**ADDING A POLICY ACTION TO A POLICY**

**ADDING POLICY ACTION OUTBOUND AUTHENTICATION TO ROUTING POLICY:**

**STEP 1:**

**url:** <<protocol>>://<<host>>:<<port>>/rest/apigateway/apis

**Method: GET**

**Accept**: application/json

**Content-type**: application/json

**Response Payload:**

{

"apiResponse": [

{

"api": {

"apiName": "API Gateway Application Management Service",

"apiVersion": "10.4",

"apiDescription": "API Gateway Application Management Service allows you to manage the Applications in API Gateway. Any user with the 'Manage applications' functional privilege can manage the APIs in the API Gateway. By default, the users who are part of either API-Gateway-Providers or API-Gateway-Administrators groups will have this privilege.\n\n An application defines the precise identifiers by which messages from a particular consumer application is recognized at run time. The identifiers can be, for example, user name in HTTP headers, a range of IP addresses, such that API Gateway can identify or authenticate the consumers that are requesting an API.\n\n. The ability of API Gateway to relate a request to a specific consumer application enables it to:\n\n 1. Control access to an API at run time (that is, allow only authorized consumer applications to invoke an API).\n\n 2. Monitor an API for violations of a Service-Level Agreement (SLA) for a specified application.\n\n 3. Indicate the consumer application to which a logged transaction event belongs.\n\n\n\n An application has the following attributes for specifying the identifiers:\n\n 1. IP address, which specifies one or more IP addresses that identify requests from a particular consumer application. Example: 192.168.0.10\n\n This attribute is queried when the Identify and Authorize Application policy is configured to identify consumer applications by IP address.\n\n 2. JWT claims set, which specifies one or more claims that identify requests from a particular consumer application. The claims are a set of name-value pairs that provide sufficient information about the application. Example: sub = Administrator.\n\n This attribute is queried when the Identify and Authorize Application policy is configured to identify consumer applications by JSON Web Token (JWT).\n\n 3. Consumer certificate, which specifies the X.509 certificates that identify requests from a particular consumer.\n\n This attribute is queried when the Identify and authenticate consumer policy is configured to identify the consumer applications by a consumer certificate.\n\n 4. Identification token, which specifies the host names, user names or other distinguishing strings that identify requests from a particular consumer application.\n\n This attribute is queried when the Identify and authenticate consumer policy action is configured to identify consumer applications by host name, HTTP user name, and WSS user name.\n\n \n\n As an API provider or an API Gateway Administrator you can create and manage applications, and register applications with the APIs.\n\n These are the high level stages of managing and using an application:\n\n 1. API developers request the API Gateway administrators to create an application for access as per the required identification criteria.\n\n 2. API Gateway provider or administrator validates the request and creates a new application, there by provisioning the application specific access tokens (API access key and OAuth credentials).\n\n 3. API Developer, upon finding a suitable API, sends a request to API Gateway for consumption by providing the application details.\n\n 4. After validating the request, API Gateway provider or administrator associates the application with the API. Keys are generated for applications and not for every API that the application consumes. \n\n Note: The approval process, if any, is handled by the requesting application and not handled by API Gateway.\n\n 5. The API developer can then use the application with the proper identifier (such as the access key or identifier) to access the API. \n\n Strategies: \n\n A strategy is a way to authenticate the incoming request and provides multiple authentication mechanisms or multiple authorization servers for a single authentication scheme. API Gateway identifies the application and validates the token submitted through the strategy configured in the application. You can create multiple strategies authorized by an API for an application. For example, in case of OAuth authentication scheme, you want the application to support both OKTA and PINGFederate or OKTA with multiple tenants. This can be configured as OAuth strategy for the application.",

"isActive": true,

"type": "REST",

"publishedPortals": [],

"systemVersion": 1,

"id": "2f8d2bc6-d73b-4357-b76d-af74620b4b64"

},

"responseStatus": "SUCCESS"

