

Day 4

SAP Advanced Event Mesh – Use APIs Mission Control

CONTENTS

EXERCISE: USE APIS	3
PREREQUISITES.....	3
SOLUTION 1: USE APIS	4

Day 4

Exercise: Use APIs

- Understand the different APIs of Advanced Event Mesh
- Work with an Postman collection: Create and Delete a new Development Broker Service
- Publish a "Rest"-Event via Postman
- Create a new Queue via SEMP API
- Check Queues with SEMP API

Prerequisites

- You access and use the same broker you setup previously

Day 4 Solution 1

Solution 1: Use APIs

Exercise 1

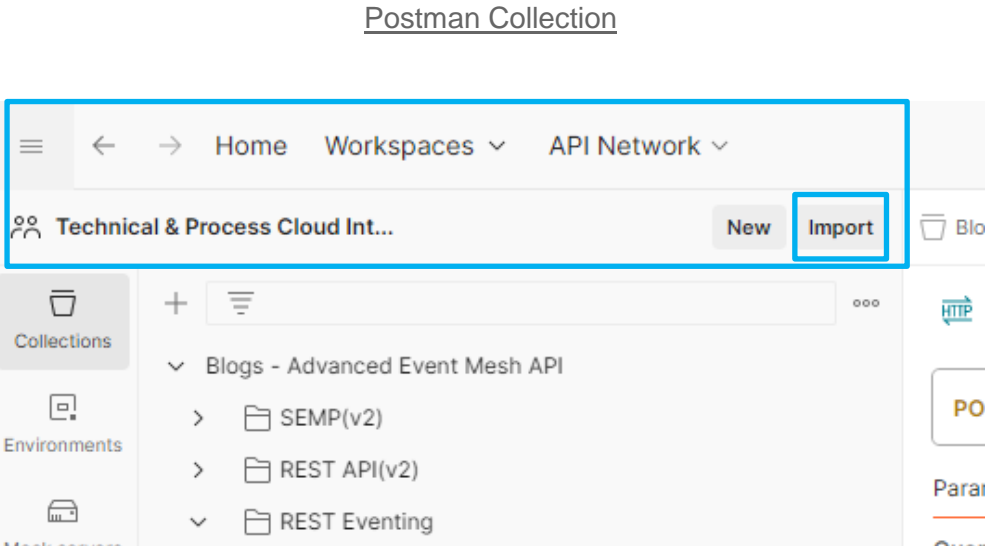
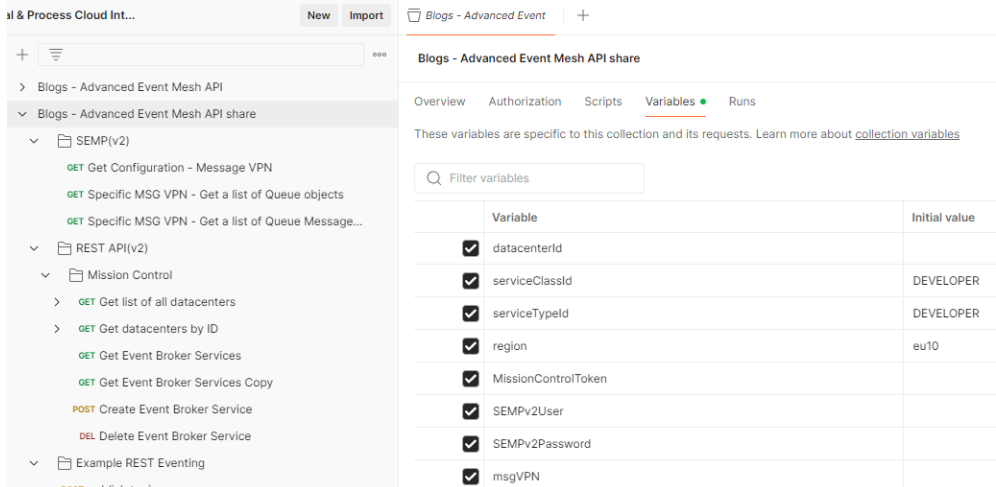
- Create Token for Mission Control
- Download PostMan – Collection
- Identify REST API(v2) Cloud URL
- Use the Mission Control Folder in PostMan
 - Get list of all Datacenters
 - Get list of all Event Broker Service in your Landscape
 - Delete 1 of the Event Broker via specific ID (delete your demo Event Broker)
 - Create 1 new Event Broker with Development Attributes

