic store functionality works when started with the api-store profile.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-6: Check for other start up errors [Version : 1]	
Author:	Nirodha Gallage
Summary:	
When started with the api-store profile you should not see any other errors due to the profile.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-7: Verify that GW and KM functionality are not working [Version : 1]	
Author:	Nirodha Gallage
Summary:	
When started with this profile it should not provide Gateway and Key manager functionality.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-8: Check whether the statistics are displayed (when configured with BAM) [Version : 1]	
Author:	Nirodha Gallage
Summary:	
Statics components should load when the server is started with store profile.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-9: Check the workflow integration [Version : 1]	
Author:	Nirodha Gallage
Summary:	
Workflow integration scenarios should work with the store profile as in https://docs.wso2.com/display/AM180/Adding+Workflow+Extensions	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

3.3.Test Suite : api-key-manager

Test Artifact
Mind Map

Test Case APIM-10: Check whether the key manager functionality works [Version : 1]	
Author:	Nirodha Gallage
Summary:	
When started with key-manager profile key manager and auth manager functioanlity should work properly.	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-11: Check for startup errors [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-12: Verify that the Publisher, store, Gateway functionality are blocked [Version : 1]	
Author:	Nirodha Gallage
Summary:	
After starting the server with keymanager profile it should block the publisher, store and gateway functionality	
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

3.4.Test Suite : gateway-manager

Test Artifact
Mind Map

Test Case APIM-13: Check whether the GW functionality work with the profile [Version : 1]

Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-14: Check for other start up errors [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-15: Verify that the key manager, publisher, store functionality are blocked [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

3.5.Test Suite : gateway-worker

Test Artifact	
Mind Map	

Test Case APIM-16: Check whether Gateway worker functionality work with this profile [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-17: Verify that the key manager, publisher, store functionality are blocked [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-18: Check for other start up errors [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-19: Check whether the stat publishing to BAM works properly [Version : 1]	
Author:	Nirodha Gallage
Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

4.Test Suite : APIM user roles

Test Artifact
Mind Map

Test Case APIM-427: Try to subscribe by a user who has only login permission [Version : 1]				
Author:	Ushani Balasooriya			
Summary: A user without subscrive persmission should not be able to susbcribe to an application				
Preconditions: 1. A user should be created and assigned to a role which does not have subscribe permission. 2. Create an API and publish it by a publisher				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the store by the user created with login permission	User should be sucessfully logged in to the store		
2	Try to subscribe to an app by that user	User should not be able to subscribe since they dont have subscribe permission		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

5.Test Suite : Throttling

@Sewmini, please move the test cases under Throttling under the relevant sub folders

Test Artifact
Mind Map

5.1.Test Suite : API Level Throttling

Test Artifact
Mind Map

Test Case APIM-156: Check whether Unlimited Resource level throttling tier works. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user with permission should be able to invoke API with maximum of 1-2 request per minute If the user is subscribed to an API which has Unlimited resource level throttling , using Unlimited api throttling tier.				
Preconditions:				
1. There should be a published api having all 4 api throttling tiers. 2. All reources of the API should have given Unlimited as throttling tiers. 3. A subscriber should have subscribed to the API as unlimited subscriprion and the application used for the subscription should have unlimitted as application throttling tier. 4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 300 request within a minute.	No request should fial due to throttling tier.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

5.2.Test Suite : Application Level Throttling

Test Artifact
Mind Map

Test Case APIM-155: Check whether Bronze Resource level throttling tier works. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
Successfully logged in user with permission should be able to invoke API with maximum of 1-2 request per minute If the user is subscribed to an API which has Bronze resource level throttling , using Unlimited api throttling tier.				
Preconditions:				
1. There should be a published api having all 4 api throttling tiers. 2. All reources of the API should have given Bronze as throttling tiers. 3. A subscriber should have subscribed to the API as unlimited subscriprion and the application used for the subscription should have unlimitted as application throttling tier. 4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 5 request within a minute.	after 1st or 2nd request, rest of the requests should fail.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

5.3.Test Suite : Resource Level Throttling

Test Artifact
Mind Map

Test Case APIM-154: Check whether Silver Resource level throttling tier works. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Successfully logged in user with permission should be able to invoke API with maximum of 5-6 request per minute If the user is subscribed to an API which has Silver resource level throttling , using Unlimited api throttling tier.				
Preconditions: 1. There should be a published api having all 4 api throttling tiers. 2. All reources of the API should have given Silver as throttling tiers. 3. A subscriber should have subscribed to the API as unlimited subscriprion and the application used for the subscription should have unlimitted as application throttling tier. 4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 15 request within a minute.	after 5th / 6th request, rest of the requests should fail.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

5.4.Test Suite : IP Level Throttling

Test Artifact
Mind Map

Test Case APIM-153: Check whether GOLD Resource level throttling tier works. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Successfully logged in user with permission should be able to invoke API with maximum of 20-21 request per minute If the user is subscribed to an API which has gold resource level throttling , using Unlimited api throttling tier.				
Preconditions: 1. There should be a published api having all 4 api throttling tiers. 2. All reources of the API should have given gold as throttling tiers. 3. A subscriber should have subscribed to the API as unlimited subscriprion and the application used for the subscription should have unlimitted as application throttling tier. 4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 30 request within a minute.	after 20th / 21th request, rest of the requests should fail.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

5.5.Test Suite : Other Throttling Scenarios

Test Artifact
Mind Map

Test Case APIM-152: Check whether Unlimited Application level throttling tier works. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Successfully logged in user with permission should be able to invoke API with maximum of unlimited request per minute If the user is subscribed to API as a Unlimited subscription.				
Preconditions: 1. There should be a published api having all 4 api throttling tiers. 2. All reources of the API should have given unlimited as throttling tiers. 3. A subscriber should have subscribed to the API as Unlimited subscriprion and the application used for the subscription should have Unlimited as application throttling tier. 4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 300 request within a minute.	No requests should fail due to throttling.		
Execution type:	Manual			

Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-151: Check whether GOLD API level throttling tier works. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to invoke API with maximum of 20-21 request per minute If the user is subscribed to API as a GOLD subscription.

Preconditions:

1. There should be a published api having all 4 api throttling tiers.

2. All reources of the API should have given unlimited as throttling tiers.

3. A subscriber should have subscribed to the API as GOLD subscriprion and the application used for the subscription should have unlimitted as application throttling tier.

4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 30 request within a minute.	after 20th / 21th request, rest of the requests should fail.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-150: Check whether Bronze Application level throttling tier works. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to invoke API with maximum of 1-2 request per minute If the user is subscribed to API as a Bronze subscription.

Preconditions:

1. There should be a published api having all 4 api throttling tiers.

2. All reources of the API should have given unlimited as throttling tiers.

3. A subscriber should have subscribed to the API as Unlimited subscriprion and the application used for the subscription should have Bronze as application throttling tier.

4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 8 request within a minute.	after 1th / 2th request rest of the requests should fail.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-149: Check whether Silver Application level throttling tier works. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to invoke API with maximum of 5-6 request per minute If the user is subscribed to API as a Silver subscription.

Preconditions:

1. There should be a published api having all 4 api throttling tiers.

2. All reources of the API should have given unlimited as throttling tiers.

3. A subscriber should have subscribed to the API as Unlimited subscriprion and the application used for the subscription should have Silver as application throttling tier.

4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 15 request within a minute.	after 5th / 6th request rest of the requests should fail.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-148: Check whether Unlimited API level throttling tier works. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to invoke API with maximum of unlimited request per minute If the user is subscribed to API as a Unlimited subscription.

Preconditions:

1. There should be a published api having all 4 api throttling tiers.

2. All reources of the API should have given unlimited as throttling tiers.

3. A subscriber should have subscribed to the API as Unlimited subscriprion and the application used for the subscription should have unlimitted as application throttling tier.

4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 200 request within a minute.	API should get invoked successfully for all requests		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-147: Check whether Bronze API level throttling tier works. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to invoke API with maximum of 1-2 request per minute If the user is subscribed to API as a Bronze subscription.

Preconditions:

1. There should be a published api having all 4 api throttling tiers.

2. All reources of the API should have given unlimited as throttling tiers.

3. A subscriber should have subscribed to the API as Bronze subscriprion and the application used for the subscription should have unlimitted as application throttling tier.

4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 5 request within a minute.	after 1th / 2th request, rest of the requests should fail.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-146: Check whether SILVER API level throttling tier works. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to invoke API with maximum of 5-6 request per minute If the user is subscribed to API as a SILVER subscription.

Preconditions:

1. There should be a published api having all 4 api throttling tiers.

2. All reources of the API should have given unlimited as throttling tiers.

3. A subscriber should have subscribed to the API as SILVER subscriprion and the application used for the subscription should have unlimitted as application throttling tier.

4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:

1	Send 10 request within a minute.	after 5th / 6th request, rest of the requests should fail.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-144: Check whether tier of api loads correctly in api store according to the tier configure in publisher. [Version : 1]

Author: Sewmini Jayaweera

Summary:

Tiers of the API should load correctly and display to the subscribers on API store > API > <particular api> overview page as specified tiers when creating the API

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Create API given unlimited tier and publish.	On the API store particular API should have only unlimited tier on tier drop down list.		
2	Create an API given gold tier and publish.	On the API store particular API should have only gold tier on tier drop down list.		
3	Create an api given silver tier and publish.	On the API store particular API should have only silver tier on tier drop down list.		
4	Create an api given bronze tier and publish.	On the API store particular API should have only bronze tier on tier drop down list.		
5	create an api given gold,silver,bronze and unlimited tiers and publish.	On the API store particular API should have gold,silver,bronze and unlimited tier on tier drop down list.		
6	Create an api given gold and silver as tiers and publish.	On the API store particular API should have gold and silver tier on tier drop down list.		
7	Create api given gold and bronze as tier and publish.	On the API store particular API should have gold and bronze tiers on tier drop down list.		
8	Create api given gold and unlimited tiers and publish.	On the API store particular API should have gold and unlimited tiers on tier drop down list.		
9	Create api given silver and bronze tiers and publish	On the API store particular API should have silver and bronze tiers on tier drop down list.		
10	Create api given silver and unlimited as tiers and publish	On the API store particular API should have silver and unlimited tiers on tier drop down list.		
11	Create api given unlimited and bronze tiers and publish.	On the API store particular API should have bronze and unlimited tier on tiers drop down list.		
12	Create api given gold,silver and bronze tiers and publish.	On the API store particular API should have gold, silver and bronze tiers on tier drop down list.		
13	Create api given gold,silver and unlimited tiers and publish.	On the API store particular API should have gold, silver and unlimited tiers on tier drop down list.		
14	Create api given silver, bronze and unlimited tiers and publish.	On the API store particular API should have silver, bronze and unlimited tiers on tier drop down list.		
15	Create api given gold, bronze and unlimited.	On the API store particular API should have gold,silver and unlimited tiers on tier drop down list.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-145: Check whether Gold Application level throttling tier works. [Version : 1]

Author: Sewmini Jayaweera

Summary:

Successfully logged in user with permission should be able to invoke API with maximum of 20-21 request per minute If the user is subscribed to API as a gold subscription.

Preconditions:

1. There should be a published api having all 4 api throttling tiers.

2. All reources of the API should have given unlimited as throttling tiers.

3. A subscriber should have subscribed to the API as Unlimited subscriprion and the application used for the subscription should have GOLD as application throttling tier.

4. API manager should contain default tier configurations which is gold - 20 requests per mins, silver - 5 requests per min, bronze - 1 request per min and unlimited - umlimited requests per min.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 30 request within a minute.	after 20th / 21th request rest of the requests should fail.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			

Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-157: Check whether throttling works correctly given different throttling tiers for different resources. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

When each resource given a different throttling tier In the same api, Each resource should show correct behavior according to it's throttling level.

Preconditions:

1. There should be an API given reource level throttling tiers as bellow.

GET - Unlimited , Post - Gold , Put - silver, Delete - bronze

2. A subscriber must subscribe to API as Unlimited subscription using an application which has unlimited as it's throttling tier.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke API with 200 Get requests per min.	All the requests should be successful(should not fail due to throttling)		
2	Invoke API with 50 PUT requests per min.	Only first 5-6 request should be successfull. Other requests should fail due to exceeding throttling tier.		
3	Invoke API with 50 POST requests per min.	Only first 20-21 request should be successfull. Other requests should fail due to exceeding throttling tier.		
4	Invoke API with 10 DELETE requests per min.	Only first 1-2 request should be successfull. Other requests should fail due to exceeding throttling tier.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-162: Check whether system behaves correctly when each resource have different throttling tiers. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check where requests hit API level throttling, application level throttling resource level throttling when user invokes different resources which has a different throttling tier for each resource.

Preconditions:

1. There should be no changes done to the default thottling configs comes with API manager.

2. There should be an API created given unlimited for get resource, silver for post resource, bronze for put resources and gold gor delete resource.

3. There should be a subscriber who has a gold subscription to the API using an application which is throttling tier is Silver.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send; • 4 - bronze requests • 10 - post requests • 6 - delete requests within a minute.	• only the first bronze request and first post request should be successfull. • second post request onwards all the other requests should fail.		
2	In the next minute again send the requests as specified above.	You should see the same expected results as above.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-164: Check the behaviou when user subscribe to same API to 2 applications with different throttling tiers [Version : 1]

Author:	Sewmini Jayaweera			
Summary: Check whether throttling works when user do unlimitted subscription to the same api using two different application which has gold and silver throttling tiers respectively.				
Preconditions: 1. There should be an API created an dpublished having throttling tiers for resources as unlimited 2.There should be a subscriber subscribed to the same API twise using 2 different applications which have gold (app1) and silver (app2) having selected unlimited as API throttling tier.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke api through app1 and send 25 requests within a min.	after first 20 requests rest of the requests should fail due to throttling.		
2	Invoke API through app2 and send 15 requests within a min.	after first 5 requests of the requests others should fail due to throttling.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

6.Test Suite : Caching

Test Artifact	
Mind Map	

Test Case APIM-428: Verify JWT caching [Version : 1]				
Author:	Ushani Balasooriya			
Summary: Verify JWT caching works in a setup				
Preconditions: 1. A clustr/single node should have been setup 2. Enable JWT and JWT caching in api-manager.xml by enabling EnableTokenGeneration>true</EnableTokenGeneration> and <EnableJWTCache>true</EnableJWTCache> 3. Enable JWT cache and Key Manager cache. Key manager caching should be enabled to cache the JWT token in key manager side. Note: In here it is better to disable GW caching, because if by any chance GW caching is on, it will cache the JWT token until any change is done in the store. Once any of the activity happens in store (e.g., delete app, edit app) it will clear the token in gateway when the gateway caching is enabled 4. API should be created and published to the store				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Subscribe to an application and invoke by sending a request and decode the JWT assertion receieved in response ({token infor}.{claims list}.{signature})and movitor the claims list	It should have values as below. E.g., {"iss":"wso2.org/products/am","exp":1432899350680,"http://wso2.org/claims/subscriber":"admin","http://wso2.org/claims/applicationid":"2","http://wso2.org/claims/applicationname":"app3","http://wso2.org/claims/applicationtier":"Unlimited","http://wso2.org/claims/apicontext":"/api3/1.0.0/test","http://wso2.org/claims/version":"1.0.0","http://wso2.org/claims/tier":"Unlimited","http://wso2.org/claims/keytype":"PRODUCTION","http://wso2.org/claims/usertype":"APPLICATION","http://wso2.org/claims/enduser":"admin@carbon.super","http://wso2.org/claims/enduserTenantId":"-1234"}		
2	Send another request and decode the JWT assertion receieved in response	Since the JWT caching is enabled expiration time will be same		
3	Disable JWT cache and send another request to invoke and monitor the decoded JWT assertion receieved in response	Since the token is not taken from the cache, expiration time will be different		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	High			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

