Weekly Progress Meeting

Date: 2024/07/23

Ibrahim RIjal

BMWLAB

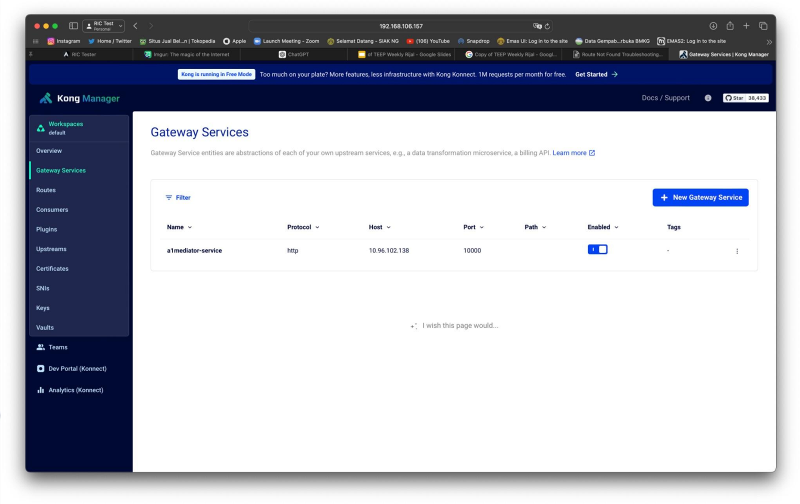
Progress so far

a1mediator Troubleshooting

Fixing where the a1mediator is inaccesible from local network
Trouble is on the built-in kong proxy inside the ricplt

Solution:

Installing new kong proxy

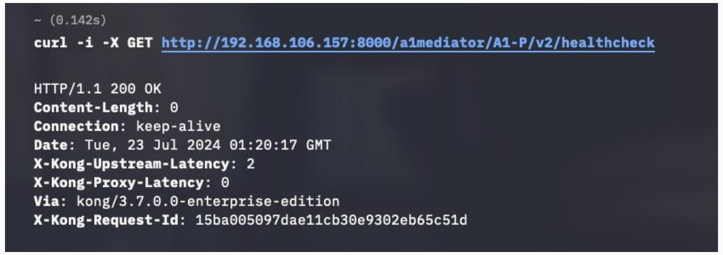


Too much on your plate? More features, less infrastructure with Kong Konnect. 1M requests per month for free.

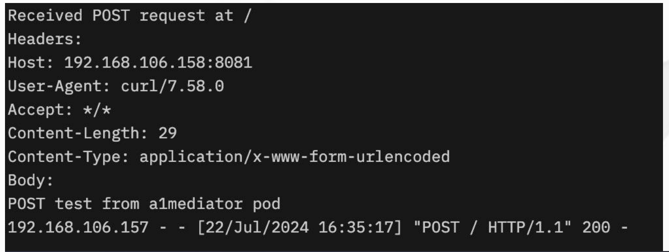
Gateway Services

Gateway Service entities are abstractions of each of your own upstream services, e.g., a data transformation microservice, a billing API. Learn more

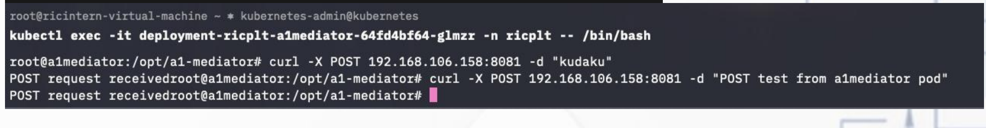
Name	Protocol	Host	Port	Path	Enabled	Tags
a1mediator-service	http	10.96.102.138	10000		●	-



```
~ (0.142s)
curl -i -X GET http://192.168.106.157:8000/a1mediator/A1-P/v2/healthcheck

HTTP/1.1 200 OK
Content-Length: 0
Connection: keep-alive
Date: Tue, 23 Jul 2024 01:20:17 GMT
X-Kong-Upstream-Latency: 2
X-Kong-Proxy-Latency: 0
Via: kong/3.7.0.0-enterprise-edition
X-Kong-Request-Id: 15ba005097dae11cb30e9302eb65c51d
```



```
Received POST request at /
Headers:
Host: 192.168.106.158:8081
User-Agent: curl/7.58.0
Accept: */*
Content-Length: 29
Content-Type: application/x-www-form-urlencoded
Body:
POST test from a1mediator pod
192.168.106.157 - - [22/Jul/2024 16:35:17] "POST / HTTP/1.1" 200 -
```