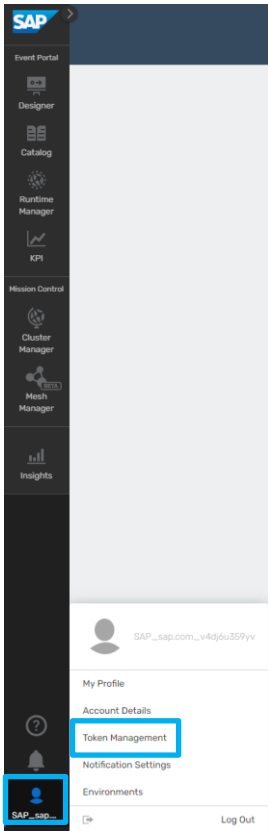
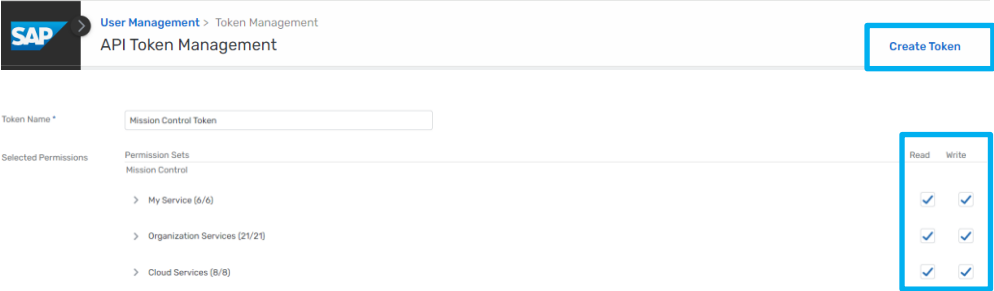
Exercise 2

- Create a new Queue in your Event Broker Service
- Create a new Topic subscription in your Event Broker Service
- Publish a new Message in a specific Queue / Topic subscription

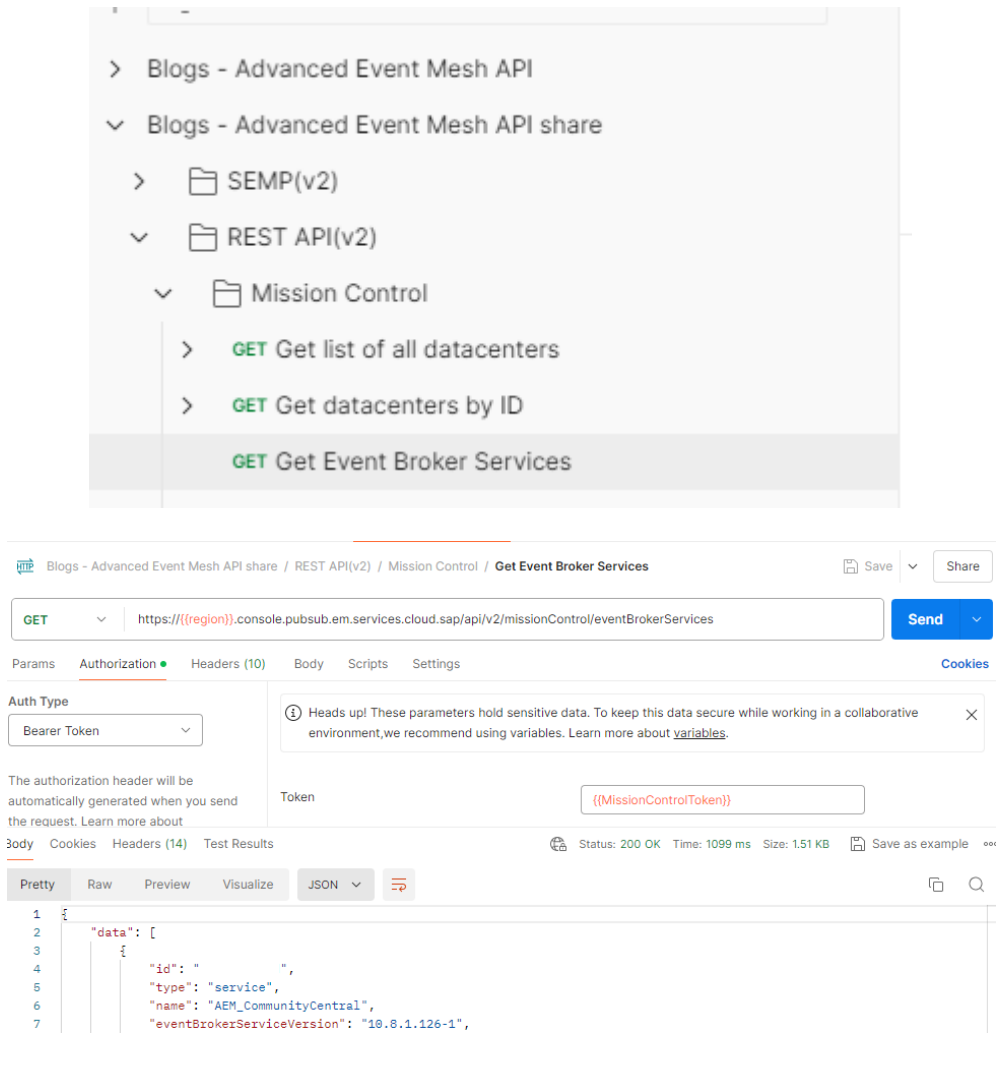
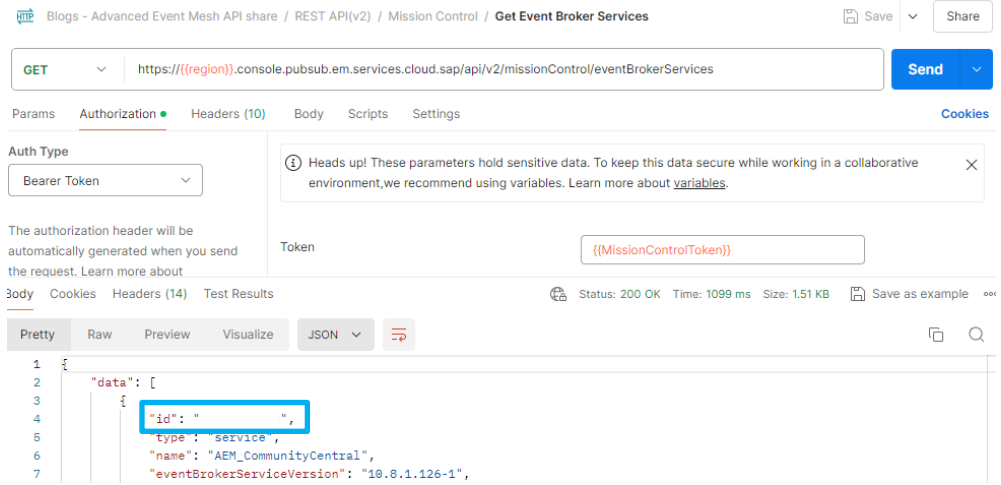
Exercise 3