7.Test Suite : API invocation

Test Artifact
Mind Map

7.1.Test Suite : Resources - Auth types

Test Artifact
Mind Map

Test Case APIM-194: Check whether changes applies when user change auth type of a resource 'application & application user' to 'application user' [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Check whether the changes reflect properly once user change auth type of a resource application & application user to application user. - when user changes the the auth type to application, only user tokens can be used to access the particular resource.				
Preconditions: 1. There should be an published API which has a resource (resource 1) given 'application and application user' as the auth type. 2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time. 3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application user' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should not be able to access the resource.		
3	Invoke resource using **application** token.	User should not be able to access API using application token.		
4	Invoke resource using a **user** token of a registered user.	User should be able to access the resource.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-193: Check whether changes applies when user change auth type of a resource 'application & application user' to application [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Check whether the changes reflect properly once user change auth type of a resource 'application & application user' to 'application'. - when user changes the the auth type to application only application token can be used to access the particular resource.				
Preconditions: 1. There should be an published API which has a resource (resource 1) given 'application & application user' as the auth type. 2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time. 3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should not be able to access the resource.		
3	Invoke resource using **application** token.	User should be able to access API using application token.		
4	Invoke resource using a **user** token of a registered user.	User should not be able to access the resource.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-192: Check whether changes applies when user change auth type of a resource 'application & application user' to none [Version : 1]

Author:	Sewmini Jayaweera
Summary:	Check whether the changes reflect properly once user change auth type of a resource 'application and application user' to 'none'. - when user changes the the auth type to none, no user tokens are neccesary to invoke the particular resource.
Preconditions:	1. There should be an published API which has a resource (resource 1) given 'application & application user' as the auth type. 2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time. 3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application user' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should be able to access the resource.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-191: Check whether changes applies when user change auth type of a resource 'application user' to application & application user' [Version : 1]

Author:	Sewmini Jayaweera
Summary:	Check whether the changes reflect properly once user change auth type of a resource application user to 'application & application user'. - when user changes the the auth type to application and application user, user should be able to access resource either using application token or user token.
Preconditions:	1. There should be an published API which has a resource (resource 1) given 'application user' as the auth type. 2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time. 3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application and application user' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should not be able to access the resource.		
3	Invoke resource using **application** token.	User should be able to access API using application token.		
4	Invoke resource using a **user** token of a registered user.	User should be able to access the resource.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-190: Check whether changes applies when user change auth type of a resource 'application user' to 'application' [Version : 1]

Author:	Sewmini Jayaweera
Summary:	Check whether the changes reflect properly once user change auth type of a resource 'application user' to 'application'. - when user changes the the auth type to application only application token can be used to access the particular resource.
Preconditions:	1. There should be an published API which has a resource (resource 1) given 'application user' as the auth type. 2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time. 3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should not be able to access the resource.		
3	Invoke resource using **application** token.	User should be able to access API using application token.		
4	Invoke resource using a **user** token of a registered user.	User should not be able to access the resource.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-189: Check whether changes applies when user change auth type of a resource 'application' to 'none' [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check whether the changes reflect properly once user change auth type of a resource 'application' to 'none'. - when user changes the the auth type to none, no user tokens are neccesary to invoke the particular resource.

Preconditions:

1. There should be an published API which has a resource (resource 1) given 'application' as the auth type.

2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time.

3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application user' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should be able to access the resource.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-188: Check whether changes applies when user change auth type of a resource 'none' to 'application user' [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check whether the changes reflect properly once user change auth type of a resource none to application user. - when user changes the the auth type to application, only user tokens can be used to access the particular resource.

Preconditions:

1. There should be an published API which has a resource (resource 1) given 'none' as the auth type.

2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time.

3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application user' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should not be able to access the resource.		
3	Invoke resource using **application** token.	User should not be able to access API using application token.		
4	Invoke resource using a **user** token of a registered user.	User should be able to access the resource.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				

Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-187: Check whether changes applies when user change auth type of a resource 'application' to 'application & application user' [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check whether the changes reflect properly once user change auth type of a resource application to 'application & application user'. - when user changes the the auth type to application and application user, user should be able to access resource either using application token or user token.

Preconditions:

1. There should be an published API which has a resource (resource 1) given 'application' as the auth type.

2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time.

3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application and application user' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should not be able to access the resource.		
3	Invoke resource using **application** token.	User should be able to access API using application token.		
4	Invoke resource using a **user** token of a registered user.	User should be able to access the resource.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-186: Check whether changes applies when user change auth type of a resource 'none' to 'application & application user [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check whether the changes reflect properly once user change auth type of a resource none to 'application & application user'. - when user changes the the auth type to application and application user, user should be able to access resource either using application token or user token.

Preconditions:

1. There should be an published API which has a resource (resource 1) given 'none' as the auth type.

2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time.

3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application and application user' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should not be able to access the resource.		
3	Invoke resource using **application** token.	User should be able to access API using application token.		
4	Invoke resource using a **user** token of a registered user.	User should be able to access the resource.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-185: Check whether changes applies when user change auth type of a resource 'application' to 'application user' [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check whether the changes reflect properly once user change auth type of a resource 'application' to 'application user'. - when user changes the the auth type to application, only **user** tokens can be used to access the particular resource.

Preconditions:

1. There should be an published API which has a resource (resource 1) given 'application' as the auth type.

2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time.

3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application user' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should not be able to access the resource.		
3	Invoke resource using **application** token.	User should not be able to access API using application token.		
4	Invoke resource using a **user** token of a registered user.	User should be able to access the resource.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-182: Check whether resources with auth type non can be access with no access token. **[Version : 1]**

Author: Sewmini Jayaweera

Summary:

Check whether users can invoke api resources which has auth type non with no access tokens.

Preconditions:

1. There should be an API(api1) with a secured resource(resource1) using auth type as non.

2. There should be a user(application owner) who is successfully logged in to the store and subscribed to api1

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Users should be able to access api resources which are not secured by (resource auth type as non)	user should be able to invoke resource.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-181: Check wether resources given auth type as 'application user' can be invoked using user access tokens. **[Version : 1]**

Author: Sewmini Jayaweera

Summary:

Check wheter system allows users to invoke resources given 'application users' auth type, using any valid user access tokens.

Preconditions:

1. There should be an API(api1) with a secured resource(resource1) using auth type as application user.

2. There should be a user(application owner) who is successfully logged in to the store and subscribed to api1

3. User should (application owner) have generated tokens given valid time as -1

4. There should be another user (user1) registred under api store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Use the consumer key, consumer secret of the application which api1 is being subscribed, and user credentials of user1 and generate a token.	Avalid user token should be generated.		
2	Invoke api1 reource1 using the token generated.	User should be able to invoke api1 resource1 successfully.		
3	Try to invoke api1 resource1 with the application access token given for the particular application (access token which is given in the api store for the particular application.).	resource1 should not be able to invoke using application access token.		

Execution type:	Manual
Estimated exec. duration (min):	

Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-180: Check whether resources secured using 'application' auth type cannot be invoked using user access tokens. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Registered users should not be able to invoke a resource which is protected using 'application' auth type, using user access tokens generated for users.

Preconditions:

1. There should be an API(api1) with a secured resource(resource1) using auth type as application.

2. There should be a user(application owner) who is successfully logged in to the store and subscribed to api1

3. User should have generated tokens given valid time as -1

4. There should be another user (user1) registred under api store.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Use the consumer key, consumer secret of the application which api1 is being subscribed, and user credentials of user1 and generate a token.	Avalid user token should be generated.		
2	Invoke resource1 of api1 using generated application token.	user should not be able to invoke resource1.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-179: Check whether resources secured using 'application' auth type can be accessed using application token generated for the application. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check whether system allows to access reosurces secured with 'application' auth type, using application token.

Preconditions:

1. There should be an API(api1) with a secured resource(resource1) using auth type as application.

2. There should be a user who is successfully logged in to the store and subscribed to api1

3. User should have generated tokens given valid time as -1

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke resource1 of the api1 using the generated application token.	User should be able to invoke reource1 using access token.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-183: Check whether each resource show correct behaviour according to the auth type assigned to the resource. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check whether each resource show correct behavior according to the auth type assinged to the resource.

Preconditions:

1. There should be a created api(api1) given different resource paths as resource1, resource2, resource3, resource4 and assigned auth types none, application, application user, application and application user respectively.

2. A user (user1) should be successfully logged into the publisher, subscribed to api1 under application 1 and generated access tokens.

3. There should be registered users as user2 , user3

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	access resource 1 without given an access token.	User should be able to access resource 1 without giving an access token.		
2	Use application token (on user one my subscription page relavant to the application) invoke reource 2.	user 1 should be able to access resource 1 successfully.		
3	Use user token of user 1 and invoke reource 3.	User 1 should not be able to access resource 3		
4	Use access token of user 1 and invoke reource 4 .	user 1 should be able to invoke resource 4		
5	use consumer key and consumer secret give user name and password of user 2 and generage access token 2	User 2 should get a valid access token.		
6	use access token 2 and invoke resource 2.	user 2 should not be able to access resource 2		
7	Use access token 2 and invoke resource 3	user 2 should be able to access resource 3		
8	Use access token 3 and invoke resource 4	user 2 should be able to access resource 4		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-184: Check whether changes applies when user change auth type of a resource 'none' to 'application'. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Check whether the changes reflect properly once user change auth type of a resource none to application. - when user changes the the auth type to application only application token can be used to access the particular resource.

Preconditions:

1. There should be an published API which has a resource (resource 1) given 'none' as the auth type.

2. There should be a user who has subscribed to the API using an application and generated keys given -1 as the validity time.

3. Gateway cach should be disabled by setting `EnableGatewayResourceCache to false in <APIM_HOME>/repository/conf/api-manager.xml` file.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to the publisher, open edit view of the API, change the auth type of the resource to 'application' in the manage stage and click save and publish.	Change should be reflect when user reopens the edit view of the API and check the resource on the manage stage.		
2	Invoke the resource without a user token.	User should not be able to access the resource.		
3	Invoke resource using **application** token.	User should be able to access API using application token.		
4	Invoke resource using a **user** token of a registered user.	User should not be able to access the resource.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

7.2.Test Suite : lifecycle

Test Artifact	
Mind Map	

7.3.Test Suite : scopes

Test Artifact	
Mind Map	

Test Case APIM-406: A user should not be able to invoke an API method which is not bound to a paticular scope that is assigned to a different role once the inovkable scope is changed to a scope which is mapped to the different role [Version : 1]

Author:	Ushani Balasooriya

Summary:

A user should not be able to invoke an API method which is not bound to a paticular scope that is assigned to a different role once the inovkable scope is changed to a scope which is mapped to the different role

Preconditions:

1. A user should be logged in to publisher.

2. Create an api and assign a scope which is bound to a paticular role. E.g., scope 1 - role 1 for GET resource of API1

3. API should have been published

4. User assigned to role1 should be logged in to the store

5. Subscribe API1 to an application (E.g., app1)

6. Select scope1 from the popup and generate a token

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke the API1 by GET	API should be invoked		
2	Edit the API1 by assigning scope2 to GET method which is assigned to role2 and publish it	It should be assigned to scope2		
3	Login to store by user1 (belongs to role1) and regenerate token and invoke the GET methos	User should not be able to invoke. I thouls return an error message mentioning the token is not valid for the resource or etc.		
4	Edit the API vise versa (Assign a scope which is assigned to role1, publish and invoke by the role 1 user)	User should be able to invoke again.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-365: A user should be able to generate a token via CURL for a paticular scope of an API when there are multiple scopes defined for an API and invoke [Version : 1]

Author: Ushani Balasooriya

Summary:

A user should be able to generate a token via CURL for a paticular scope of an API when there are multiple scopes defined for an API and invoke

Preconditions:

1. A user should be logged in to publisher.

2. Create an api and assign a scope which is bound to a paticular role. E.g., scope 1 - role 1 for GET resource of API1 and scope 2 - role2 for POST

3. API should have been published

4. User assigned to role1 should be logged in to the store

5. Subscribe API1 to an application (E.g., app1)

6. User should have executed steps in APIM-407

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke by the generated token (scope2) by APIM-407 step3 for GET method by user 2	User should not be able to invoke since scope 1 (role1) is assigned to GET and user2 belongs to role2		
2	Try to invoke by the generated token (scope2) by APIM-407 step3 for POST method by user2	User should be able to invoke sucessfully since scope2 (role2) is assigned to POST method		
3	This should work vise versa for step2 in APIM-407. Invoke GET method by token generated for scope 1 by user1 (role 1)	User should be able to invoke since the scope 1 is assigned to GET		
4	This should work vise versa for step2 in APIM-407. Invoke POST method by token generated for scope 1 by user1 (role 1)	User should not be able to invoke since the POST is assigned to scope 2 (assigned to role 2)		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.Test Suite : Statistics

Test Artifact
Mind Map

8.1.Test Suite : API Store Statistics

Test Artifact
Mind Map

8.1.1.Test Suite : Configure with BAM (1.8.0 >)

Note : Statistics doesn't update real time therefore it will take 2-3 mins to update statistics and reflect the latest changes.