},

{

"api": {

"apiName": "AppMessenger",

"apiVersion": "1.0",

"isActive": true,

"type": "REST",

"publishedPortals": [],

"systemVersion": 1,

"id": "9e7db103-af5e-4f78-92c2-75b3b59006d3"

},

"responseStatus": "SUCCESS

}

]

}

**STEP 2**:

**Url**: <<protocol>>://<<host>>:<<port>>//rest/apigateway/apis/<<specific api id>>

**Method: GET**

**Accept**: application/json

**Content-type**: application/json

**Response Payload:**

{

"apiResponse": {

"api": {

"apiDefinition": {

"type": "rest",

"info": {

"version": "1.0",

"title": "AppMessenger"

},

"serviceRegistryDisplayName": "AppMessenger\_1.0",

"schemes": [],

"security": [],

"paths": {

"/messages": {

"get": {

"description": "This method is used to retrieve all the messages",

"operationId": "getMessage",

"parameters": [

{

"in": "query",

"name": "messageId",

"parameterSchema": {

"type": "string"

},

"required": true,

"type": "string"

}

],

"responses": {

"200": {

"description": "This status indicates that a list of available messages is retrieved successfully",

"content": {

"application/json": {

"schema": {

"type": "gateway"

},

"examples": {},

"example": "[\n {\n \"message\": \"hii\",\n \"messageId\": 1,\n \"sender\": \"salu\"\n },\n {\n \"message\": \"hello\",\n \"messageId\": 2,\n \"sender\": \"heebu\"\n }\n]"

}

},

"links": {},

"schema": {},

"examples": {},

"headers": {}

},

"401": {

"description": "This status code indicates that either user didn't provide right credentials or user doesn't have required privileges to access this API.\n",

"content": {},

"links": {},

"schema": {},

"examples": {},

"headers": {}

}

},

"mockedResponses": {},

"mockedConditionsBasedCustomResponsesList": [],

"enabled": true,

"scopes": [],

"requestBody": {

"content": {},

"name": "getMessage"

}

},

"post": {

"description": "This method is used to create a new message",

"operationId": "addMessage",

"responses": {

"200": {

"description": "This status code indicates that a message has been created successfully",

"content": {

"application/json": {

"schema": {

"type": "gateway"

},

"examples": {},

"example": "{\n \"message\": \"hurray\",\n \"messageId\": 3,\n \"sender\": \"saluma\"\n}"

}

},

"links": {},

"schema": {},

"examples": {},

"headers": {}

},

"401": {

"description": "This status code indicates that either user didn't provide right credentials or user doesn't have required privileges to access this API",

"content": {},

"links": {},

"schema": {},

"examples": {},

"headers": {}

}

},

"mockedResponses": {},

"mockedConditionsBasedCustomResponsesList": [],

"enabled": true,

"scopes": [],

"requestBody": {

"content": {

"application/json": {

"schema": {

"type": "gateway"

},

"examples": {},

"example": "{\n \"message\":\"hurray\",\n \"sender\":\"saluma\"\n }"

}

},

"name": "addMessage"

}

},

"scopes": [],

"displayName": "/messages",

"enabled": true

},

"/messages/{messageId}": {

"get": {

"description": "This method is used to retrieve a specified message",

"operationId": "getMessageById",

"responses": {

"200": {

"description": "This status code indicates that a specified message has been retrieved successfully",

"content": {

"application/json": {

"schema": {

"type": "gateway"

},

"examples": {},

"example": "{\n \"message\": \"wassup\",\n \"messageId\": 3,\n \"sender\": \"reshu\"\n}"

}

},

"links": {},

"schema": {},

"examples": {},

"headers": {}

},

"401": {

"description": "This status code indicates that either user didn't provide right credentials or user doesn't have required privileges to access this API",

"content": {},

"links": {},

"schema": {},

"examples": {},

"headers": {}

}

},

"mockedResponses": {},

"mockedConditionsBasedCustomResponsesList": [],

"enabled": true,

"scopes": [],

"requestBody": {

"content": {},

"name": "getMessageById"