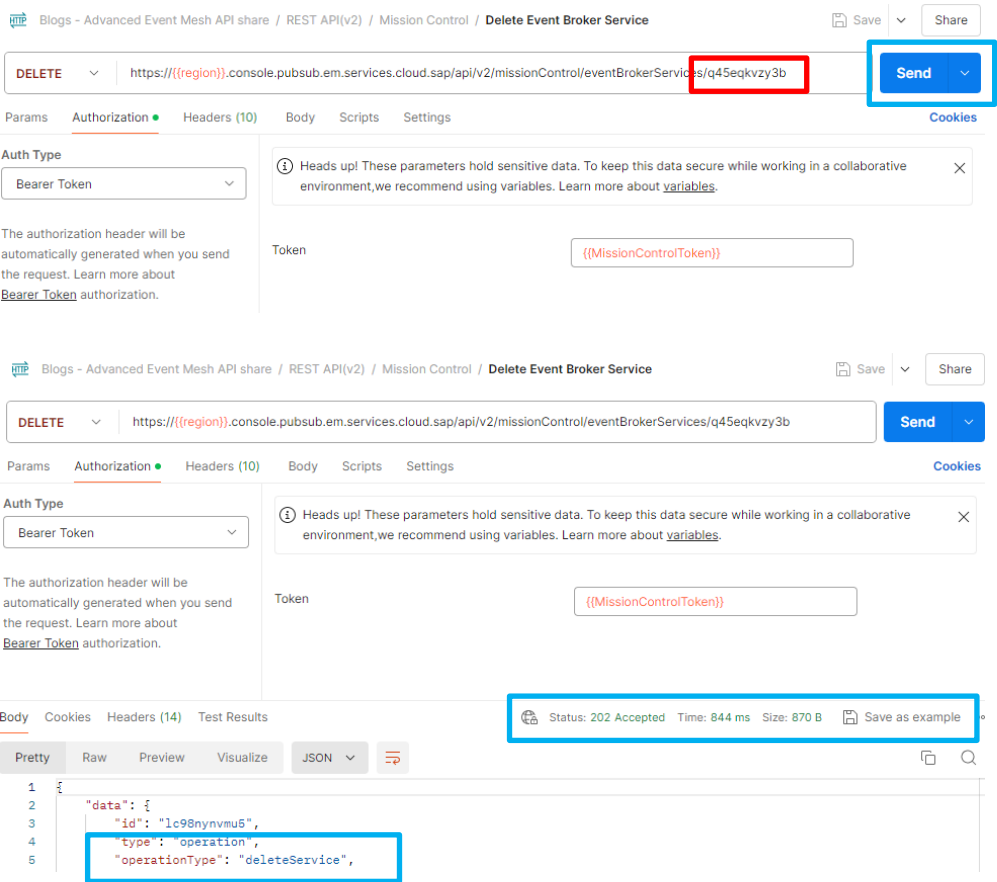
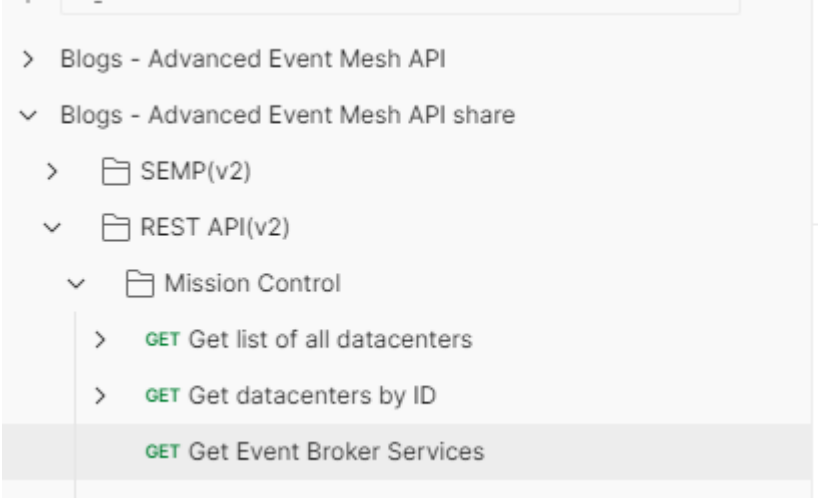
- Leverage the SEMP – API
 - Create a new Queue by Payload and Publish an other Rest Message
 - Identify your published message in a specific Queue