Test Artifact
Mind Map

8.1.1.1.Test Suite : permission based

Test Artifact
Mind Map

Test Case APIM-246: Check whether users logged into publisher who only has creator role can view both All Statistics and 'My APIs' > Statistics tabs on the side menu. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: User who has successfully logged into the API publisher and who has Creator role assigned should be able to view APIs > All Statistics as well as My APIs > statistics tabs on the menu bar reside on the left side of the page.				
Preconditions: 1. There should be a user who has only login and creator permission. 2. User should be successfully logged into API publisher. 3. API manager should be integrated with BAM by following the steps in [1] [1] https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Check whether APIs > All statistics and My APIs > statistics are available on side menu.	All statistics should be listed under APIs and statistics should be listed under 'My APIs'		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-245: Check whether users logged into publisher who only has publisher role cannot view "My APIs > statis [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: User who has successfully logged into the API publisherand who has publisher role assigned should not be able to view statistics menu item listed under 'My APIs' but 'All Statistics' listed under 'APIs' .				
Preconditions: 1. There should be a user who has only login and publish permission. 2. User should be successfully logged into API publisher. 3. API manager should be integrated with BAM by following the steps in [1] [1] https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Check whether only the 'All Statistics' and 'Browse' tabs are available in the menu on the left side of the page.	Only only the 'All Statistics' and 'Browse' tabs should be listed under 'APIs'. 'My APIs' or any menu items under that should NOT be available.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				

Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.1.2.Test Suite : Subscription statistics

Test Artifact
Mind Map

Test Case APIM-248: Check whether system depicts correct subscription statistics to user who has creator permission when system has APIs created two different creators. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Each user who has login to API publisher and creator permission should be able to view correct statistics for API subscriptions under all statistics and My APIs.				
Preconditions: 1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 (API1,API3) and another published API (API2) created by creator2 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2 and API3. 4. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1 and go to 'Overall API Subscriptions' by clicking on APIs > All statistics > API subscriptions.	• Three graphs should be available (for API1, API2 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/3 % (33%) should depic in the graph created for API1 and 1/3% (33%) should depict in the graph created for API2 and 1/3% (33%) should depict in the graph created for API3.		
2	Then Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Only two graphs should be available (for API1 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/2 % (50%) should depic in the graph created for API1 and 1/2% (50%) should depict in the graph created for API3. NOTE : calculation should be as below. (# of subscriptions a perticular API has) / (Total number of subscriptions have for the APIs, created by the particular API creator)		
3	Log out and loggin to API publisher as Creator2 and go to APIs > All Statistics > API subscriptions.	• Three graphs should be available (for API1, API2 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/3 % (33%) should depic in the graph created for API1 and 1/3% (33%) should depict in the graph created for API2 and 1/3% (33%) should depict in the graph created for API3.		
4	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Only a graph should be available (for API2) having API name, creator and number of subscription listed below the graph. • In this scenario number of subscription should be 1 and graph should depic 100%.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-247: Check whether system depicts correct subscription statistics to user who has creator permission when system only has APIs created by himself. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: User who has successfully logged into API Publisher with login and creator permission should be able to view correct API subscription statistics when there are only APIs created by him with subscriptions.				
Preconditions: 1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be three APIs created by creator1. (API1, API2, API3) 3. There should be no subscriptions for API3, one subscription for API1 and 2 subscriptions for API2. 4. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:

1	Login to API publisher as creator1 and go to 'Overall API Subscriptions' by clicking on APIs > All statistics > API subscriptions.	• Two graphs should be available (Since only 2 of the APIs have subscriptions) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/3 % should depic in the graph created for API1 and 2/3 should depict in the graph created for API2.		
2	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Two graphs should be available (Since only 2 of the APIs have subscriptions) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/3 % (33%) should depic in the graph created for API1 and 2/3% (67%) should depict in the graph created for API2.		
3	Log out and loggin to API publisher as Creator2 and go to APIs > All Statistics > API subscriptions.	• Two graphs should be available (Since only 2 of the APIs have subscriptions) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/3 % (33%) should depic in the graph created for API1 and 2/3% (67%) should depict in the graph created for API2.		
4	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	No data found or similar message should display on My API subscription page.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-249: Check whether Statistics get update properly once subscriber unsubscribe from the API. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User who has permission should be able to see updated subscription statistics accurately when an existing subscriber unsubscribe from an API.

Preconditions:

1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager.

2. There should be two published APIs created by creator1 (API1,API3) and another published API (API2) created by creator2

3. There should be two subscribers (subscriber1 and subscriber2) both have subscriptions for API1, API2 AND API3.

4. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1 and go to 'Overall API Subscriptions' by clicking on APIs > All statistics > API subscriptions.	• Three graphs should be available (for API1, API2 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 2/6 % (33%) should depic in the graph created for API1 and 2/6% (33%) should depict in the graph created for API2 and 2/6% (33%) should depict in the graph created for API3.		
2	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Only two graphs should be available (for API1 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 2/4 % (50%) should depic in the graph created for API1 and 2/4% (50%) should depict in the graph created for API3. NOTE : calculation should be as below. (# of subscriptions a perticular API has) / (Total number of subscriptions have for the APIs, created by the particular API creator)		
3	Log out and loggin to API publisher as Creator2 and go to APIs > All Statistics > API subscriptions.	• Three graphs should be available (for API1, API2 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 2/6 % (33%) should depic in the graph created for API1 and 2/6% (33%) should depict in the graph created for API2 and 2/6% (33%) should depict in the graph created for API3.		
4	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Only a graph should be available (for API2) having API name, creator and number of subscription listed below the graph. • In this scenario number of subscription should be 2 and graph should depic 100%.		
5	Login as subscriber1 and unsubscribe API1 subscription.	Subscriber1 should unsubscribe from API1 successfully.		
6	Wait for 2-3 mins, login to API publisher as creator1 and go to 'Overall API Subscriptions' by clicking on APIs > All statistics > API subscriptions.	Change should reflect as below. • Statistics of API 1subscription should decrease to 20% (1/5%) • Statistics of API2 subscription should increase to 2/5% (40%) • Statistics of API3 subscription should increase to 2/5 (40%)		
7	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	Subscription statistics of API1 and API3 should be 33% and 67% respectively.		

8	Log out and loggin to API publisher as Creator2 and go to APIs > All Statistics > API subscriptions.	Change should reflect as below. • Statistics of API 1subscription should decrease to 20% (1/5%) • Statistics of API2 subscription should increase to 2/5% (40%) • Statistics of API3 subscription should increase to 2/5 (40%)		
9	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Only a graph should be available (for API2) having API name, creator and number of subscription listed below the graph. • In this scenario number of subscription should be 2 and graph should depic 100%.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-250: Check whether user can successfully view subscription statistics relevant to a given time duration. **[Version : 1]**

Author:	Sewmini Jayaweera

Summary:

User who is successfully logged into the API publisher who has login, creator permission should be able to view API subscription statistics relevant to a time duration specified by the user.

Preconditions:

1. There should be a user (creator1) having login and creator permission, registered in API manager.

2. There should be two published APIs created by creator1 (API1,API2).

3. There should be a subscriber (subscriber1) subscribed to AP11 and API2 on day1, and another subscriber (subscriber2) subscribed to API2 on day2.

4. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1 and go to 'Overall API Subscriptions' by clicking on APIs > All statistics > API subscriptions.	User should successfully get redirect to the "Overall API Subscriptions (Across All Versions)" and API subscription statistics relavent to the default time duration chosen should be available.		
2	Set time duration in a way it only includes day one.	• Two graphs should be available (for API1, API2) having API name, Creator and number of subscriptions below each graph. • Statistics of API 1subscription should decrease to 50% • Statistics of API2 subscription should increase to 50%		
3	Reset time duration in a way it only includes day two.	• A graphs should be available (forAPI2) having API name, Creator and number of subscriptions below each graph. • Statistics of API 1subscription should decrease to 100%		
4	Go to My APIs > statistics > API subscriptions and set time duration in a way it only includes day1	• Two graphs should be available (for API1, API2) having API name, Creator and number of subscriptions below each graph. • Statistics of API 1subscription should decrease to 50% • Statistics of API2 subscription should increase to 50%		
5	Reset time duration in a way it only includes day two.	• A graphs should be available (forAPI2) having API name, Creator and number of subscriptions below each graph. • Statistics of API 1subscription should decrease to 100%		
6	Reset time duration in a way it includes both day1 and day2	• Two graphs should be available (for API1, API2) having API name, Creator and number of subscriptions below each graph. • Statistics of API 1subscription should decrease to 1/3% • Statistics of API2 subscription should increase to 2/3%		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.1.3.Test Suite : Usage statistics

Test Artifact
Mind Map

Test Case APIM-255: Check whether User can successfully view API usage statistics relevant to a particular time duration. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User who has successfully logged into API Publisher should be able to view API usage statistics of the APIs based on the				
Preconditions:				
1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be three two published APIs created by creator1 and creator2 as API1 and API2 respectively. 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. 5. API1 and API2 should be invoked 10 times each on during day 1. 6. API1 and API2 should be invoked 15 times and 5 times respectively. [1] https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher using creator1 and go to APIs > All statistics > API usage page and set the date range to include only day1.	• The pie chart should depic api invocation percentage related to API1 and API2 as 50% each. • Table should list API calls related to API1 as 10 and API calls related API2 as 10.		
2	Reset date range to include both day1 and day2.	• Pie chart should depict API invocation percentage related to API1 as 25/40 % • Pie chart should depict API invocation percentage related to API2 as 15/40 % • Table will list number of api calls related to API1 as 25 and number of API calls related to API2 as 15.		
3	Reset date range to include only day2 as the date range.	• The pie chart should depic api invocation percentage related to API1 as 75% and API2 as 25% each. • Table should list API calls related to API1 as 15 and API calls related API2 as 5.		
4	Go to MyAPIs > statistics > API usage page and select date range to inlude only the day1.	• Pie chart should depict API call percentage as 100% • When hover API name will Appear as API1 and number of calls will appear as 10 • In the able only API1 will be listed having number of calls as 10.		
5	Re-login to API publisher creator2, go to APIs > All statistics > API usage page and set the date range to include only day1.	• The pie chart should depic api invocation percentage related to API1 and API2 as 50% each. • Table should list API calls related to API1 as 10 and API calls related API2 as 10.		
6	Reset date range to include both day1 and day2.	• The pie chart should depic api invocation percentage related to API1 as 75% and API2 as 25% each. • Table should list API calls related to API1 as 15 and API calls related API2 as 5.		
7	Go to MyAPIs > statistics > API usage page and select date range to inlude only the day2.	• Pie chart should depict API call percentage as 100% • When hover API name will Appear as API2 and number of calls will appear as 5 • In the able only API2 will be listed having number of calls as 5.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-253: Check whether API manager statistics are updated accurately when another version of the same API is [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
When user has invoked two versions of the same API, the total number of invocations should list under API usage statistics of the particular API correctly.				
Preconditions:				
1. Execute test case 252 2. Get a copy of API1 (API1 2.0.0) publish it 2. Subscribe to 2.0.0 version of API1 and invoke it 10 times.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:

1	Login to API publisher using creator1 and go to APIs > All statistics > API usage page.	• There should be a pie chart which depicts API usage. • There should be a table which contains API name and number of API calls. • Number of API calls relevant to API1 should be listed as 15 • Number of API calls relevant to API2 should be listed as 10 • Pie chart should depic this information accurately by showing 60% for API1 and 40% for API2.		
2	Go to My APIs > statistics > API usage page.	• Pie chart should depic Number of API calls to API1 as 100% (15 calls) • Table should list only API1 and number of calls related to API1 as 15.		
3	Logout and login as creator2 and go to APIs > All statistics > API usage page.	• There should be a pie chart which depicts API usage. • There should be a table which contains API name and number of API calls. • Number of API calls relevant to API1 should be listed as 15 • Number of API calls relevant to API2 should be listed as 10 • Pie chart should depic this information accurately by showing 60% for API1 and 40% for API2.		
4	Go to My APIs > statistics > API usage page.	• Pie chart should depic percentage of API calls related to API2 as 100% • When hover the pie chart it should give name of the API as API2 and number of calls as 10. • Table should list API2 and number of API calls as 10.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-252: Check whether API creator can successfully view API usage statistics of API created by him as well as Usage statistics available for all APIs. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User who has successfully logged into API Publisher should be able to view API usage statistics of the APIs create by the particular user as well as usage statistics of all the APIs available in API manager.

Preconditions:

1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager.

2. There should be three two published APIs created by creator1 and creator2 as API1 and API2 respectively.

3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2.

4. API manager should be integrated with BAM by following [1].

5. Subscriber1 should have invoked API1 5 times successfully.

6. Subscriber2 should have invoked API2 10 times successfully.

[1] https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher using creator1 and go to APIs > All statistics > API usage page.	• There should be a pie chart which depicts API usage. • There should be a table which contains API name and number of API calls. • Number of API calls relevant to API1 should be listed as 5 • Number of API calls relevant to API2 should be listed as 10 • Pie chart should depic this information accurately by showing 33.3% for API1 and 66.7% for API2.		
2	Go to My APIs > statistics > API usage page.	• Pie chart should depic Number of API calls to API1 as 100% (5 calls) • Table should list only API1 and number of calls related to API1 as 5.		
3	Logout and login as creator2 and go to APIs > All statistics > API usage page.	• There should be a pie chart which depicts API usage. • There should be a table which contains API name and number of API calls. • Number of API calls relevant to API1 should be listed as 5 • Number of API calls relevant to API2 should be listed as 10 • Pie chart should depic this information accurately by showing 33.3% for API1 and 66.7% for API2.		
4	Go to My APIs > statistics > API usage page.	• Pie chart should depic percentage of API calls related to API2 as 100% • When hover the pie chart it should give name of the API as API2 and number of calls as 10. • Table should list API2 and number of API calls as 10.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium

Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-254: Check whether Usage statistics gets updated accurately when a particular API is deleted from API man [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Once an API is deleted from API manager usage data relevant to the particular API should not be available in the API usage statistics.				
Preconditions: 1. Execute 252 test case.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Unsubscribe all subscriptions which API2 has and delete API2.	API2 should get deleted successfully.		
2	Login to the publisher as creator1 and go to APIs > all statistics > API usage page.	• Pie chart should depic percentage API calls as 100% • Ehen hovering it should display API name as API1 and 5 call(s). • In the table only API1 should listed and the number of API calls should display as 5.		
3	Go to My APIs > statistics > API usage page.	• Pie chart should depic percentage API calls as 100% • Ehen hovering it should display API name as API1 and 5 call(s). • In the table only API1 should listed and the number of API calls should display as 5.		
4	Logout and login as creator2 and go to APIs > All Statistics > API usage page.	Page should display 'No data found' or similar message.		
5	Go to My APIs > statistics > API usage page.	Page should display 'No data found' or similar message.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

8.1.1.4.Test Suite : response time

Test Artifact	
Mind Map	

Test Case APIM-256: Check whether the API response time statistics are accurate. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: User should be able to successfully view the average API response time of APIs.				
Preconditions: 1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be three two published APIs created by creator1 and creator2 as API1 and API2 respectively. 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. 5. Invoke API1 100 times using Jmeter and get average latancy using "Aggregate Report". 6. Invoke API2 100 times using Jmeter and get average latancy using "Aggregate Report". [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher using creator1 and go to APIs > All statistics > API response time page.	• API respose time of API1 should be less than or equals to the average time got from API1 jmeter invocation . • API response time of API2 should be less than oe equals to the average time got from API2 jmeter invocation.		
2	Go to My APIs > statistics > API response time page.	• Only API1 resposetime should be available • API respose time of API1 should be less than or equals to the average time got from API1 jmeter invocation .		
3	Relogin as creator2 and go to My APIs > statistics > API response time page.	• Only the API2 responsetime should be available. • API response time of API2 should be less than oe equals to the average time got from API2 jmeter invocation.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.1.5.Test Suite : Last access time

Test Artifact
Mind Map

Test Case APIM-257: Check whether user can successfully view accurate last access times of each APIs. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User who is successfully logged into API publisher with login, creator permission should be able to view last access times of APIs accurately.				
Preconditions:				
1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively. 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. 5. Invoke API1 and API2 and note down lost access times of API1 and API2. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1, go to APIs > All Statistics > last access times and compare last access times with the real times you accessed API for the last time.	Last access time given for API1 and API2 should be similar to the last access times observed.		
2	Go to My APIs > statistics > Last access times	• Only the Last access time of API1 should be visible. • Last access time given in the system should match the last access time observed.		
3	Re-loggin as creator2 and go to My APIs > statistics > Last Access times page.	• Only the last access time of API2 should be listed. • Last access time given in the system should match the last access time observed.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-272: Check whether user can view statistics related to a particular date range successfully. [Version : 1]				
Author:	Sewmini Jayaweera			
Preconditions:				
1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively. 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. 5. Invoke API1 on day1 and API2 on day2 and note down lost access times of API1 and API2. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator2, go to APIs > All Statistics > last access times, set the date range to include day1 and compare last access times with the real times you accessed API for the last time.	Only the last access time of API1 should list.		
2	Change the date range to day 2 and check whether APIs last accessed on the particular date list accurately.	Only the API2 last access time should list.		