}

},

"put": {

"description": "This method is used to update a specified message",

"operationId": "updateMessage",

"responses": {

"200": {

"description": "This status indicates that a message has been updated successfully",

"content": {

"application/json": {

"schema": {

"type": "gateway"

},

"examples": {},

"example": "{\n \"message\": \"wassup\",\n \"messageId\": 3,\n \"sender\": \"reshma\"\n}"

}

},

"links": {},

"schema": {},

"examples": {},

"headers": {}

},

"401": {

"description": "This status code indicates that either user didn't provide right credentials or user doesn't have required privileges to access this API",

"content": {},

"links": {},

"schema": {},

"examples": {},

"headers": {}

}

},

"mockedResponses": {},

"mockedConditionsBasedCustomResponsesList": [],

"enabled": true,

"scopes": [],

"requestBody": {

"content": {

"application/json": {

"schema": {

"type": "gateway"

},

"examples": {},

"example": " {\n \"message\": \"wassup\",\n \"sender\": \"reshma\"\n }"

}

},

"name": "updateMessage"

}

},

"delete": {

"description": "This method is used to delete a message from the list of messages",

"operationId": "deleteMessage",

"responses": {

"204": {

"description": "This status code indicates that a message has been deleted successfully",

"content": {},

"links": {},

"schema": {},

"examples": {},

"headers": {}

},

"401": {

"description": "This status code indicates that either user didn't provide right credentials or user doesn't have required privileges to access this API",

"content": {},

"links": {},

"schema": {},

"examples": {},

"headers": {}

},

"404": {

"description": "This status indicates that the specified resource is not found",

"content": {},

"links": {},

"schema": {},

"examples": {},

"headers": {}

}

},

"mockedResponses": {},

"mockedConditionsBasedCustomResponsesList": [],

"enabled": true,

"scopes": [],

"requestBody": {

"content": {},

"name": "deleteMessage"

}

},

"parameters": [

{

"in": "path",

"name": "messageId",

"parameterSchema": {

"type": "string"

},

"required": false,

"schema": {

"type": "string"

},

"type": "string"

}

],

"scopes": [],

"displayName": "/messages/{messageId}",

"enabled": true

}

},

"securityDefinitions": {},

"definitions": {},

"parameters": {},

"baseUriParameters": [],

"externalDocs": [],

"servers": [

{

"url": "http://localhost:8089/AppMessenger/webapi",

"variables": {}

}

],

"components": {

"schemas": {},

"responses": {},

"parameters": {},

"examples": {},

"requestBodies": {},

"headers": {},

"links": {},

"callbacks": {}

}

},

"nativeEndpoint": [

{

"passSecurityHeaders": true,

"uri": "http://localhost:8089/AppMessenger/webapi",

"connectionTimeoutDuration": 0,

"alias": false

}

],

"apiName": "AppMessenger",

"apiVersion": "1.0",

"maturityState": "Beta",

"isActive": true,

"type": "REST",

"owner": "Administrator",

"policies": [

"6bb201be-06a0-475f-842a-9a81245a3e98"

],

"scopes": [],

"publishedPortals": [],

"creationDate": "2019-03-18 05:23:51 GMT",

"lastModified": "2019-03-21 11:11:34 GMT",

"systemVersion": 1,

"id": "9e7db103-af5e-4f78-92c2-75b3b59006d3"

},

"responseStatus": "SUCCESS",

"gatewayEndPoints": [

"http://SAG-6HYB152:5555/gateway/AppMessenger/1.0"

],

"versions": [

{

"versionNumber": "1.0",

"apiId": "9e7db103-af5e-4f78-92c2-75b3b59006d3"

}

]

}

}

**STEP 3:**

**url:** <<protocol>>://<<host>>:<<port>>/rest/apigateway/policies/<<policy id of the specific api>>

{

**Method: GET**

**Accept**: application/json

**Content-type**: application/json

**Response payload:**

{

"policy": {

"id": "6bb201be-06a0-475f-842a-9a81245a3e98",

"names": [

{

"value": "Default Policy for API AppMessenger",

"locale": "English"