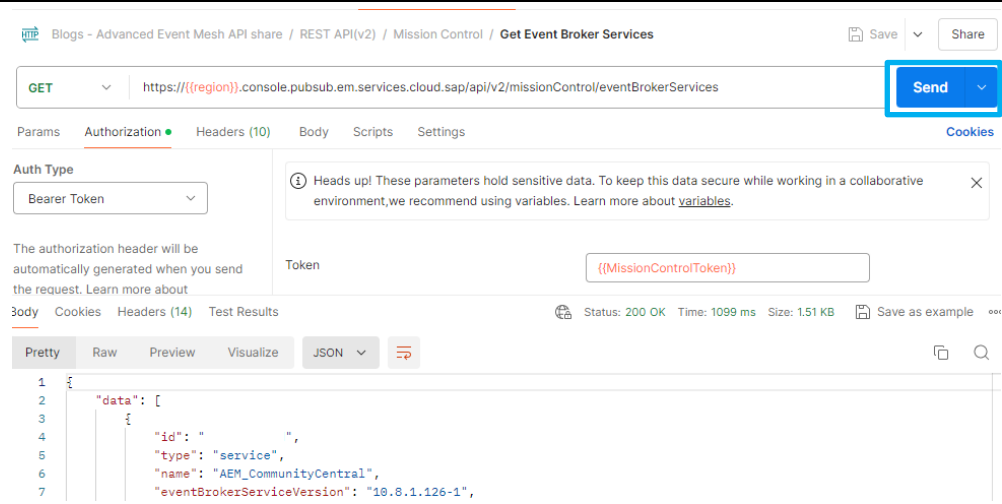
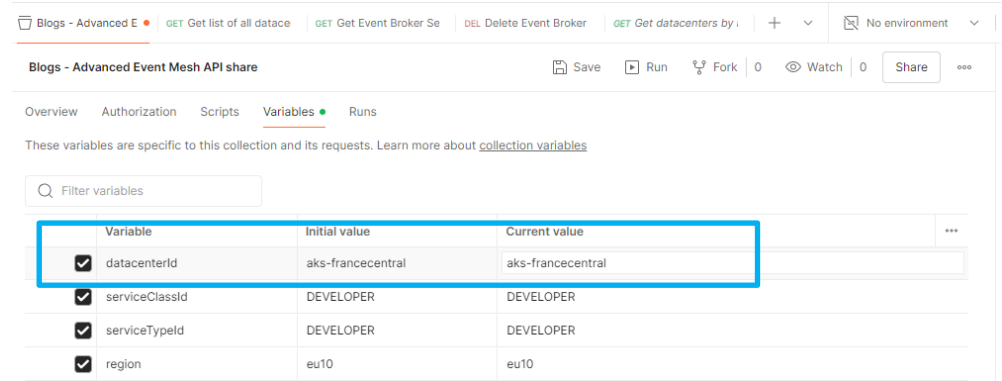
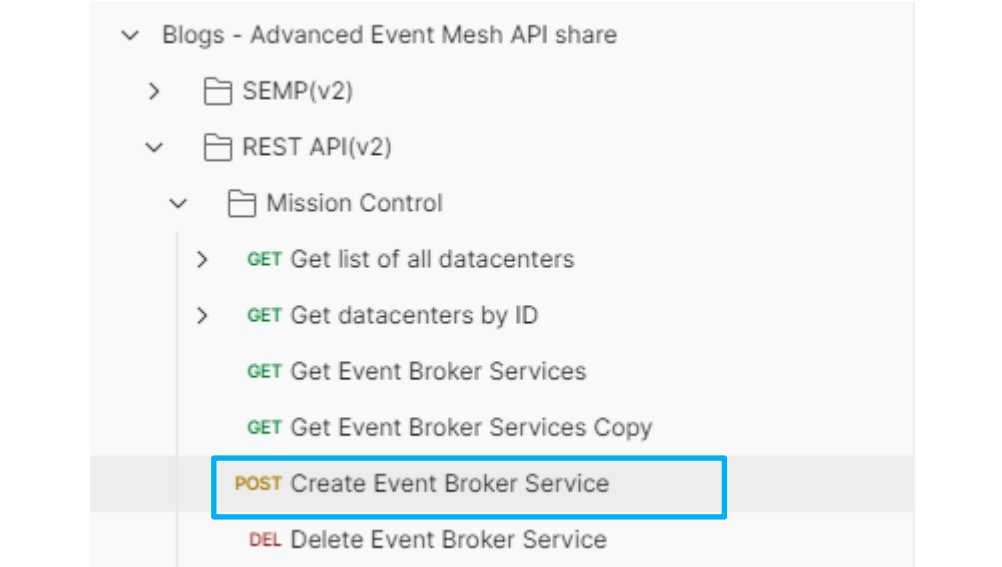
Explanation	Screenshot																		
<p>Download the Postman collection with some basic apis and import this collection in your postman</p>	 <p>Postman Collection</p> <p>Home Workspaces API Network</p> <p>Technical & Process Cloud Int... New Import</p> <p>Collections</p> <p>Environments</p> <p>Blogs - Advanced Event Mesh API</p> <p>SEMP(v2)</p> <p>REST API(v2)</p> <p>REST Eventing</p>																		
<p>Identify your region variable and update it in the variables section of Postman</p>	 <p>Technical & Process Cloud Int... New Import</p> <p>Blogs - Advanced Event Mesh API share</p> <p>Overview Authorization Scripts Variables Runs</p> <p>These variables are specific to this collection and its requests. Learn more about collection variables</p> <p>Filter variables</p> <table border="1"> <thead> <tr> <th>Variable</th><th>Initial value</th></tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> datacenterId</td><td></td></tr> <tr> <td><input checked="" type="checkbox"/> serviceClassId</td><td>DEVELOPER</td></tr> <tr> <td><input checked="" type="checkbox"/> serviceTypeId</td><td>DEVELOPER</td></tr> <tr> <td><input checked="" type="checkbox"/> region</td><td>eu10</td></tr> <tr> <td><input checked="" type="checkbox"/> MissionControlToken</td><td></td></tr> <tr> <td><input checked="" type="checkbox"/> SEMPv2User</td><td></td></tr> <tr> <td><input checked="" type="checkbox"/> SEMPv2Password</td><td></td></tr> <tr> <td><input checked="" type="checkbox"/> msgVPN</td><td></td></tr> </tbody> </table> <p>Blogs - Advanced Event Mesh API</p> <p>SEMP(v2)</p> <p>GET Get Configuration - Message VPN</p> <p>GET Specific MSG VPN - Get a list of Queue objects</p> <p>GET Specific MSG VPN - Get a list of Queue Message...</p> <p>REST API(v2)</p> <p>Mission Control</p> <p>GET Get list of all datacenters</p> <p>GET Get datacenters by ID</p> <p>GET Get Event Broker Services</p> <p>GET Get Event Broker Services Copy</p> <p>POST Create Event Broker Service</p> <p>DEL Delete Event Broker Service</p> <p>Example REST Eventing</p> <p>POST publish topic</p>	Variable	Initial value	<input checked="" type="checkbox"/> datacenterId		<input checked="" type="checkbox"/> serviceClassId	DEVELOPER	<input checked="" type="checkbox"/> serviceTypeId	DEVELOPER	<input checked="" type="checkbox"/> region	eu10	<input checked="" type="checkbox"/> MissionControlToken		<input checked="" type="checkbox"/> SEMPv2User		<input checked="" type="checkbox"/> SEMPv2Password		<input checked="" type="checkbox"/> msgVPN	
Variable	Initial value																		
<input checked="" type="checkbox"/> datacenterId																			
<input checked="" type="checkbox"/> serviceClassId	DEVELOPER																		
<input checked="" type="checkbox"/> serviceTypeId	DEVELOPER																		
<input checked="" type="checkbox"/> region	eu10																		
<input checked="" type="checkbox"/> MissionControlToken																			
<input checked="" type="checkbox"/> SEMPv2User																			
<input checked="" type="checkbox"/> SEMPv2Password																			
<input checked="" type="checkbox"/> msgVPN																			