3	Go to My APIs > statistics > last access time and set date range only to include day1.	No records should be listed.		
4	Set date range to include both day1 and day2.	Last access time related to API2 should list.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

8.1.1.6.Test Suite : API Usage by Resource Path

Test Artifact	
Mind Map	

Test Case APIM-269: Check whether user can view API usage by resource paths took place during a particular date range [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User who is successfully logged in to API publisher with login and creator permission should be able to view API usage of each API by resource path accurately during a given date range.				
Preconditions:				
1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively (Both API end points point to JAX-RS web application hosted in wso2 app server). 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. 5. Send 10 GET requests to API1 and 100 GET requests to API2 on day1 6. Send 5 DELETE requests to API1 and 5 POST requests to API2 on day two. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1, go to APIs > All Statistics > API usage by resource path and specify date range which only include day1.	Table should only list only 10 GET requests to API1 and 100 GET requests to API2.		
2	Change date range selection in a way only day and Day2 are in the range.	Table should list • 10 access to API using GET method • 100 acess to API2 using GET method • 5 access to API1 using DELETE method and • 5 post access to API2 using POST method.		
3	Go to My APIs > statistics > API usage by resource path page and select date range in a way only day1 is included.	Only 10 GET requests to API1 should be listed in the table.		
4	Select date range in a way only day2 is included	Table should list 10 DELETE requests to API1		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-258: Check whether user can view API requests sent to different resource paths of APIs accurately. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User who is successfully logged in to API publisher with login and creator permission should be able to view API usage of each API by resource path accurately.				
Preconditions:				
1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be three two published APIs created by creator1 and creator2 as API1 and API2 respectively (Both API end points point to JAX-RS web application hosted in wso2 app server). 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:

1	Use a client and do below invocation. • Send 10 GET requests to API1 • Send 5 DELETE requests to API1 • Send 100 GET requests to API2 • send 5 POST requests to API2	API1 and API2 should get invoked successfully.		
2	Login to API publisher as creator1 and go to APIs > All Statistics > API usage by resource path.	All invocations did in the previous step should list accurately.		
3	Go to My APIs > statistics > API usage by resource path page	Only API1 relared invocations should be listed accurately.		
4	Re-login as creator2 and go to My APIs > statistics > API usage by resource path.	Only API2 invocations should be listed accurately.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

8.1.1.7.Test Suite : Usage by destination

Test Artifact
Mind Map

Test Case APIM-270: Check whether user can view API usage by destination relevant to a particular date range [Version : 1]

Author: Sewmini Jayaweera

Summary:

User who is successfully logged into the API publisher who has creator permission should be able to view API usage by destination statistics relavent to a selected date range.

Preconditions:

1. There should be a registered user who has login, creator permission.

2. There should be 2 wso2 Application servers having port offset 1,2 respectively.

3. There should be a published API (API1 created by creator1), given JAX-RX application end points hosted in app server 1 and app server2 as 2 load balanced endpoints. and 'Destination-Based Usage Tracking' should be enabled.

4. Subscriber1 should have successfully subscribed to API1.

5. API manager should be integrated with BAM by following [1].

6. Invoke API 100 times during the day one and monitor how many requests were served by each end point (end point given to AS1 and end point given to AS2). Say endpoint 1 was hit by 55 requests and end point 2 was hit by 45 requests on day1

7. Invoke API 200 times during the day2 and monitor how many requests were served by each end point. (say endpoint 1 was hit by 100 requests and end point 2 was hit by 100 requests on day2)

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
2	Login to API publisher as creator1 and go to APIs >All statistics > API usage by destination page. (date range should be selected in a way it only includes day1 and day2)	Table should list number of access accurately as below. • Number of access for end point1 should be 155 • Number of access of end point2 should be 145.		
3	Change the date range to include only day1.	Table should list number of access accurately as below. • Number of access for end point1 should be 55 • Number of access of end point2 should be 45.		
4	Change the date range to include only day2.	Table should list number of access accurately as below. • Number of access for end point1 should be 100 • Number of access of end point2 should be 100.		
5	Go to My API > statisctics > API usage by destination page. (date range should be selected in a way it only includes day1 and day2)	Table should list number of access accurately as below. • Number of access for end point1 should be 100 • Number of access of end point2 should be 100.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			

Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-267: Check whether user can view API usage by destination statistics successfully. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User who is successfully logged into the API publisher who has creator permission should be able to view API usage by destination statistics.

Preconditions:

1. There should be a registered user who has login, creator permission.

2. There should be 2 wso2 APPlication servers having port offset 1,2 respectively.

3. There should be a published API (API1), given JAX-RX application end points hosted in app server 1 and app server2 as 2 load balanced endpoints. and 'Destination-Based Usage Tracking' should be enabled.

4. Subscriber1 should have successfully subscribed to API1.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke API1 10 times and monitor per each end point how many requests were hit. (say end point1 - 5, end point2 -5)	In each application server log there should be 5 entried related to this invocation.		
2	Login to API publisher as creator1 and go to APIs >All statistics > API usage by destination page.	Table should list number of access accurately for each end point. (If you observe 5 requests per each endpoint table should list 5 in front of each end points)		
3	Go to My API > statisctics > API usage by destination page.	Table should list number of access accurately for each end point. (If you observe 5 requests per each endpoint table should list 5 in front of each end points)		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.1.8.Test Suite : Usage by user

Test Artifact
Mind Map

Test Case APIM-259: Check whether user can successfully view API usage of each API by user accurately. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User who is logged into API publisher with login , creator permissions should be able to view view API usage of each API by user.

Preconditions:

1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager.

2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively.

3. Version 2.0.0 of API1 should be published.

3. Subscriber1 should be subscribed to API1 and subscriber2 should be subscribed to API1 version 2.0.0 and API2

4. Subscriber1 should have invoked API1 100 times, Subscriber2 should have invoked API2 400 times and API1-2.0.0 100 times.

4. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to api publisher as API creator1 and go to APIs > All statistics > API usage by user page.	Table should look like below. API | version | user | Number of access API1 | 1.0.0 | subscriber1 | 100 API2 | 1.0.0 | subscriber2 | 400 API1 | 2.0.0 | subscriber2 | 100		
2	Go to My APIs > statistics > API usage by user page	Only the API1 related rows from the table shown in step1 should be visible.		

3	Relogin to API Publisher as creator2 and go to My APIs > statistics > API usage by user page.	Only the API2 related rows from the table shown in step1 should be visible.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-271: Check whether user can view API usage by user related to a particular date range. [Version : 1]

Author:	Sewmini Jayaweera			
Summary:	User who is successfully logged into API publisher with creator permission should be able to view API usage by user related to a particular date range.			
Preconditions:	1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively. 3. Version 2.0.0 of API1 should be published. 3. Subscriber1 should be subscribed to API1 and subscriber2 should be subscribed to API1 version 2.0.0 and API2 4. Subscriber1 should have invoked API1 100 times during day1, Subscriber2 should have invoked API2 400 times and API1-2.0.0 100 times during day2. 4. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics			
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login as creator1 to API publisher and go to APIs > all statistics > API usage by user page and set the date rage to day 1	Only the below record should be listed in the table. API1 1.0.0 subscriber1 100		
2	Change the date range to include only day2 and check whether correct statistics can be seen.	Following records should be available. API2 1.0.0 subscriber2 400 API1 2.0.0 subscriber2 100		
3	Set date rage to include both day1 and day2 and check whether correct statistics can be seen.	Following records should be available. API1 1.0.0 subscriber1 100 API2 1.0.0 subscriber2 400 API1 2.0.0 subscriber2 100		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

8.1.1.9.Test Suite : Faulty Invocation

Test Artifact
Mind Map

Test Case APIM-260: Check whether user can successfully view accurate faulty invocation statistics. [Version : 1]

Author:	Sewmini Jayaweera
Summary:	User who is successfully logged into API publihser with login and creator permission should be able to view API faulty invocation related statistics accurately.
Preconditions:	1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively. 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2.

4. API manager should be integrated with BAM by following [1].

5. API1 should be invoked 15 times successfully, API2 should be invoked 45 times successfully.

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 5 requests to API1 after stopping server which is API1's back end is hosted.	All 5 requests should fail with given " esponse code: 500 Internal Server Error"		
2	Send 5 requests to API2 after stopping server which is API2's back end is hosted.	All 5 requests should fail with given " esponse code: 500 Internal Server Error"		
3	Login to API publisher as creator1 and check whether failed requests are also updated in APIs > All statistics > faulty invocations page.	• A bar chart should be visible which depicts number of faulty invocations of API1 and API2. • The tabl must list accurate faulty invocation details related to APIs. • In this test number of faulty invocation count of API1 should be 5 and the percentage should be 25%. Number of fault count of API2 should be 5 and percentage should be 10% NOTE : calculation should be as below (# of faulty invocation)/((number of successfull invocations) + (number of faulty invocations))		
4	Go to My APIs > statistics > faulty invocations page.	Only API1 related statistics should be available. • Number of faulty calls should list as 5 and the percentage should be 25.		
5	Re-login as creator2 and Go to My APIs > statistics > faulty invocations page.	Only API1 related statistics should be visible. • Number of faulty calls should list as 5 and the percentage should be 10%		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-273: Check whether user can view correct faulty API related statistics for a selected date range. **[Version : 1]**

Author:	Sewmini Jayaweera

Summary:

Succeessfully logged in user to API publisher with permission should be able to view statistics relevant to a select date range.

Preconditions:
1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager.

2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively.

3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2.

4. API manager should be integrated with BAM by following [1].

5. API1 should be invoked 15 times successfully, API2 should be invoked 45 times successfully.

6. Send 5 requests to API1 after stopping server which is API1's back end is hosted during day1.

7. Send 5 requests to API2 after stopping server which is API2's back end is hosted during day2.

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1 and check whether failed requests are also updated in APIs > All statistics > faulty invocations page given date range as day1	Only faulty invocations related to API1 should dispay. Graph should show 25% as the percentage of faulty invocations related to API Table should list 5 as the number of faulty invocations.		
2	Change the date range to day2.	Only faulty invocations related to API2 should be able to see Number of fault count of API2 should be 5 and percentage should be 10%		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None

Keywords:	None
Last Result	**Not Run**

8.1.1.10.Test Suite : API store statistics

Test Artifact
Mind Map

Test Case APIM-261: Check whether API usage statistics per application gets updated accurately. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User who is successfully logged into API store who has Subscriber permission should be able to view Usage statistics of the Applications he/she owns accurately.				
Preconditions:				
1. There should be 2 users (subscriber1 and subscriber2) successfully registered in API store. 2. Subscriber1 should have two applications called default and APP1. 3. Subscriber 2 should have a default application. 4. Subscriber1 should subscribe to AP1 using both applications and to API2 using app1. 5. Subscriber2 should subscribed to API1 using Default application. 6. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Do the following invocations. • Invoke API1 10 times using default app of subscriber1 • Invoke API1 20 times using APP1 of subscriber1 • Invoke AP12 5 times using App1 of subscriber1 • Invoke API1 100 times using default app of subscriber2.	All the invocations should be successful.		
2	Login to API store using Subscriber1 and go to API store > statistics > API usage per application. page.	• There should be two pie charts related to APP1 and default application. • Pie chart of App1 should depict 80% as API1 usage and 20% as API2 usage. • Table which contains APP1 statistics should list 20 as no of api calls of API1 and 5 as the number of api calls of API2. • Pie chart of default application should depict 100% having number of api calls nade to API1 10.		
3	Relogin to Store as subscriber2 and go to statistics > API usage per application page.	• pie chart should depic percentage of API calls of default application as 100. • Table should list 100 as the number of API calls made to API1.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-262: Check whether Faulty invocation per application gets updated correctly . [Version : 1]				
Author:	Sewmini Jayaweera			
Summary:				
User who is successfully logged into API store should be able to view statistics os faulty invocations per application successfully.				
Preconditions:				
1. There should be a registered user (subscriber1) in API store. 2. Subscriber1 should have APP1 created. 3. Subscribe to API1 and API2 using app1. 4. API manager should be integrated with BAM by following [1]. 5. Down the backend server of APIs and do 2 faulty invocations for API1 and 4 faulty invocations for API2. (The error code 500 should return as the response of faulty invocations) [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API store as subscriber1 and go to statistics > faulty invocation per application page.	Table should list two entries related to API1 and API2 and the faulty api invocation count should be 2 and 4 for API1 and API2 respectively.		
Execution type:	Manual			
Estimated exec. duration (min):				

Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-263: Check whether user can successfully view api usage from resource path per application. [Version : 1]

Author: Sewmini Jayaweera

Summary:

User who is successfully logged into API store should be able to see API usage from resource path per application.

Preconditions:

1. There should be a registered user in API store (subscriber1)

2. There should be 2 published APIs (API1 and API2) which is back end is pointed to JAX-RS application hosted in wso2 application server.

3. Subscriber1 should have APP1 and APP2.

4. App1 has API1 and API2 subscriptions and APP2 has only API1 subscription.

4. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Do following invocations. • Send a Get request to API1 using App1 • Send a Delete request to API2 using App1 • Send a Get request to API1 using App2 • Send a DELETE request to API1 using App2 •	All requests should be sent successfully.		
2	login to API store as subscriber1 and go to statistics > API usage from resource path.	table should look like below. Application name | API name | API USAGE FROM RESOURCE PATH PER APPLICATION APP1 | API1 | /customerservice/customers/<id> (GET) | API2 | customerservice/customers/<id> (DELETE) APP2 | API1 | customerservice/customers/<id> (GET) | API1 | customerservice/customers/<id> (DELETE)		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-268: Check whether user can successfully view top users per applications. [Version : 1]

Author: Sewmini Jayaweera

Summary:

User who has successfully logged into to API manager store should be able to view top users per applications

Preconditions:

1. There should be three users who are registered in the API store. (subscriber1, rush, admin)

2. Subscriber1 should create two applications (App1 and App2).

3. Subscriber1 should subscribe to API1 and API2 using App1.

4. Subscribe1 should subscribe to API1 using App2.

5. User tokens should be generated for rash and admin for APP1.

6. Invoke APIs accroding to the below table.

App name	API	User	No of API calls

APP1	API1	admin	300
	API1	subscriber1	150
	API2	rash	50
App2	API1	subscriber1	10

7. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API store using subscriber1 and go to statistics > Top users per application page.	• There should be two charts under Registered Users For Applications, depicting users in app1 and app2. • App1 should show 3 (75%) users and app2 show 1 (25%) users. • Pie chart related to APP1 under 'Top Users For Applications' should depect percentages of API calls for admin, subscriber and rash as 60%, 30% and 10% respectively. • Table which contains information related app1 should accurately display api calls made by admin, subscriber1 and rash as 300, 150 and 50 respectively. • Pie chart related to APP1 under 'Top Users For Applications' should depect percentages of API calls for subscriber1 as 100% and the table related to app2 should show no of api calls made by subscriber1 as 10.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.2.Test Suite : Configure with with BAM from UI (1.9.0 <)

Test Artifact
Mind Map

8.1.2.1.Test Suite : permission based

Test Artifact
Mind Map

Test Case APIM-317: Check whether users logged into publisher who only has creator role can view both All Statistics and [Version : 1]

Author:	Shamin Goonetileke
Summary:	User who has successfully logged into the API publisher and who has Creator role assigned should be able to view APIs > All Statistics as well as My APIs > statistics tabs on the menu bar reside on the left side of the page.
Preconditions:	1. There should be a user who has only login and creator permission. 2. User should be successfully logged into API publisher. 3. API manager should be integrated with BAM by following the steps in [1] [1] https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Check whether APIs > All statistics and My APIs > statistics are available on side menu.	All statistics should be listed under APIs and statistics should be listed under 'My APIs'		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-318: Check whether users logged into publisher who only has publisher role cannot view "My APIs > statis [Version : 1]

Author:	Shamin Goonetileke
Summary:	User who has successfully logged into the API publisherand who has publisher role assigned should not be able to view statistics menu item listed under 'My APIs' but 'All Statistics' listed under 'APIs' .