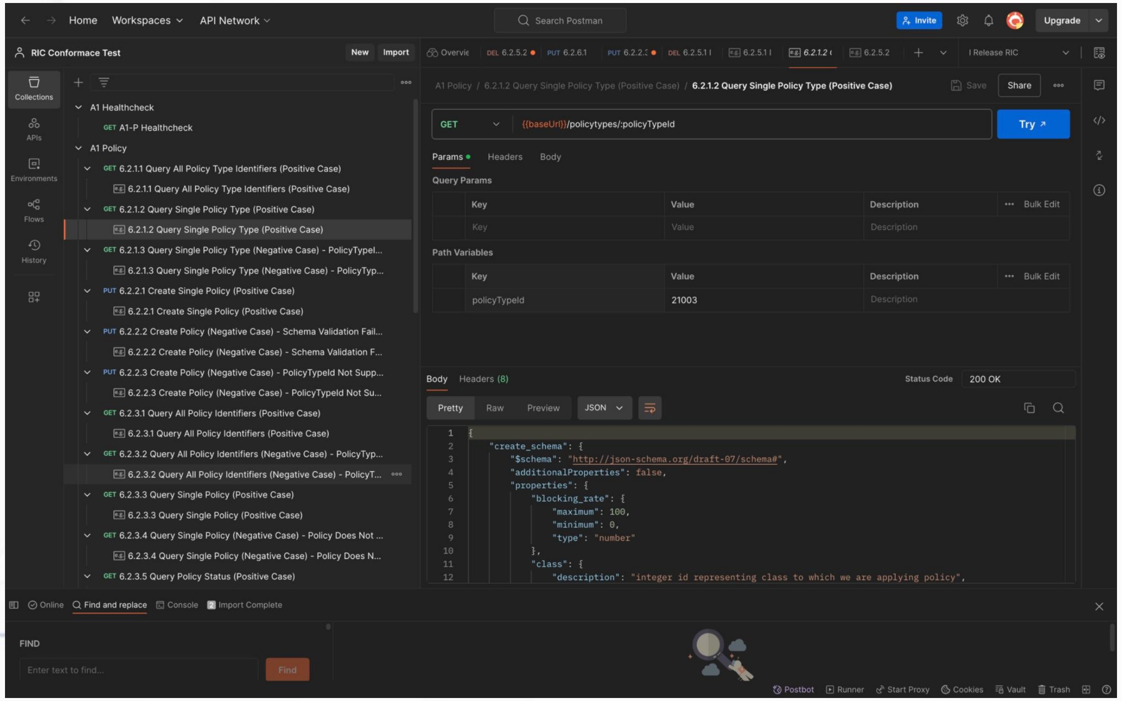
```
root@ricintern-virtual-machine ~ * kubernetes-admin@kubernetes
kubectl exec -it deployment-ricplt-a1mediator-64fd4bf64-glmzr -n ricplt -- /bin/bash

root@a1mediator:/opt/a1-mediator# curl -X POST 192.168.106.158:8081 -d "kudaku"
POST request receivedroot@a1mediator:/opt/a1-mediator# curl -X POST 192.168.106.158:8081 -d "POST test from a1mediator pod"
POST request receivedroot@a1mediator:/opt/a1-mediator#
```

https://hackmd.io/@irphotoarts/r1Z0XYhOA

Created Test Cases Based on A1TS

Created test cases using Postman to test the a1mediator API.

The Near-RT RIC is the DUT.



RIC Conformace Test

- A1 Healthcheck
  - GET A1-P Healthcheck
- A1 Policy
  - GET 6.2.1.1 Query All Policy Type Identifiers (Positive Case)
    - 6.2.1.1 Query All Policy Type Identifiers (Positive Case)
  - GET 6.2.1.2 Query Single Policy Type (Positive Case)
    - 6.2.1.2 Query Single Policy Type (Positive Case)
  - GET 6.2.1.3 Query Single Policy Type (Negative Case) - PolicyTypeI...
    - 6.2.1.3 Query Single Policy Type (Negative Case) - PolicyTyp...
  - PUT 6.2.2.1 Create Single Policy (Positive Case)
    - 6.2.2.1 Create Single Policy (Positive Case)
  - PUT 6.2.2.2 Create Policy (Negative Case) - Schema Validation Fail...
    - 6.2.2.2 Create Policy (Negative Case) - Schema Validation F...
  - PUT 6.2.2.3 Create Policy (Negative Case) - PolicyTypeId Not Supp...
    - 6.2.2.3 Create Policy (Negative Case) - PolicyTypeId Not Su...
  - GET 6.2.3.1 Query All Policy Identifiers (Positive Case)
    - 6.2.3.1 Query All Policy Identifiers (Positive Case)
  - GET 6.2.3.2 Query All Policy Identifiers (Negative Case) - PolicyTyp...
    - 6.2.3.2 Query All Policy Identifiers (Negative Case) - PolicyT...
  - GET 6.2.3.3 Query Single Policy (Positive Case)
    - 6.2.3.3 Query Single Policy (Positive Case)
  - GET 6.2.3.4 Query Single Policy (Negative Case) - Policy Does Not ...
    - 6.2.3.4 Query Single Policy (Negative Case) - Policy Does N...
  - GET 6.2.3.5 Query Policy Status (Positive Case)