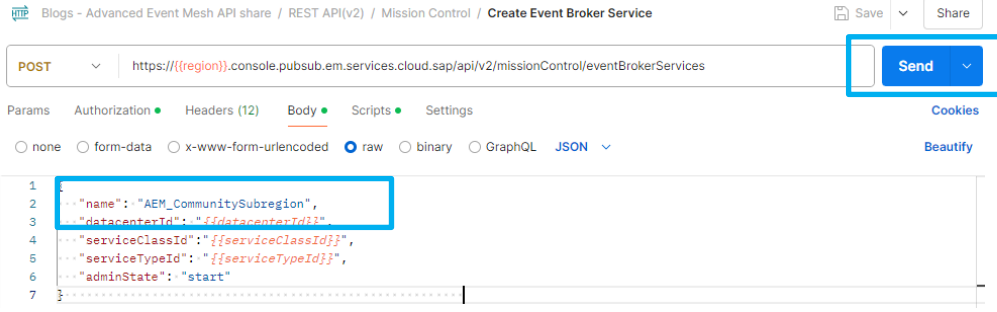
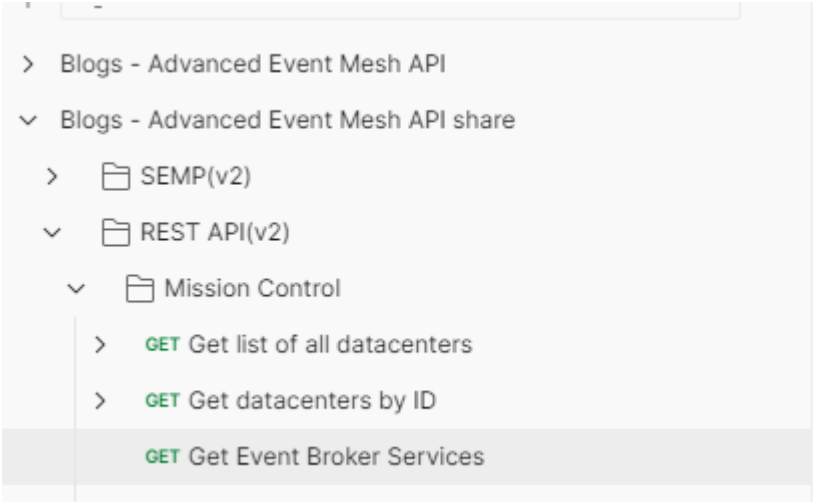
Explanation	Screenshot
<p>Open Token Management under the User & Account overview</p> <p>Hint: Create up to 50 API tokens (per user) in an account</p> <p>Recommendation: Manage API Token via a central technical user/users (if more then 50)</p>	 <p>The screenshot shows the SAP User Management interface. On the left is a dark sidebar with various navigation icons. The main area is light gray. At the bottom of the sidebar, the user profile 'SAP_sap...' is visible. In the main area, the 'Token Management' option is highlighted with a blue box. Other options like 'My Profile', 'Account Details', 'Notification Settings', and 'Environments' are also visible.</p>
<p>Create a new Token for “Mission Control Token” with specific rights for Mission Control</p> <p>Very granular permissions can be assigned according to individual needs and company policies.</p>	 <p>The screenshot shows the 'Create Token' form in SAP. The breadcrumb is 'User Management > Token Management'. The title is 'API Token Management'. There is a 'Create Token' button. The 'Token Name' field contains 'Mission Control Token'. Under 'Selected Permissions', there are three permission sets: 'My Service (6/6)', 'Organization Services (21/21)', and 'Cloud Services (8/8)'. On the right, there is a table with 'Read' and 'Write' columns, both of which have checkmarks in the rows corresponding to the selected permissions.</p>