Preconditions:

1. There should be a user who has only login and publish permission.

2. User should be successfully logged into API publisher.

3. API manager should be integrated with BAM by following the steps in [1]

[1] https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Check whether only the 'All Statistics' and 'Browse' tabs are available in the menu on the left side of the page.	Only only the 'All Statistics' and 'Browse' tabs should be listed under 'APIs'. 'My APIs' or any menu items under that should NOT be available.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.2.2.Test Suite : Subscription statistics

Test Artifact	
Mind Map	

Test Case APIM-319: Check whether system depicts correct subscription statistics to user who has creator permission when [Version : 1]

Author:	Shamin Goonetileke

Summary:

Each user who has login to API publisher and creator permission should be able to view correct statistics for API subscriptions under all statistics and My APIs.

Preconditions:

1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager.

2. There should be two published APIs created by creator1 (API1,API3) and another published API (API2) created by creator2

3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2 and API3.

4. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1 and go to 'Overall API Subscriptions' by clicking on APIs > All statistics > API subscriptions.	• Three graphs should be available (for API1, API2 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/3 % (33%) should depic in the graph created for API1 and 1/3% (33%) should depict in the graph created for API2 and 1/3% (33%) should depict in the graph created for API3.		
2	Then Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Only two graphs should be available (for API1 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/2 % (50%) should depic in the graph created for API1 and 1/2% (50%) should depict in the graph created for API3. NOTE : calculation should be as below. (# of subscriptions a perticular API has) / (Total number of subscriptions have for the APIs, created by the particular API creator)		
3	Log out and loggin to API publisher as Creator2 and go to APIs > All Statistics > API subscriptions.	• Three graphs should be available (for API1, API2 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/3 % (33%) should depic in the graph created for API1 and 1/3% (33%) should depict in the graph created for API2 and 1/3% (33%) should depict in the graph created for API3.		
4	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Only a graph should be available (for API2) having API name, creator and number of subscription listed below the graph. • In this scenario number of subscription should be 1 and graph should depic 100%.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-320: Check whether system depicts correct subscription statistics to user who has creator permission when [Version : 1]				
Author:	Shamin Goonetileke			
Summary: User who has successfully logged into API Publisher with login and creator permission should be able to view correct API subscription statistics when there are only APIs created by him with subscriptions.				
Preconditions: 1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be three APIs created by creator1. (API1, API2, API3) 3. There should be no subscriptions for API3, one subscription for API1 and 2 subscriptions for API2. 4. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1 and go to 'Overall API Subscriptions' by clicking on APIs > All statistics > API subscriptions.	• Two graphs should be available (Since only 2 of the APIs have subscriptions) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/3 % should depic in the graph created for API1 and 2/3 should depict in the graph created for API2.		
2	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Two graphs should be available (Since only 2 of the APIs have subscriptions) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/3 % (33%) should depic in the graph created for API1 and 2/3% (67%) should depict in the graph created for API2.		
3	Log out and loggin to API publisher as Creator2 and go to APIs > All Statistics > API subscriptions.	• Two graphs should be available (Since only 2 of the APIs have subscriptions) having API name, Creator and number of subscriptions below each graph. • In this scenario 1/3 % (33%) should depic in the graph created for API1 and 2/3% (67%) should depict in the graph created for API2.		
4	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	No data found or similar message should display on My API subscription page.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-321: Check whether Statistics get update properly once subscriber unsubscribe from the API. [Version : 1]				
Author:	Shamin Goonetileke			
Summary: User who has permission should be able to see updated subscription statistics accurately when an existing subscriber unsubscribe from an API.				
Preconditions: 1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 (API1,API3) and another published API (API2) created by creator2 3. There should be two subscribers (subscriber1 and subscriber2) both have subscriptions for API1, API2 AND API3. 4. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1 and go to 'Overall API Subscriptions' by clicking on APIs > All statistics > API subscriptions.	• Three graphs should be available (for API1, API2 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 2/6 % (33%) should depic in the graph created for API1 and 2/6% (33%) should depict in the graph created for API2 and 2/6% (33%) should depict in the graph created for API3.		
2	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Only two graphs should be available (for API1 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 2/4 % (50%) should depic in the graph created for API1 and 2/4% (50%) should depict in the graph created for API3. NOTE : calculation should be as below. (# of subscriptions a perticular API has) / (Total number of subscriptions have for the APIs, created by the particular API creator)		

3	Log out and loggin to API publisher as Creator2 and go to APIs > All Statistics > API subscriptions.	• Three graphs should be available (for API1, API2 and API3) having API name, Creator and number of subscriptions below each graph. • In this scenario 2/6 % (33%) should depic in the graph created for API1 and 2/6% (33%) should depict in the graph created for API2 and 2/6% (33%) should depict in the graph created for API3.		
4	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Only a graph should be available (for API2) having API name, creator and number of subscription listed below the graph. • In this scenario number of subscription should be 2 and graph should depic 100%.		
5	Login as subscriber1 and unsubscribe API1 subscription.	Subscriber1 should unsubscribe from API1 successfully.		
6	Wait for 2-3 mins, login to API publisher as creator1 and go to 'Overall API Subscriptions' by clicking on APIs > All statistics > API subscriptions.	Change should reflect as below. • Statistics of API 1subscription should decrease to 20% (1/5%) • Statistics of API2 subscription should increase to 2/5% (40%) • Statistics of API3 subscription should increase to 2/5 (40%)		
7	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	Subscription statistics of API1 and API3 should be 33% and 67% respectively.		
8	Log out and loggin to API publisher as Creator2 and go to APIs > All Statistics > API subscriptions.	Change should reflect as below. • Statistics of API 1subscription should decrease to 20% (1/5%) • Statistics of API2 subscription should increase to 2/5% (40%) • Statistics of API3 subscription should increase to 2/5 (40%)		
9	Click on My APIs > Statistics > API subscriptions link and go to 'Overall API Subscriptions' page	• Only a graph should be available (for API2) having API name, creator and number of subscription listed below the graph. • In this scenario number of subscription should be 2 and graph should depic 100%.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-322: Check whether user can successfully view subscription statistics relevant to a given time duration. **[Version : 1]**

Author:	Shamin Goonetileke

Summary:

User who is successfully logged into the API publisher who has login, creator permission should be able to view API subscription statistics relevant to a time duration specified by the user.

Preconditions:

1. There should be a user (creator1) having login and creator permission, registered in API manager.

2. There should be two published APIs created by creator1 (API1,API2).

3. There should be a subscriber (subscriber1) subscribed to AP11 and API2 on day1, and another subscriber (subscriber2) subscribed to API2 on day2.

4. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1 and go to 'Overall API Subscriptions' by clicking on APIs > All statistics > API subscriptions.	User should successfully get redirect to the "Overall API Subscriptions (Across All Versions)" and API subscription statistics relavent to the default time duration chosen should be available.		
2	Set time duration in a way it only includes day one.	• Two graphs should be available (for API1, API2) having API name, Creator and number of subscriptions below each graph. • Statistics of API 1subscription should decrease to 50% • Statistics of API2 subscription should increase to 50%		
3	Reset time duration in a way it only includes day two.	• A graphs should be available (forAPI2) having API name, Creator and number of subscriptions below each graph. • Statistics of API 1subscription should decrease to 100%		
4	Go to My APIs > statistics > API subscriptions and set time duration in a way it only includes day1	• Two graphs should be available (for API1, API2) having API name, Creator and number of subscriptions below each graph. • Statistics of API 1subscription should decrease to 50% • Statistics of API2 subscription should increase to 50%		
5	Reset time duration in a way it only includes day two.	• A graphs should be available (forAPI2) having API name, Creator and number of subscriptions below each graph. • Statistics of API 1subscription should decrease to 100%		
6	Reset time duration in a way it includes both day1 and day2	• Two graphs should be available (for API1, API2) having API name, Creator and number of subscriptions below each graph. • Statistics of API 1subscription should decrease to 1/3% • Statistics of API2 subscription should increase to 2/3%		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.2.3.Test Suite : Usage statistics

Test Artifact
Mind Map

Test Case APIM-323: Check whether User can successfully view API usage statistics relevant to a particular time duration [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
User who has successfully logged into API Publisher should be able to view API usage statistics of the APIs based on the				
Preconditions:				
1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be three two published APIs created by creator1 and creator2 as API1 and API2 respectively. 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. 5. API1 and API2 should be invoked 10 times each on during day 1. 6. API1 and API2 should be invoked 15 times and 5 times respectively. [1] https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher using creator1 and go to APIs > All statistics > API usage page and set the date range to include only day1.	• The pie chart should depic api invocation percentage related to API1 and API2 as 50% each. • Table should list API calls related to API1 as 10 and API calls related API2 as 10.		
2	Reset date range to include both day1 and day2.	• Pie chart should depict API invocation percentage related to API1 as 25/40 % • Pie chart should depict API invocation percentage related to API2 as 15/40 % • Table will list number of api calls related to API1 as 25 and number of API calls related to API2 as 15.		
3	Reset date range to include only day2 as the date range.	• The pie chart should depic api invocation percentage related to API1 as 75% and API2 as 25% each. • Table should list API calls related to API1 as 15 and API calls related API2 as 5.		
4	Go to MyAPIs > statistics > API usage page and select date range to inlude only the day1.	• Pie chart should depict API call percentage as 100% • When hover API name will Appear as API1 and number of calls will appear as 10 • In the able only API1 will be listed having number of calls as 10.		
5	Re-login to API publisher creator2, go to APIs > All statistics > API usage page and set the date range to include only day1.	• The pie chart should depic api invocation percentage related to API1 and API2 as 50% each. • Table should list API calls related to API1 as 10 and API calls related API2 as 10.		
6	Reset date range to include both day1 and day2.	• The pie chart should depic api invocation percentage related to API1 as 75% and API2 as 25% each. • Table should list API calls related to API1 as 15 and API calls related API2 as 5.		
7	Go to MyAPIs > statistics > API usage page and select date range to inlude only the day2.	• Pie chart should depict API call percentage as 100% • When hover API name will Appear as API2 and number of calls will appear as 5 • In the able only API2 will be listed having number of calls as 5.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				

Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-324: Check whether API manager statistics are updated accurately when another version of the same API is [Version : 1]

Author:	Shamin Goonetileke

Summary:

When user has invoked two versions of the same API, the total number of invocations should list under API usage statistics of the particular API correctly.

Preconditions:

1. Execute test case 252

2. Get a copy of API1 (API1 2.0.0) publish it

2. Subscribe to 2.0.0 version of API1 and invoke it 10 times.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher using creator1 and go to APIs > All statistics > API usage page.	• There should be a pie chart which depicts API usage. • There should be a table which contains API name and number of API calls. • Number of API calls relevant to API1 should be listed as 15 • Number of API calls relevant to API2 should be listed as 10 • Pie chart should depic this information accurately by showing 60% for API1 and 40% for API2.		
2	Go to My APIs > statistics > API usage page.	• Pie chart should depic Number of API calls to API1 as 100% (15 calls) • Table should list only API1 and number of calls related to API1 as 15.		
3	Logout and login as creator2 and go to APIs > All statistics > API usage page.	• There should be a pie chart which depicts API usage. • There should be a table which contains API name and number of API calls. • Number of API calls relevant to API1 should be listed as 15 • Number of API calls relevant to API2 should be listed as 10 • Pie chart should depic this information accurately by showing 60% for API1 and 40% for API2.		
4	Go to My APIs > statistics > API usage page.	• Pie chart should depic percentage of API calls related to API2 as 100% • When hover the pie chart it should give name of the API as API2 and number of calls as 10. • Table should list API2 and number of API calls as 10.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-325: Check whether API creator can successfully view API usage statistics of API created by him as well a [Version : 1]

Author:	Shamin Goonetileke

Summary:

User who has successfully logged into API Publisher should be able to view API usage statistics of the APIs create by the particular user as well as usage statistics of all the APIs available in API manager.

Preconditions:

1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager.

2. There should be three two published APIs created by creator1 and creator2 as API1 and API2 respectively.

3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2.

4. API manager should be integrated with BAM by following [1].

5. Subscriber1 should have invoked API1 5 times successfully.

6. Subscriber2 should have invoked API2 10 times successfully.

[1] https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher using creator1 and go to APIs > All statistics > API usage page.	• There should be a pie chart which depicts API usage. • There should be a table which contains API name and number of API calls. • Number of API calls relevant to API1 should be listed as 5 • Number of API calls relevant to API2 should be listed as 10 • Pie chart should depic this information accurately by showing 33.3% for API1 and 66.7% for API2.		

2	Go to My APIs > statistics > API usage page.	• Pie chart should depic Number of API calls to API1 as 100% (5 calls) • Table should list only API1 and number of calls related to API1 as 5.		
3	Logout and login as creator2 and go to APIs > All statistics > API usage page.	• There should be a pie chart which depicts API usage. • There should be a table which contains API name and number of API calls. • Number of API calls relevant to API1 should be listed as 5 • Number of API calls relevant to API2 should be listed as 10 • Pie chart should depic this information accurately by showing 33.3% for API1 and 66.7% for API2.		
4	Go to My APIs > statistics > API usage page.	• Pie chart should depic percentage of API calls related to API2 as 100% • When hover the pie chart it should give name of the API as API2 and number of calls as 10. • Table should list API2 and number of API calls as 10.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-326: Check whether Usage statistics gets updated accurately when a particular API is deleted from API man [Version : 1]				
Author:	Shamin Goonetileke			
Summary: Once an API is deleted from API manager usage data relevant to the particular API should not be available in the API usage statistics.				
Preconditions: 1. Execute 252 test case.				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Unsubscribe all subscriptions which API2 has and delete API2.	API2 should get deleted successfully.		
2	Login to the publisher as creator1 and go to APIs > all statistics > API usage page.	• Pie chart should depic percentage API calls as 100% • Ehen hovering it should display API name as API1 and 5 call(s). • In the table only API1 should listed and the number of API calls should display as 5.		
3	Go to My APIs > statistics > API usage page.	• Pie chart should depic percentage API calls as 100% • Ehen hovering it should display API name as API1 and 5 call(s). • In the table only API1 should listed and the number of API calls should display as 5.		
4	Logout and login as creator2 and go to APIs > All Statistics > API usage page.	Page should display 'No data found' or similar message.		
5	Go to My APIs > statistics > API usage page.	Page should display 'No data found' or similar message.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

8.1.2.4.Test Suite : response time

Test Artifact
Mind Map

Test Case APIM-327: Check whether the API response time statistics are accurate. [Version : 1]
Author: Shamin Goonetileke
Summary: User should be able to successfully view the average API response time of APIs.
Preconditions: 1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be three two published APIs created by creator1 and creator2 as API1 and API2 respectively.