A1 Policy / 6.2.1.2 Query Single Policy Type (Positive Case) / **6.2.1.2 Query Single Policy Type (Positive Case)**

GET {{baseUrl}}/policytypes/:policyTypeId

Query Params

Key	Value	Description
Key	Value	Description

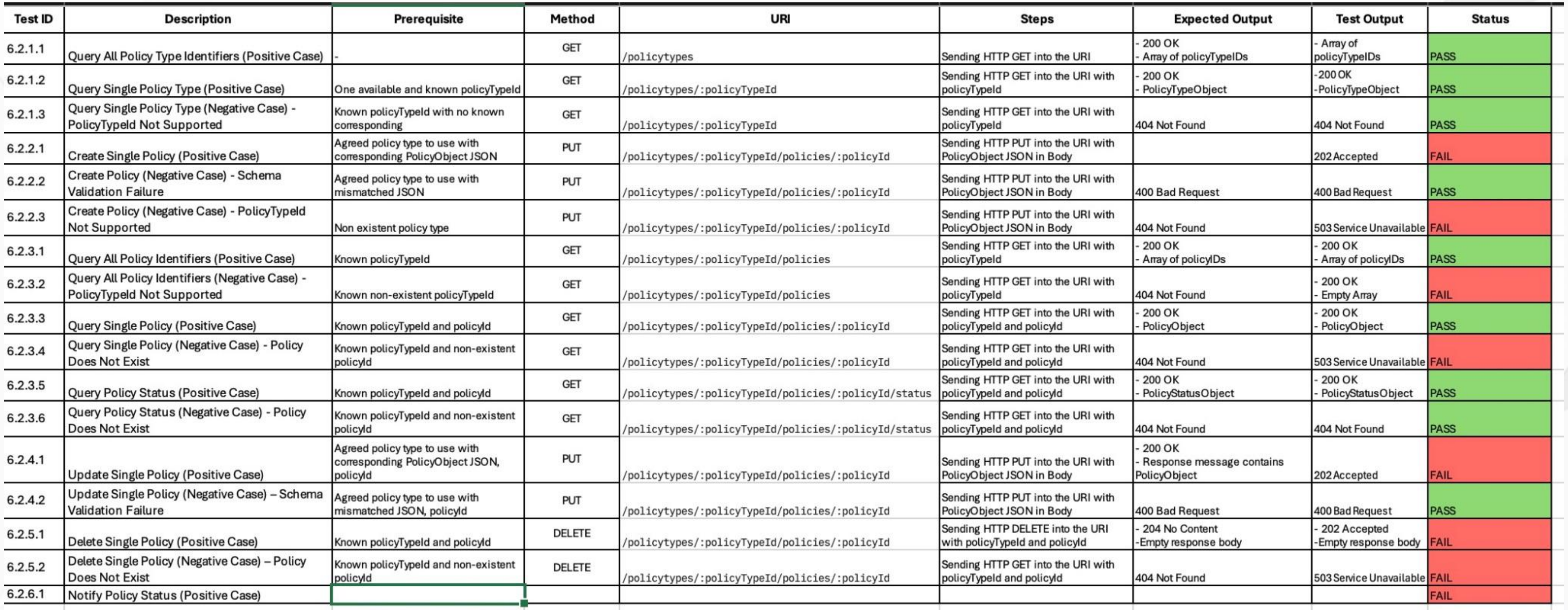
Path Variables

Key	Value	Description
policyTypeId	21003	Description

Status Code 200 OK

```
{
    "create_schema": {
        "$schema": "http://json-schema.org/draft-07/schema#",
        "additionalProperties": false,
        "properties": {
            "blocking_rate": {
                "maximum": 100,
                "minimum": 0,
                "type": "number"
            },
            "class": {
                "description": "integer id representing class to which we are applying policy",
```