}

],

"descriptions": [

{

"value": "",

"locale": "English"

}

],

"scope": {

"scopeConditions": []

},

"policyEnforcements": [

{

"enforcements": [

{

"enforcementObjectId": "078610c9-50ab-4599-b9aa-c25960a7ce6a",

"order": null

}

],

"stageKey": "routing"

},

{

"enforcements": [

{

"enforcementObjectId": "7bb6bc4d-d132-4ce0-b4f7-64f51bacb074",

"order": null

}

],

"stageKey": "transport"

}

],

"policyScope": "SERVICE",

"global": false,

"active": false,

"systemPolicy": false

}

}

**STEP 4:**

**Url**: <<protocol>>://<<host>>:<<port>>//rest/apigateway/policyActions

**Method: POST**

**Accept**: application/json

**Content-type**: application/json

**Request Payload:**

{

"policyAction": {

"names": [

{

"value": "Outbound Authentication - Transport",

"locale": "en"

}

],

"templateKey": "outboundTransportAuthentication",

"parameters": [

{

"templateKey": "transportSecurity",

"parameters": [

{

"templateKey": "authType",

"values": [

"HTTP\_BASIC"

]

},

{

"templateKey": "authMode",

"values": [

"NEW"

]

}

]

}

],

"active": false

}

}

**Response Payload :**

{

"policyAction": {

"id": "bf1fd0be-6b20-47f5-8bc5-c00bac219a44",

"names": [

{

"value": "Outbound Authentication - Transport",

"locale": "en"

}

],

"templateKey": "outboundTransportAuthentication",

"parameters": [

{

"templateKey": "transportSecurity",

"parameters": [

{

"templateKey": "authType",

"values": [

"HTTP\_BASIC"

]

},

{

"templateKey": "authMode",

"values": [

"NEW"

]

}

]

}

],

"active": false

}

}

**STEP 5:**

**url:** <<protocol>>://<<host>>:<<port>>/rest/apigateway/policies<<id of the corresponding policy action added>>

**Method:** PUT

**Accept**: application/json

**Content-type**: application/json

Request:

{

"policy": {

"id": "6bb201be-06a0-475f-842a-9a81245a3e98",

"names": [

{

"value": "Default Policy for API AppMessenger",

"locale": "English"

}

],

"descriptions": [

{

"value": "",

"locale": "English"

}

],

"scope": {

"scopeConditions": []

},

"policyEnforcements": [

{

"enforcements": [

{

"enforcementObjectId": "078610c9-50ab-4599-b9aa-c25960a7ce6a",

"order": null

},

{

"enforcementObjectId": "bf1fd0be-6b20-47f5-8bc5-c00bac219a44",

"order": null

}

],

"stageKey": "routing"

},

{

"enforcements": [

{

"enforcementObjectId": "7bb6bc4d-d132-4ce0-b4f7-64f51bacb074",

"order": null

}

],

"stageKey": "transport"

}

],

"policyScope": "SERVICE",

"global": false,

"active": false,

"systemPolicy": false

}

}

Response:

{

"policy": {

"id": "6bb201be-06a0-475f-842a-9a81245a3e98",

"names": [

{

"value": "Default Policy for API AppMessenger",

"locale": "English"

}

],

"descriptions": [

{

"value": "",

"locale": "English"

}

],

"scope": {

"scopeConditions": []

},

"policyEnforcements": [

{

"enforcements": [

{

"enforcementObjectId": "078610c9-50ab-4599-b9aa-c25960a7ce6a",

"order": null

},

{

"enforcementObjectId": "bf1fd0be-6b20-47f5-8bc5-c00bac219a44",

"order": null

}

],

"stageKey": "routing"

},

{

"enforcements": [

{

"enforcementObjectId": "7bb6bc4d-d132-4ce0-b4f7-64f51bacb074",

"order": null

}

],

"stageKey": "transport"

}

],

"policyScope": "SERVICE",

"global": false,

"active": false,

"systemPolicy": false

}

}

**STEP 6:**

By refreshing Api gateway,the policy actions which was added will be seen.