3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2.

4. API manager should be integrated with BAM by following [1].

5. Invoke API1 100 times using Jmeter and get average latancy using "Aggregate Report".

6. Invoke API2 100 times using Jmeter and get average latancy using "Aggregate Report".

[1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher using creator1 and go to APIs > All statistics > API response time page.	• API respose time of API1 should be less than or equals to the average time got from API1 jmeter invocation . • API response time of API2 should be less than oe equals to the average time got from API2 jmeter invocation.		
2	Go to My APIs > statistics > API response time page.	• Only API1 resposetime should be available • API respose time of API1 should be less than or equals to the average time got from API1 jmeter invocation .		
3	Relogin as creator2 and go to My APIs > statistics > API response time page.	• Only the API2 responsetime should be available. • API response time of API2 should be less than oe equals to the average time got from API2 jmeter invocation.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.2.5.Test Suite : Last access time

Test Artifact
Mind Map

Test Case APIM-328: Check whether user can successfully view accurate last access times of each APIs. [Version : 1]

Author:	Shamin Goonetileke
Summary:	
User who is successfully logged into API publisher with login, creator permission should be able to view last access times of APIs accurately.	
Preconditions:	
1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager.	
2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively.	
3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2.	
4. API manager should be integrated with BAM by following [1].	
5. Invoke API1 and API2 and note down lost access times of API1 and API2.	
[1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1, go to APIs > All Statistics > last access times and compare last access times with the real times you accessed API for the last time.	Last access time given for API1 and API2 should be similar to the last access times observed.		
2	Go to My APIs > statistics > Last access times	• Only the Last access time of API1 should be visible. • Last access time given in the system should match the last access time observed.		
3	Re-loggin as creator2 and go to My APIs > statistics > Last Access times page.	• Only the last access time of API2 should be listed. • Last access time given in the system should match the last access time observed.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-329: Check whether user can view statistics related to a particular date range successfully. [Version : 1]				
Author:	Shamin Goonetileke			
Preconditions: 1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively. 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. 5. Invoke API1 on day1 and API2 on day2 and note down lost access times of API1 and API2. [1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator2, go to APIs > All Statistics > last access times, set the date range to include day1 and compare last access times with the real times you accessed API for the last time.	Only the last access time of API1 should list.		
2	Change the date range to day 2 and check whether APIs last accessed on the particular date list accurately.	Only the API2 last access time should list.		
3	Go to My APIs > statistics > last access time and set date range only to include day1.	No records should be listed.		
4	Set date range to include both day1 and day2.	Last access time related to API2 should list.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

8.1.2.6.Test Suite : API Usage by Resource Path

Test Artifact
Mind Map

Test Case APIM-330: Check whether user can view API usage by resource paths took place during a particular date range [Version : 1]				
Author:	Shamin Goonetileke			
Summary: User who is successfully logged in to API publisher with login and creator permission should be able to view API usage of each API by resource path accurately during a given date range.				
Preconditions: 1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively (Both API end points point to JAX-RS web application hosted in wso2 app server). 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. 5. Send 10 GET requests to API1 and 100 GET requests to API2 on day1 6. Send 5 DELETE requests to API1 and 5 POST requests to API2 on day two. [1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as creator1, go to APIs > All Statistics > API usage by resource path and specify date range which only include day1.	Table should only list only 10 GET requests to API1 and 100 GET requests to API2.		
2	Change date range selection in a way only day and Day2 are in the range.	Table should list • 10 access to API using GET method • 100 acess to API2 using GET method • 5 access to API1 using DELETE method and • 5 post access to API2 using POST method.		
3	Go to My APIs > statistics > API usage by resource path page and select date range in a way only day1 is included.	Only 10 GET requests to API1 should be listed in the table.		
4	Select date range in a way only day2 is included	Table should list 10 DELETE requests to API1		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			

Last Result	**Not Run**

Test Case APIM-331: Check whether user can view API requests sent to different resource paths of APIs accurately. [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
User who is successfully logged in to API publisher with login and creator permission should be able to view API usage of each API by resource path accurately.				
Preconditions:				
1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be three two published APIs created by creator1 and creator2 as API1 and API2 respectively (Both API end points point to JAX-RS web application hosted in wso2 app server). 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Use a client and do below invocation. • Send 10 GET requests to API1 • Send 5 DELETE requests to API1 • Send 100 GET requests to API2 • send 5 POST requests to API2	API1 and API2 should get invoked successfully.		
2	Login to API publisher as creator1 and go to APIs > All Statistics > API usage by resource path.	All invocations did in the previous step should list accurately.		
3	Go to My APIs > statistics > API usage by resource path page	Only API1 relared invocations should be listed accurately.		
4	Re-login as creator2 and go to My APIs > statistics > API usage by resource path.	Only API2 invocations should be listed accurately.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

8.1.2.7.Test Suite : Usage by destination

Test Artifact	
Mind Map	

Test Case APIM-332: Check whether user can view API usage by destination relevant to a particular date range [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
User who is successfully logged into the API publisher who has creator permission should be able to view API usage by destination statistics relavent to a selected date range.				
Preconditions:				
1. There should be a registered user who has login, creator permission. 2. There should be 2 wso2 Application servers having port offset 1,2 respectively. 3. There should be a published API (API1 created by creator1), given JAX-RX application end points hosted in app server 1 and app server2 as 2 load balanced endpoints. and 'Destination-Based Usage Tracking' should be enabled. 4. Subscriber1 should have successfully subscribed to API1. 5. API manager should be integrated with BAM by following [1]. 6. Invoke API 100 times during the day one and monitor how many requests were served by each end point (end point given to AS1 and end point given to AS2). Say endpoint 1 was hit by 55 requests and end point 2 was hit by 45 requests on day1 7. Invoke API 200 times during the day2 and monitor how many requests were served by each end point. (say endpoint 1 was hit by 100 requests and end point 2 was hit by 100 requests on day2) [1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
2	Login to API publisher as creator1 and go to APIs >All statistics > API usage by destination page. (date range should be selected in a way it only includes day1 and day2)	Table should list number of access accurately as below. • Number of access for end point1 should be 155 • Number of access of end point2 should be 145.		
3	Change the date range to include only day1.	Table should list number of access accurately as below. • Number of access for end point1 should be 55 • Number of access of end		

		point2 should be 45.		
4	Change the date range to include only day2.	Table should list number of access accurately as below. • Number of access for end point1 should be 100 • Number of access of end point2 should be 100.		
5	Go to My API > statisctics > API usage by destination page. (date range should be selected in a way it only includes day1 and day2)	Table should list number of access accurately as below. • Number of access for end point1 should be 100 • Number of access of end point2 should be 100.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-333: Check whether user can view API usage by destination statistics successfully. [Version : 1]

Author:	Shamin Goonetileke

Summary:

User who is successfully logged into the API publisher who has creator permission should be able to view API usage by destination statistics.

Preconditions:

1. There should be a registered user who has login, creator permission.

2. There should be 2 wso2 APPlication servers having port offset 1,2 respectively.

3. There should be a published API (API1), given JAX-RX application end points hosted in app server 1 and app server2 as 2 load balanced endpoints. and 'Destination-Based Usage Tracking' should be enabled.

4. Subscriber1 should have successfully subscribed to API1.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke API1 10 times and monitor per each end point how many requests were hit. (say end point1 - 5, end point2 -5)	In each application server log there should be 5 entried related to this invocation.		
2	Login to API publisher as creator1 and go to APIs >All statistics > API usage by destination page.	Table should list number of access accurately for each end point. (If you observe 5 requests per each endpoint table should list 5 in front of each end points)		
3	Go to My API > statisctics > API usage by destination page.	Table should list number of access accurately for each end point. (If you observe 5 requests per each endpoint table should list 5 in front of each end points)		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.2.8.Test Suite : Usage by user

Test Artifact	
Mind Map	

Test Case APIM-334: Check whether user can successfully view API usage of each API by user accurately. [Version : 1]

Author:	Shamin Goonetileke

Summary:

User who is logged into API publisher with login , creator permissions should be able to view view API usage of each API by user.

Preconditions:

1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager.

2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively.

3. Version 2.0.0 of API1 should be published.

3. Subscriber1 should be subscribed to API1 and subscriber2 should be subscribed to API1 version 2.0.0 and API2

4. Subscriber1 should have invoked API1 100 times, Subscriber2 should have invoked API2 400 times and API1-2.0.0 100 times.

4. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to api publisher as API creator1 and go to APIs > All statistics > API usage by user page.	Table should look like below. API | version | user | Number of access API1 | 1.0.0 | subscriber1 | 100 API2 | 1.0.0 | subscriber2 | 400 API1 | 2.0.0 | subscriber2 | 100		
2	Go to My APIs > statistics > API usage by user page	Only the API1 related rows from the table shown in step1 should be visible.		
3	Relogin to API Publisher as creator2 and go to My APIs > statistics > API usage by user page.	Only the API2 related rows from the table shown in step1 should be visible.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-335: Check whether user can view API usage by user related to a particular date range. [Version : 1]

Author:	Shamin Goonetileke

Summary:

User who is successfully logged into API publisher with creator permission should be able to view API usage by user related to a particular date range.

Preconditions:
1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager.

2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively.

3. Version 2.0.0 of API1 should be published.

3. Subscriber1 should be subscribed to API1 and subscriber2 should be subscribed to API1 version 2.0.0 and API2

4. Subscriber1 should have invoked API1 100 times during day1, Subscriber2 should have invoked API2 400 times and API1-2.0.0 100 times during day2.

4. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login as creator1 to API publisher and go to APIs > all statistics > API usage by user page and set the date rage to day 1	Only the below record should be listed in the table. API1 1.0.0 subscriber1 100		
2	Change the date range to include only day2 and check whether correct statistics can be seen.	Following records should be available. API2 | 1.0.0 | subscriber2 | 400 API1 | 2.0.0 | subscriber2 | 100		
3	Set date rage to include both day1 and day2 and check whether correct statistics can be seen.	Following records should be available. API1 | 1.0.0 | subscriber1 | 100 API2 | 1.0.0 | subscriber2 | 400 API1 | 2.0.0 | subscriber2 | 100		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None

Last Result	**Not Run**

8.1.2.9.Test Suite : Faulty Invocation

Test Artifact
Mind Map

Test Case APIM-338: Check whether user can successfully view accurate faulty invocation statistics. [Version : 1]				
Author:	Shamin Goonetileke			
Summary: User who is successfully logged into API publihser with login and creator permission should be able to view API faulty invocation related statistics accurately.				
Preconditions: 1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively. 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. 5. API1 should be invoked 15 times successfully, API2 should be invoked 45 times successfully. [1].https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Send 5 requests to API1 after stopping server which is API1's back end is hosted.	All 5 requests should fail with given " response code: 500 Internal Server Error"		
2	Send 5 requests to API2 after stopping server which is API2's back end is hosted.	All 5 requests should fail with given " response code: 500 Internal Server Error"		
3	Login to API publisher as creator1 and check whether failed requests are also updated in APIs > All statistics > faulty invocations page.	• A bar chart should be visible which depicts number of faulty invocations of API1 and API2. • The tabl must list accurate faulty invocation details related to APIs. • In this test number of faulty invocation count of API1 should be 5 and the percentage should be 25%. Number of fault count of API2 should be 5 and percentage should be 10% NOTE : calculation should be as below (# of faulty invocation)/((number of successfull invocations) + (number of faulty invocations))		
4	Go to My APIs > statistics > faulty invocations page.	Only API1 related statistics should be available. • Number of faulty calls should list as 5 and the percentage should be 25.		
5	Re-login as creator2 and Go to My APIs > statistics > faulty invocations page.	Only API1 related statistics should be visible. • Number of faulty calls should list as 5 and the percentage should be 10%		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-339: Check whether user can view correct faulty API related statistics for a selected date range. [Version : 1]				
Author:	Shamin Goonetileke			
Summary: Succeessfully logged in user to API publisher with permission should be able to view statistics relevant to a select date range.				
Preconditions: 1. There should be two users (creator1, creator2), having login and creator permission, registered in API manager. 2. There should be two published APIs created by creator1 and creator2 as API1 and API2 respectively. 3. There should be a subscriber (subscriber1) subscribed to AP11 and another subscriber (subscriber2) subscribed to API2. 4. API manager should be integrated with BAM by following [1]. 5. API1 should be invoked 15 times successfully, API2 should be invoked 45 times successfully. 6. Send 5 requests to API1 after stopping server which is API1's back end is hosted during day1. 7. Send 5 requests to API2 after stopping server which is API2's back end is hosted during day2. [1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:

1	Login to API publisher as creator1 and check whether failed requests are also updated in APIs > All statistics > faulty invocations page given date range as day1	Only faulty invocations related to API1 should dispay. Graph should show 25% as the percentage of faulty invocations related to API Table should list 5 as the number of faulty invocations.		
2	Change the date range to day2.	Only faulty invocations related to API2 should be able to see Number of fault count of API2 should be 5 and percentage should be 10%		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.2.10.Test Suite : API store statistics

Test Artifact
Mind Map

Test Case APIM-340: Check whether API usage statistics per application gets updated accurately. **[Version : 1]**

Author:	Shamin Goonetileke
Summary:	
User who is successfully logged into API store who has Subscriber permission should be able to view Usage statistics of the Applications he/she owns accurately.	
Preconditions:	
1. There should be 2 users (subscriber1 and subscriber2) successfully registered in API store. 2. Subscriber1 should have two applications called default and APP1. 3. Subscriber 2 should have a default application. 4. Subscriber1 should subscribe to AP1 using both applications and to API2 using app1. 5. Subscriber2 should subscribed to API1 using Default application. 6. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Do the following invocations. • Invoke API1 10 times using default app of subscriber1 • Invoke API1 20 times using APP1 of subscriber1 • Invoke AP12 5 times using App1 of subscriber1 • Invoke API1 100 times using default app of subscriber2.	All the invocations should be successful.		
2	Login to API store using Subscriber1 and go to API store > statistics > API usage per application. page.	• There should be two pie charts related to APP1 and default application. • Pie chart of App1 should depict 80% as API1 usage and 20% as API2 usage. • Table which contains APP1 statistics should list 20 as no of api calls of API1 and 5 as the number of api calls of API2. • Pie chart of default application should depict 100% having number of api calls nade to API1 10.		
3	Relogin to Store as subscriber2 and go to statistics > API usage per application page.	• pie chart should depic percentage of API calls of default application as 100. • Table should list 100 as the number of API calls made to API1.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-341: Check whether Faulty invocation per application gets updated correctly . [Version : 1]

Author:	Shamin Goonetileke
Summary:	User who is successfully logged into API store should be able to view statistics os faulty invocations per application successfully.
Preconditions:	1. There should be a registered user (subscriber1) in API store. 2. Subscriber1 should have APP1 created. 3. Subscribe to API1 and API2 using app1. 4. API manager should be integrated with BAM by following [1]. 5. Down the backend server of APIs and do 2 faulty invocations for API1 and 4 faulty invocations for API2. (The error code 500 should return as the response of faulty invocations) [1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API store as subscriber1 and go to statistics > faulty invocation per application page.	Table should list two entries related to API1 and API2 and the faulty api invocation count should be 2 and 4 for API1 and API2 respectively.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-342: Check whether user can successfully view api usage from resource path per application. [Version : 1]

Author:	Shamin Goonetileke
Summary:	User who is successfully logged into API store should be able to see API usage from resource path per application.
Preconditions:	1. There should be a registered user in API store (subscriber1) 2. There should be 2 published APIs (API1 and API2) which is back end is pointed to JAX-RS application hosted in wso2 application server. 3. Subscriber1 should have APP1 and APP2. 4. App1 has API1 and API2 subscriptions and APP2 has only API1 subscription. 4. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Do following invocations. • Send a Get request to API1 using App1 • Send a Delete request to API2 using App1 • Send a Get request to API1 using App2 • Send a DELETE request to API1 using App2 •	All requests should be sent successfully.		
2	login to API store as subscriber1 and go to statistics > API usage from resource path.	table should look like below. Application name | API name | API USAGE FROM RESOURCE PATH PER APPLICATION APP1 | API1 | /customerservice/customers/<id> (GET) | API2 | customerservice/customers/<id> (DELETE) APP2 | API1 | customerservice/customers/<id> (GET) | API1 | customerservice/customers/<id> (DELETE)		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None

Last Result	**Not Run**

Test Case APIM-343: Check whether user can successfully view top users per applications. [Version : 1]					
Author:	Shamin Goonetileke				
Summary:					
User who has successfully logged into to API manager store should be able to view top users per applications					
Preconditions:					
1. There should be three users who are registered in the API store. (subscriber1, rush, admin) 2. Subscriber1 should create two applications (App1 and App2). 3. Subscriber1 should subscribe to API1 and API2 using App1. 4. Subscribe1 should subscribe to API1 using App2. 5. User tokens should be generated for rash and admin for APP1. 6. Invoke APIs accroding to the below table.					