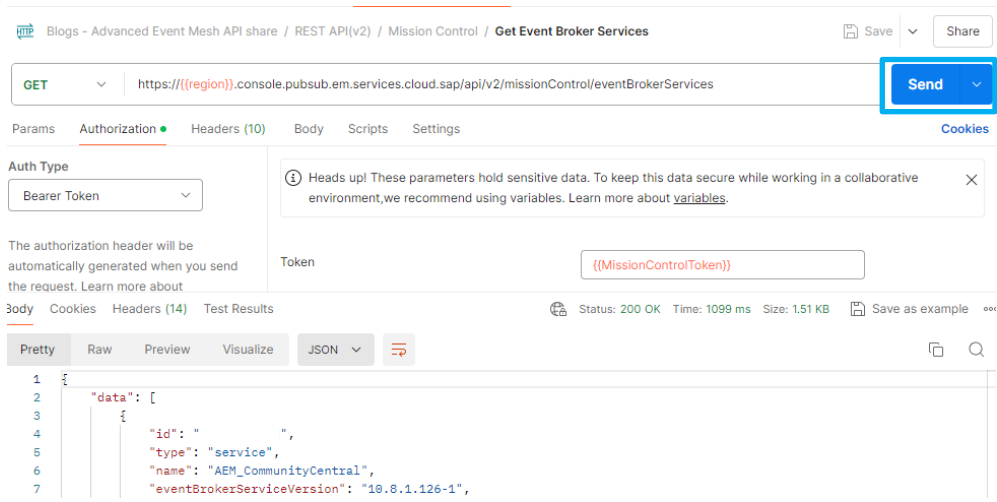
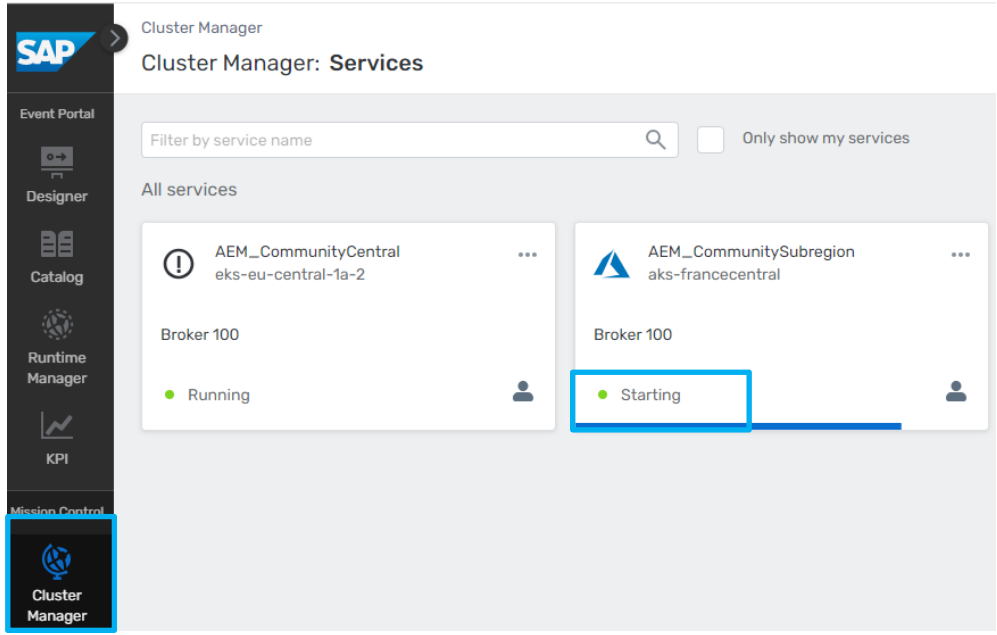
Explanation	Screenshot
Copy Mission Control Token	<p>API Token</p> <p>To use the API token, you must include it in the Authorization Bearer header when making a REST call.</p> <p>Copy and retain your token. Once closed, you cannot reaccess it for security reasons</p> <p>> API Token Details</p> <p>Copy</p> <p>Close</p>
Update Mission ControlToken variable in Postman Collection.	<p>Postman interface showing the 'Variables' tab for the 'Blogs - Advanced Event Mesh API share' collection. The 'MissionControlToken' variable is highlighted with a blue box.</p>
<p>Check the function in Postman. Go to: REST API(v2) → Mission Control → Get list of all datacenters.</p> <p>Perform a “Send” request</p> <p>Result: A list of all Datecenters will be there.</p>	<p>Postman interface showing the 'GET' request to 'https://(region).console.pubsub.em.services.cloud.sap/api/v2/missionControl/datacenters'. The 'Send' button is highlighted with a blue box.</p> <p>The 'Test Results' tab shows the JSON response:</p> <pre> 1 { 2 "data": [3 { 4 "id": "eks-ap-southeast-3a", 5 "type": "datacenter", 6 }, 7], 8 } </pre>

Explanation	Screenshot
<p>Go to: REST API(v2) → Mission Control → Get Event Broker Services.</p> <p>Perform a “Send” request</p> <p>Result: <i>A list of all Broker Services will be there.</i></p>	 <p>The screenshot shows the SAP API Explorer interface. The breadcrumb path is: Blogs - Advanced Event Mesh API share / REST API(v2) / Mission Control / Get Event Broker Services. The HTTP method is GET, and the URL is https://(region).console.pubsub.em.services.cloud.sap/api/v2/missionControl/eventBrokerServices. The authorization type is Bearer Token, and the token is ((MissionControlToken)). The response is a JSON array with one object: {"id": "", "type": "service", "name": "AEM_CommunityCentral", "eventBrokerServiceVersion": "10.0.1.126-1"}. The 'id' field is highlighted in blue.</p>
<p>Pick the id of one Broker and get ready for deletion.</p>	 <p>The screenshot shows the SAP API Explorer interface. The breadcrumb path is: Blogs - Advanced Event Mesh API share / REST API(v2) / Mission Control / Get Event Broker Services. The HTTP method is GET, and the URL is https://(region).console.pubsub.em.services.cloud.sap/api/v2/missionControl/eventBrokerServices. The authorization type is Bearer Token, and the token is ((MissionControlToken)). The response is a JSON array with one object: {"id": "", "type": "service", "name": "AEM_CommunityCentral", "eventBrokerServiceVersion": "10.0.1.126-1"}. The 'id' field is highlighted in blue.</p>