App name	API	User	No of API calls
APP1	API1	admin	300
	API1	subscriber1	150
	API2	rash	50
App2	API1	subscriber1	10

7. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API store using subscriber1 and go to statistics > Top users per application page.	• There should be two charts under Registered Users For Applications, depicting users in app1 and app2. • App1 should show 3 (75%) users and app2 show 1 (25%) users. • Pie chart related to APP1 under 'Top Users For Applications' should depect percentages of API calls for admin, subscriber and rash as 60%, 30% and 10% respectively. • Table which contains information related app1 should accurately display api calls made by admin, subscriber1 and rash as 300, 150 and 50 respectively. • Pie chart related to APP1 under 'Top Users For Applications' should depect percentages of API calls for subscriber1 as 100% and the table related to app2 should show no of api calls made by subscriber1 as 10.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

8.1.2.11.Test Suite : Stats available for publisher

Test Artifact
Mind Map

Test Case APIM-344: Check whether publisher can view API usage of the particular API by versions. [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
User who is successfully logged into api publisher should be able to view API usage of the particular API by versions.				
Preconditions:				
1. User with publisher permission should be successfully registered in API manager. 2. There should be 2 versions of the same API (API 1.0.0 , API 2.0.0). 3. API version 1.0.0 should be invoked 100 times by different applications. 4. API version 2.0.0 should be invoked 25 times by different applications. 5. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher and click on one of the versions of API1 > versions .	API usage by verion pie chart should depict version 2.0.0 as 20% and version 1.0.0 as 80% and the table should list the number of invocations for API1 and API2 as 100 and 25 respectively.		
2	Check version statisctics for both API versions.	Results should be same for both.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-345: Check whether user can successfully view statistics of API subscriptions by versions. [Version : 1]

Author:	Shamin Goonetileke

Summary:

User who is successfully logged in to API publisher should be able to view statistics of API subscriptions by versions successfully.

Preconditions:

1. User with publisher permission should successfully registered in API manager.

2. There should two versions published from the same API. (API version 1.0.0 and API version 2.0.0)

3. Subscriber1, subscriber2, subscriber3 should get subscribe to API version 1.0.0

4. Subscriber4 should get subscribe to API version 2.0.0.

5. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as Publisher1 and go to API - 1.0.0 > versions.	API Subscriptions by Versions pie chart should depict 75% as subscriptions for API version 1.0.0 and 25 % for API version 2.0.0. Number of subscriptions for version 1.0.0 should list as 3 and number of subscriptions API version 2.0.0 has should list as 1.		
2	Unsubscribe subscriber 3 from API version 1.0.0 and go to API > version page.	API Subscriptions by Versions pie chart should depict 66.7% as subscriptions for API version 1.0.0 and 33.3 % for API version 2.0.0. Number of subscriptions for version 1.0.0 should be 2 Number of subscriptions for version 2.0.0 should be 1.		
3	Subscribe to API version 2.0.0 using subscriber 3 and got to version tab of api version 2.0.0.	API Subscriptions by Versions pie chart should depict 50% as subscriptions for API version 1.0.0 and 50 % for API version 2.0.0. Number of subscriptions for version 1.0.0 should be 2 Number of subscriptions for version 2.0.0 should be 2.		
4	Check statistics of API subscriptions by versions using both API versions. (API - 1.0.0 and API - 2.0.0)	Statisctics should be the same irrespective of API version.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-346: Check whether user can successfully view statistics of API Usage by Current Subscribers. [Version : 1]

Author:	Shamin Goonetileke

Summary:

User who is successfully logged into API publisher should be able to view statistics of API Usage by Current Subscribers.

Preconditions:

1. There should be a user (publisher1), having publisher permission, registered in API manager.

2. There should be two versions of the same API (API1) published.

3. Subscriptions and invocations relavent to API versions should be as below.

API version	Subscriber	Number of API calls.
1.0.0	subscriber1	10
1.0.0	subscriber2	20
1.0.0	subscriber3	20
2.0.0	Subscriber1	30

4. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as publisher1 go to browse > API1 v1.0.0 > users page.	• 'Usage by Current Subscribers (v-1.0.0)' pie chart should show percentage of api calls made by each subscriber as subscriber1 - 20% , subscriber2 - 40% , subscriber 3 -40%. • Also there should be a list having number of api calls made by each subscriber as subscriber1 - 10 , subscriber2- 20, subscriber3 - 20. • 'Usage by Current Subscribers (Across All Versions)' Pie chart should show API calls made across all versions as subscriber1 - 50 %, subscriber2 - 25%, subscriber3 - 25%. • Table related to the pie chart should show numbsr of api calls made by subscriber1 - 40 , subscriber2 - 20, subscriber3 - 20.		
2	Go to browse > API1 v2.0.0 > users page.	• Usage by Current Subscribers (v-2.0.0) pie chart should show percentage as 100% and table should list subscriber1 with 30 api calls. • 'Usage by Current Subscribers (Across All Versions)' Pie chart should show API calls made across all versions as subscriber1 - 50 %, subscriber2 - 25%, subscriber3 - 25%. • Table related to the pie chart should show numbsr of api calls made by subscriber1 - 40 , subscriber2 - 20, subscriber3 - 20.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.1.2.12.Test Suite : Configure through admin dashboard

Test Artifact	
Mind Map	

Test Case APIM-408: Successfully logged user should be configure to BAM through admin dashboard [Version : 1]				
Author:	Shamin Goonetileke			
Summary:				
User who has successfully logged into admin dashboard (https://<Server Host>:9443/admin-dashboard) should be able to configure BAM through admin dashboard				
Preconditions:				
1) API manager should be integrated with BAM by following the steps in [1] [1] https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User logged into admin dashboard using valid user name and password	User should be successfully log into the admin dashboard		
2	Click the Configure Analytics menu	Configure analytics menu should be successfully appeared		
3	Select the Enable check-box to enable statistical data publishing	Text fields should be appeared after enable		
4	User gives correct event receiver configurations and press add url group	url group should be added successfully		
5	User gives data Analyzer Configurations	configurations should be added successfully		
6	User fills statistics summary datasource (url,jdbc class, username,password) User gives more options if needed (max active, max wait)	configuration should be added successfully		
7	User clicks save button	configuration should be saved successfully		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-409: User saved without adding URL group configure to BAM through admin dashboard [Version : 1]	
Author:	Shamin Goonetileke
Summary:	
User who has successfully logged into admin dashboard (https://<Server Host>:9443/admin-dashboard) should be able to configure BAM through admin dashboard try to save without adding URL group	

Preconditions:

1) API manager should be integrated with BAM by following the steps in [1]

[1] https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	User logged into admin dashboard using valid user name and password	User should be successfully log into the admin dashboard		
2	Click the Configure Analytics menu	Configure analytics menu should be successfully appeared		
3	Select the Enable check-box to enable statistical data publishing	Text fields should be appeared after enable		
4	User gives data Analyzer Configurations	configurations should be added successfully		
5	User fills statistics summary datasource (url,jdbc class, username,password) User gives more options if needed (max active, max wait)	configuration should be added successfully		
6	User clicks save button	message should be prompted message text - "Please add at least one URL Group"		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-410: User should be able to view already configuration after logged into admin dashboard again [Version : 1]

Author:	Shamin Goonetileke

Summary:

User who has successfully logged into admin dashboard again (https://<Server Host>:9443/admin-dashboard) should be able to view already added configuration

Preconditions:

1) API manager should be integrated with BAM by following the steps in [1]

[1] https://docs.wso2.com/display/AM190/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click the Configure Analytics menu	Configure analytics menu should be successfully appeared with already added URL group, Data analyzer configurations and Data analyzer configurations on text fields		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

8.2.Test Suite : API Publisher Statitics

Test Artifact	
Mind Map	

Test Case APIM-266: Check whether user can successfully view statistics of API Usage by Current Subscribers. [Version : 1]

Author:	Sewmini Jayaweera

Summary:

User who is successfully logged into API publisher should be able to view statistics of API Usage by Current Subscribers.

Preconditions:

1. There should be a user (publisher1), having publisher permission, registered in API manager.

2. There should be two versions of the same API (API1) published.

3. Subscriptions and invocations relavent to API versions should be as below.

API version	Subscriber	Number of API calls.
1.0.0	subscriber1	10
1.0.0	subscriber2	20
1.0.0	subscriber3	20
2.0.0	Subscriber1	30

4. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as publisher1 go to browse > API1 v1.0.0 > users page.	• 'Usage by Current Subscribers (v-1.0.0)' pie chart should show percentage of api calls made by each subscriber as subscriber1 - 20% , subscriber2 - 40% , subscriber 3 -40%. • Also there should be a list having number of api calls made by each subscriber as subscriber1 - 10 , subscriber2- 20, subscriber3 - 20. • 'Usage by Current Subscribers (Across All Versions)' Pie chart should show API calls made across all versions as subscriber1 - 50 %, subscriber2 - 25%, subscriber3 - 25%. • Table related to the pie chart should show numbsr of api calls made by subscriber1 - 40 , subscriber2 - 20, subscriber3 - 20.		
2	Go to browse > API1 v2.0.0 > users page.	• Usage by Current Subscribers (v-2.0.0) pie chart should show percentage as 100% and table should list subscriber1 with 30 api calls. • 'Usage by Current Subscribers (Across All Versions)' Pie chart should show API calls made across all versions as subscriber1 - 50 %, subscriber2 - 25%, subscriber3 - 25%. • Table related to the pie chart should show numbsr of api calls made by subscriber1 - 40 , subscriber2 - 20, subscriber3 - 20.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-265: Check whether user can successfully view statistics of API subscriptions by versions. [Version : 1]

Author: Sewmini Jayaweera

Summary:

User who is successfully logged in to API publisher should be able to view statistics of API subscriptions by versions successfully.

Preconditions:

1. User with publisher permission should successfully registered in API manager.

2. There should two versions published from the same API. (API version 1.0.0 and API version 2.0.0)

3. Subscriber1, subscriber2, subscriber3 should get subscribe to API version 1.0.0

4. Subscriber4 should get subscribe to API version 2.0.0.

5. API manager should be integrated with BAM by following [1].

[1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher as Publisher1 and go to API - 1.0.0 > versions.	API Subscriptions by Versions pie chart should depict 75% as subscriptions for API version 1.0.0 and 25 % for API version 2.0.0. Number of subscriptions for version 1.0.0 should list as 3 and number of subscriptions API version 2.0.0 has should list as 1.		
2	Unsubscribe subscriber 3 from API version 1.0.0 and go to API > version page.	API Subscriptions by Versions pie chart should depict 66.7% as subscriptions for API version 1.0.0 and 33.3 % for API version 2.0.0. Number of subscriptions for version 1.0.0 should be 2 Number of subscriptions for version 2.0.0 should be 1.		
3	Subscribe to API version 2.0.0 using subscriber 3 and got to version tab of api version 2.0.0.	API Subscriptions by Versions pie chart should depict 50% as subscriptions for API version 1.0.0 and 50 % for API version 2.0.0. Number of subscriptions for version 1.0.0 should be 2 Number of subscriptions for version 2.0.0 should be 2.		
4	Check statistics of API subscriptions by versions using both API versions. (API - 1.0.0 and API - 2.0.0)	Statisctics should be the same irrespective of API version.		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-264: Check whether publisher can view API usage of the particular API by versions. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: User who is successfully logged into api publisher should be able to view API usage of the particular API by versions.				
Preconditions: 1. User with publisher permission should be successfully registered in API manager. 2. There should be 2 versions of the same API (API 1.0.0 , API 2.0.0). 3. API version 1.0.0 should be invoked 100 times by different applications. 4. API version 2.0.0 should be invoked 25 times by different applications. 5. API manager should be integrated with BAM by following [1]. [1]. https://docs.wso2.com/display/AM180/Publishing+API+Runtime+Statistics				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API publisher and click on one of the versions of API1 > versions .	API usage by verion pie chart should depict version 2.0.0 as 20% and version 1.0.0 as 80% and the table should list the number of invocations for API1 and API2 as 100 and 25 respectively.		
2	Check version statisctics for both API versions.	Results should be same for both.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

8.3.Test Suite : Southbound Stat Publishing

Test Artifact
Mind Map

9.Test Suite : API Gateway

Test Artifact	
Mind Map	

9.1.Test Suite : CORS

This feature will be used to test the possibility of accessing restricted resources (to be requested from a domain other than are in another domain outside the domain from which the resource originated) when API Store and API Gateway are running on different domains.

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_GATEWAY/CORS/Mindmaps [^]

9.1.1.Test Suite : CORS configuration enabled

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_GATEWAY/CORS/Mindmaps [^]

9.1.1.1.Test Suite : Access-Control-Allow Parameter verification

Preconditions

1. Create a user with publish and create permissions
2. Login to the API Publisher and create an API allowing all/selected HTTP methods (there might be test cases where only a few methods should be selected)
3. Login to the API Store and subscribe a user
4. Login to the API Store from the new user and subscribe the API to the 'DefaultApplication' or a new application

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_GATEWAY/CORS/Mindmaps [^]

9.1.1.1.1.Test Suite : Access-Control-Allow-Origin

Preconditions

1. Follow the precondition steps 1-4 mentioned in 'Access-Control-Allow Parameter verification' above.

2. Edit the Access-Control-Allow-Origin method of the api-manager.xml to either * or actual Store API URL

E.g.:- <Access-Control-Allow-Origin>*</Access-Control-Allow-Origin>
OR

<Access-Control-Allow-Origin>https://localhost:9443,http://localhost:9763</Access-Control-Allow-Origin>

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_GATEWAY/CORS/Mindmaps [^]

Test Case APIM-369: API invocation with Access-Control-Allow-Origin=specific valid Store address [Version : 1]

Author:	Evanthika Amarasiri			
Summary: When we have a clustered API Manager setup where the API Store and API Gateway are running on different nodes, we need to specify the correct HTTP & HTTPS URLs				
Preconditions: Edit the Access-Control-Allow-Origin method of the api-manager.xml to https://[store_hostname]:[store_httpsport],http://[store_hostname]:[store_httpport] If API Store is running without an offset, E.g.:- <Access-Control-Allow-Origin>https://localhost:9443,http://localhost:9763</Access-Control-Allow-Origin>				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke the GET/POST/PUT & DELETE methods of the API from Swagger	1. The expected response should be received 2. Wire logs should display two calls • A pre-flight call OPTIONS call to verify the allowed methods • The actual GET/POST/PUT/DELETE call		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-375: API invocation with Access-Control-Allow-Origin=an invalid Store address [Version : 1]				
Author:	Evanthika Amarasiri			
Summary: When we have a clustered API Manager setup where the API Store and API Gateway are running on different nodes, if we do not specify the correct Store URLs, the invocations done through Swagger/API-Console should fail.				
Preconditions: Edit the Access-Control-Allow-Origin method of the api-manager.xml to • https://[invalid_store_hostname]:[store_httpsport],http://[invalid_store_hostname]:[store_httpport] OR • https://[store_hostname]:[invalid_store_httpsport],http://[store_hostname]:[invalid_store_httpport] If API Store is running without an offset, E.g.:- <Access-Control-Allow-Origin>https://localhost:9453,http://localhost:9773</Access-Control-Allow-Origin>				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke the GET/POST/PUT & DELETE methods of the API from Swagger	All method calls should fail with the error "[[*To verify tomorrow and fill this*]]"		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-382: API invocation when connected to an external API Store [Version : 1]				
Author:	Evanthika Amarasiri			
Summary: This scenario will verify the CORS functionality when an external API store is configured				
Preconditions: 1. Configure an external API store as described in the document 'Publish to multiple external API stores'				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Change the methods of the API so that it allows only a few methods & invoke them using any client. E.g.:- JMeter	1. The response for the allowed method calls should be successfully retrieved 2. The restricted method calls should fail		
2	Invoke the GET/POST/PUT & DELETE methods of the API from any client E.g.:- JMeter	1. The expected response should be received 2. Wire logs should display two calls • A pre-flight call OPTIONS call to verify the allowed methods • The actual GET/POST/PUT/DELETE call		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

9.1.1.1.2.Test Suite : Access-Control-Allow-Methods

Preconditions

1. Follow the precondition steps 1-4 mentioned in 'Access-Control-Allow Parameter verification' above.

2. Edit the Access-Control-Allow-Methods method of the api-manager.xml to have either all the methods as listed below or a few methods.

E.g.:- <Access-Control-Allow-Methods>GET,POST,PUT,DELETE,OPTIONS</Access-Control-Allow-Methods>

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_GATEWAY/CORS/Mindmaps [^]

Test Case APIM-377: API invocation with Access-Control-Allow-Methods=All methods [Version : 1]	
Author:	Evanthika Amarasiri

Summary: This test case is written to verify which methods are allowed when accessing a resource. We will be verifying whether the methods allowed in the api-manager.xml are returned in the preflight OPTIONS call. -- *[need to verify the functionality]*				
Preconditions: Edit the Access-Control-Allow-Methods method of the api-manager.xml to <Access-Control-Allow-Methods>*[HTTP methods]*</Access-Control-Allow-Methods> where [*HTTP headers*] = GET,POST,PUT,DELETE,OPTIONS				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	• Create an API with the methods GET,POST,PUT,DELETE selected • Invoke the API from Swagger	1. The Access-Control-Allow-Methods defined in the api-manager.xml should be returned in the response of the preflight OPTIONS call. 2. All the GET,POST,PUT,DELETE requests should be successful		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-378: API invocation with Access-Control-Allow-Methods=Selected # of methods [Version : 1]				
Author:	Evanthika Amarasiri			
Summary: This test case is written to verify which methods are allowed when accessing a resource. We will be verifying whether the methods allowed in the api-manager.xml are returned in the preflight OPTIONS call. -- *[need to verify the functionality]*				
Preconditions: Edit the Access-Control-Allow-Methods method of the api-manager.xml to <Access-Control-Allow-Methods>*[Selected HTTP methods]*</Access-Control-Allow-Methods> where [*HTTP headers*] = GET,POST,PUT,DELETE,OPTIONS				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	• Create an API with the methods GET,POST,DELETE selected (This can be tried with different combinations of methods) • Invoke the API from Swagger	1. The Access-Control-Allow-Methods defined in the api-manager.xml (GET, POST & DELETE only) should be returned in the response of the preflight OPTIONS call. 2. All the GET,POST & DELETE requests should be successful 3. The PUT request should fail saying that the method is forbidden to be accessed.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

9.1.1.1.3.Test Suite : Access-Control-Allow-Headers

Preconditions

1. Follow the precondition steps 1-4 mentioned in 'Access-Control-Allow Parameter verification' above.

2. Edit the Access-Control-Allow-Headers method of the api-manager.xml to either one of the following

- authorization
- Access-Control-Allow-Origin
- Content-Type
- Origin
- Accept
- Any other headers

E.g.:- <Access-Control-Allow-Headers>authorization,Access-Control-Allow-Origin,Content-Type</Access-Control-Allow-Headers>

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_GATEWAY/CORS/Mindmaps [^]

Test Case APIM-376: API invocation with Access-Control-Allow-Headers=different types of headers [Version : 1]	
Author:	Evanthika Amarasiri

Summary: This test cases is used to verify whether when a preflight OPTIONS call is done, whether it would return the HTTP headers that we request for through the configuration 'Access-Control-Allow-Headers' in the api-manager.xml.	
Preconditions: Edit the Access-Control-Allow-Origin method of the api-manager.xml to <Access-Control-Allow-Headers>*[HTTP headers]*</Access-Control-Allow-Headers> where [*HTTP headers*] = • authorization • Access-Control-Allow-Origin • Content-Type • Accept • Origin • SOAPAction • tenant • wildcards i.e.:- * • any other header	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke the GET/POST/PUT & DELETE methods of the API from Swagger when Access-Control-Allow-Header=authorization	1. The authorization header should be returned in the response of the preflight OPTIONS call. 2. Verify whether the correct authorization headers values of the backend are returned		
2	Invoke the GET/POST/PUT & DELETE methods of the API from Swagger when Access-Control-Allow-Header=Access-Control-Allow-Origin	1. The Access-Control-Allow-Origin header should be returned in the response of the preflight OPTIONS call. 2. The origin of the message should be sent as the Access-Control-Allow-Origin header value		
3	Invoke the GET/POST/PUT & DELETE methods of the API from Swagger when Access-Control-Allow-Header=Content-type with different services which returns messages of different content types • application/xml • application/json • other content types	1. The Content-Type header should be returned in the response of the preflight OPTIONS call. 2. The correct Content-Type of the message (sent from the backend) should be returned.		
4	Remove the headers given by default in the api-manager.xml & invoke the API	1. The removed header details should not be returned with the preflight OPTIONS call 2. API should be invoked successfully		
5	Invoke the GET/POST/PUT & DELETE methods of the API from Swagger when Access-Control-Allow-Header=*	1. All the headers that are allowed by the backend server should be returned. *[[This feature is yet to be implemented]]*		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

9.1.1.2.Test Suite : Verify CORS for different API Types

This feature will verify how CORS behaves for different types of APIs

E.g.:-

- New APIs
- Copied APIs
- Prototyped APIs

Preconditions

1. Create a user with publish and create permissions

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_GATEWAY/CORS/Mindmaps [^]

Test Case APIM-380: Verify CORS behaviour for versioned APIs [Version : 1]	
Author:	Evanthika Amarasiri
Summary: This feature will verify how CORS functions behave when APIs are versioned.	
Preconditions: 1. Login to the API Publisher and create an API allowing all/selected HTTP methods (there might be test cases where only a few methods should be selected) 2. Publish the API 3. Create a new copy of the current version 3. Login to the API Store and subscribe a user 4. Login to the API Store from the new user and subscribe the ***versioned/copied*** API to the 'DefaultApplication' or a new application	

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	From the API Store, access the API Console and invoke the API from all available methods. • GET • POST • PUT • DELETE	1. The initial preflight OPTIONS call should be sent to the backend to verify the allowed methods. 2. The response to the relevant resource method call should be received successfully		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-381: Verify CORS behaviour for prototyped APIs [Version : 1]

Author:	Evanthika Amarasiri

Summary:

This feature will verify how CORS functions behave when APIs are versioned.

Preconditions:
1. Login to the API Publisher and create an API allowing all/selected HTTP methods (there might be test cases where only a few methods should be selected)
2. Publish the API
3. Create a new copy of the current version
3. Login to the API Store and subscribe a user
4. Login to the API Store from the new user and subscribe the ***versioned/copied*** API to the 'DefaultApplication' or a new application

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	From the API Store, access the API Console and invoke the API from all available methods. • GET • POST • PUT • DELETE	1. The initial preflight OPTIONS call should be sent to the backend to verify the allowed methods. 2. The response to the relevant resource method call should be received successfully		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

9.1.2.Test Suite : CORS configuration disabled

This test case will verify whether the Swagger Console stops functioning when the CORS configuration is disabled

Test Artifact	
Mind Map	https://github.com/wso2/wso2-qa-artifacts/tree/master/products/wso2_APIManager/API_GATEWAY/CORS/Mindmaps [^]

Test Case APIM-379: Invoking APIs when CORS configuration is disabled [Version : 1]

Author:	Evanthika Amarasiri

Summary:

This test cases will verify whether the functionality of Swagger/API console is disabled when CORS configuration is disabled

Preconditions:
Set the <CORSConfiguration>/<Enabled> to false in api-manager.xml

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Invoke the GET/POST/PUT & DELETE methods of the API from Swagger	All method calls should fail		
3	Invoke the API from some other client E.g.:- JMeter	The method invocations should be successful		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	

Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

10.Test Suite : Application sharing capabilities

Enable below property in <Product_home>/repository/conf/api-manager.xml

<GroupingExtractor>org.wso2.carbon.apimgt.impl.DefaultGroupIDExtractorImpl</GroupingExtractor>

Test Artifact
Mind Map

Test Case APIM-425: Test whether applications do not share between different tenant domain. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: User of different tenat domain with same organization and subscriber permission should not be able to share applications.				
Preconditions: 1. There should be subscrber1 in super tenant domain and subscriber 2 in another tenant domain (wso2), both having subscriber permission. 2. There should be an application created by subscriber1 (app1)				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API store using subscriber2 and check whether app1 created by subscriber1 is visible.	Applications created by subscribers of other tenant domains should not be visible.		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			

Test Case APIM-423: Test whether application sharing capabilities work accurately based on user's organisation. [Version : 1]				
Author:	Sewmini Jayaweera			
Summary: Users who are in the same organisation with subscriber permission should be able to view applications created by each subscriber user whereas other users who are not in the same organisation cannot.				
Preconditions: 1. There should be three users user1, user2 and user3 who is given subscriber permission. 2. User1 and user 2 should given same organization (say wso2) and user3 should be given another organization (say lfs) 3. User1 should have created an application called App1. 4. User3 should have created an application called APP3				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API store using user2 check whether app1 is visible for him.	App1 should be visible under, • store > my applications page • Store > APIs > API1 > applications drop down list.		
2	Re-login to API store with user3	APP1 should not be visible for user3		
3	Update organization name of user 2 given a space at the end of wso2 ("wso2 ") and relogin to API store.	App1 should be visible under, • store > my applications page • Store > APIs > API1 > applications drop down list.		
4	Change organization of user2 to organization of user3 (ifs) update changes and relogin to API store.	• API 1 should no longer visible for user2. • API 3 should be visible under store > my applicatios page and store > APIs > API1 >Applications drop down list.		
5	Update organization name of user1 and user2 to 'perera & sons' and save.	Changes should successfully get updated.		
6	Login to API store using user2	App1 should be visible under, • store > my applications page • Store > APIs > API1 > applications drop down list.		
7	By previous user create an application given the name APP2.	Application should successfully created.		
8	Login to API store using user1 and check whether app2 is visible.	App2 should be visible under, • store > my applications page • Store > APIs > API1 > applications drop down list.		
Execution type:	Manual			

Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

Test Case APIM-424: Test whether users can subscribe to APIs using shared applications, generate keys and invoke API [Version : 1]

Author:	Sewmini Jayaweera

Summary:

Users with subscriber permission should be able to subscribe to APIs using shared applications, generate keys and invoke APIs.

Preconditions:

1. There should be three users user1, user2 and user3 who is given subscriber permission.

2. All users should be given the same organization name (99X)

3. User1 should have created an application called App1.

4. There should be an already published APIs (api1 and API2).

Note : This test case should be tested with secondary user store such as open ldap, active directory.

#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Login to API store using user2 and check whether App1 is visible.	App1 should be visible under, • store > my applications page • Store > APIs > API1 > applications drop down list.		
2	By user2 subscribe to API1 using App1.	user should be able to successfully subscribed to api using app1. api1 should listed under subscribed APIs section of my subscription page.		
3	By user2 generate keys for app1 by selecting app1 from 'applications with subscriptions' drop down menu and clicking on 'generate' button.	consumer key consumer secret and application token should be successfully generated.		
4	Invoke API1 using application token generated in previous step.	API1 should be successfully invoked.		
5	Login to API store using user3 and check whether app1 is visible.	App1 should be visible under, • store > my applications page • Store > APIs > API1 > applications drop down list.		
6	by the user3 subscribe to API2 using app1	• api should listed under subscribed APIs section of my subscription page. • and the generated application keys should be visible.		
7	Invoke API2 using application token of app1	User should be able to successfully invoke api2		

Execution type:	Manual
Estimated exec. duration (min):	
Importance:	Medium
Artifacts:	
Artifacts:	
Artifacts:	
Automation Test Case:	
Requirements	None
Keywords:	None
Last Result	**Not Run**

11.Test Suite : External API Store

This suite includes test cases written for external API store

Test Artifact
Mind Map

Test Case APIM-426: Publish an API created in to an external API store [Version : 1]				
Author:	Ushani Balasooriya			
Summary: Sucessfully logged in punlisher user should be able to publish an API created in to external api store configured. **Pls note :** **This feature shold be tested in Super tenant, tenant modes and API's visibility in to role based.** ***Refer the mail thread :*** ***Clarification on visibility of APIs in an external store for restricted by roles options***				
Preconditions: 1. A user with admin priviledges should have logged in to admin console and configured an external store in registry as per the documentation. 2. A publisher user should have logged in to the publisher and created an API				
#:	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Click on publish to external api store button if the store is sucessfully configured	API should be published in the External API store		
2	Click on the go to publisher store	User should be directed to original publisher's store		
Execution type:	Manual			
Estimated exec. duration (min):				
Importance:	Medium			
Artifacts:				
Artifacts:				
Artifacts:				
Automation Test Case:				
Requirements	None			
Keywords:	None			
Last Result	**Not Run**			