Explanation	Screenshot
<p>Information</p> <p>With this call, you will delete a Broker Service. Please ensure it is only a Demo/Test Broker Service. There is no way to get this service back.</p> <p>Go to: REST API(v2) → Mission Control → Delete Event Broker Service.</p> <p>Change ID in the URL path (marked in red)</p> <p>Perform a “Send” request</p> <p>Result: <i>The specific selected Broker is deleted.</i></p>	 <p>The screenshot shows the REST client interface for the 'Delete Event Broker Service' endpoint. The URL is <code>https://((region)).console.pubsub.em.services.cloud.sap/api/v2/missionControl/eventBrokerServices/q45eqkvzy3b</code>, where the ID <code>q45eqkvzy3b</code> is highlighted in red. The 'Send' button is highlighted in blue. The response status is 202 Accepted, and the response body is a JSON object: <code>{ "data": { "id": "lc98nynvmu5", "type": "operation", "operationType": "deleteService" } }</code>, where the entire response body is highlighted in blue.</p>
<p>Cross check: Go to: REST API(v2) → Mission Control → Get Event Broker Services.</p> <p>Perform a “Send” request</p> <p>Result: <i>A list of all Broker services will be there. The Broker service which you picked for deletion is not in the list anymore.</i></p>	 <p>The screenshot shows the REST client interface for the 'Get Event Broker Services' endpoint. The endpoint is highlighted in blue. The response status is 202 Accepted, and the response body is a JSON object: <code>{ "data": { "id": "lc98nynvmu5", "type": "operation", "operationType": "deleteService" } }</code>, where the entire response body is highlighted in blue.</p>

Explanation	Screenshot
	
<p>Fill the datacenterid variable according to the result list of Get list of all datacenters.</p> <p>“Save”</p> <p>In our example we create a new Broker Service in “aks-francecentral”</p>	
<p>Go to: REST API(v2) → Mission Control → Create Event Broker Services.</p>	

Explanation	Screenshot
<p>Change name of the new Broker Service</p> <p>In our example we create a new Broker Service in "AEM_CommunitySubregion"</p> <p>Perform a "Send" request</p> <p>Result: A list of all Broker Services will be there.</p>	 <p>POST <code>https://(region).console.pubsub.em.services.cloud.sap/api/v2/missionControl/eventBrokerServices</code></p> <p>Params Authorization Headers (12) Body Scripts Settings Cookies Beautify</p> <p>none form-data x-www-form-urlencoded raw binary GraphQL JSON</p> <pre> 1 { 2 "name": "AEM_CommunitySubregion", 3 "datacenterId": "{{datacenterId}}", 4 "serviceClassId": "{{serviceClassId}}", 5 "serviceTypeId": "{{serviceTypeId}}", 6 "adminState": "start" 7 } </pre> <p>POST <code>https://(region).console.pubsub.em.services.cloud.sap/api/v2/missionControl/eventBrokerServices</code></p> <p>Params Authorization Headers (12) Body Scripts Settings Cookies Beautify</p> <p>none form-data x-www-form-urlencoded raw binary GraphQL JSON</p> <pre> 1 { 2 "name": "AEM_CommunitySubregion", 3 "datacenterId": "{{datacenterId}}", 4 "serviceClassId": "{{serviceClassId}}", 5 "serviceTypeId": "{{serviceTypeId}}", 6 "adminState": "start" 7 } </pre> <p>Body Cookies Headers (14) Test Results (0/1) Status: 202 Accepted Time: 1524 ms Size: 870 B Save as example</p> <p>Pretty Raw Preview Visualize JSON</p> <pre> 1 { 2 "data": { 3 "id": "0j0x00000000000000000000000000000000", 4 "type": "operation", 5 "operationType": "createService" 6 } 7 } </pre>
<p>Cross check: Go to: REST API(v2) → Mission Control → Get Event Broker Services.</p> <p>Perform a "Send" request</p> <p>Result: A list of all Broker services will be there. The new Broker service which you created before is now also in the list</p>	 <p>Blogs - Advanced Event Mesh API</p> <p>Blogs - Advanced Event Mesh API share</p> <p>SEMP(v2)</p> <p>REST API(v2)</p> <p>Mission Control</p> <p>GET Get list of all datacenters</p> <p>GET Get datacenters by ID</p> <p>GET Get Event Broker Services</p>

Explanation	Screenshot
	 <p>The screenshot shows the SAP API Explorer interface. The URL is <code>https://(region).console.pubsub.em.services.cloud.sap/api/v2/missionControl/eventBrokerServices</code>. The request is a GET method. The response is a JSON object with the following structure:</p> <pre>1 { 2 "data": [3 { 4 "id": " ", 5 "type": "service", 6 "name": "AEM_CommunityCentral", 7 "eventBrokerServiceVersion": "10.8.1.126-1", 8 } 9] 10 }</pre>
Cross check: Go to Cluster Manager in your Adavanced Event Mesh Portal. The new created Broker Service is in <i>Starting</i> state.	 <p>The screenshot shows the SAP Cluster Manager interface. The left sidebar contains navigation links: Event Portal, Designer, Catalog, Runtime Manager, KPI, and Cluster Manager (highlighted). The main area displays 'Cluster Manager: Services'. A search bar is present. Below, 'All services' are listed. Two services are shown: 'AEM_CommunityCentral' (Running) and 'AEM_CommunitySubregion' (Starting). The 'Starting' state of the 'AEM_CommunitySubregion' service is highlighted with a red box.</p>
Congratulations! You successfully completed the Exercise	