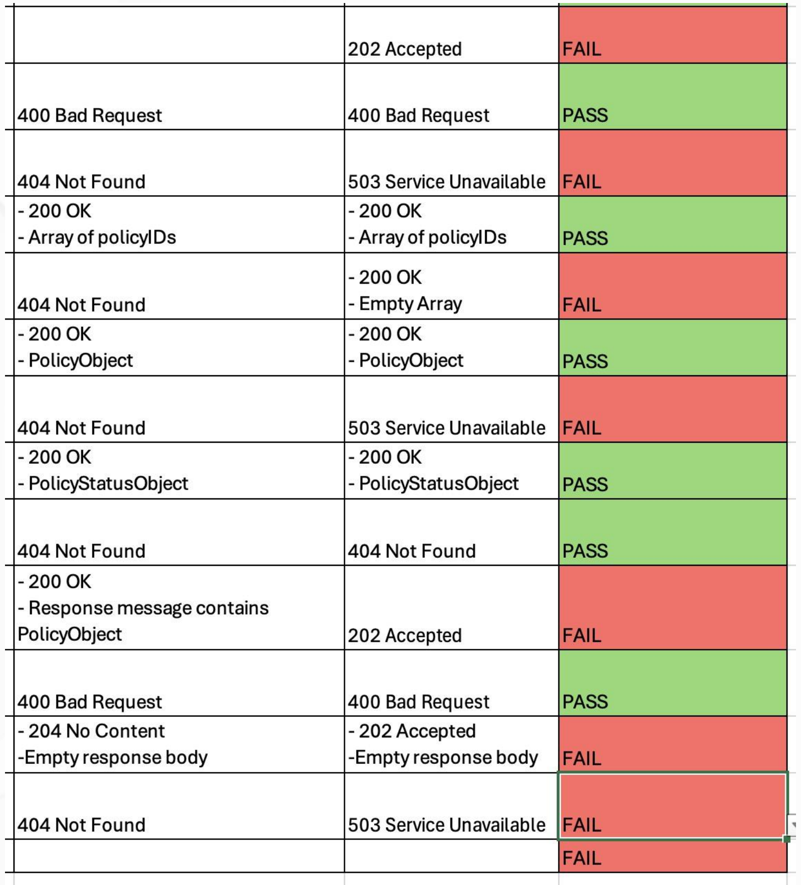
Doing Tests based on Created Test Cases



Test ID	Description	Prerequisite	Method	URI	Steps	Expected Output	Test Output	Status
6.2.1.1	Query All Policy Type Identifiers (Positive Case)	-	GET	/policytypes	Sending HTTP GET into the URI	- 200 OK - Array of policyTypeIDs	- Array of policyTypeIDs	PASS
6.2.1.2	Query Single Policy Type (Positive Case)	One available and known policyTypeId	GET	/policytypes/:policyTypeId	Sending HTTP GET into the URI with policyTypeId	- 200 OK - PolicyTypeObject	-200 OK -PolicyTypeObject	PASS
6.2.1.3	Query Single Policy Type (Negative Case) - PolicyTypeId Not Supported	Known policyTypeId with no known corresponding	GET	/policytypes/:policyTypeId	Sending HTTP GET into the URI with policyTypeId	404 Not Found	404 Not Found	PASS
6.2.2.1	Create Single Policy (Positive Case)	Agreed policy type to use with corresponding PolicyObject JSON	PUT	/policytypes/:policyTypeId/policies/:policyId	Sending HTTP PUT into the URI with PolicyObject JSON in Body		202 Accepted	FAIL
6.2.2.2	Create Policy (Negative Case) - Schema Validation Failure	Agreed policy type to use with mismatched JSON	PUT	/policytypes/:policyTypeId/policies/:policyId	Sending HTTP PUT into the URI with PolicyObject JSON in Body	400 Bad Request	400 Bad Request	PASS
6.2.2.3	Create Policy (Negative Case) - PolicyTypeId Not Supported	Non existent policy type	PUT	/policytypes/:policyTypeId/policies/:policyId	Sending HTTP PUT into the URI with PolicyObject JSON in Body	404 Not Found	503 Service Unavailable	FAIL
6.2.3.1	Query All Policy Identifiers (Positive Case)	Known policyTypeId	GET	/policytypes/:policyTypeId/policies	Sending HTTP GET into the URI with policyTypeId	- 200 OK - Array of policyIDs	- 200 OK - Array of policyIDs	PASS
6.2.3.2	Query All Policy Identifiers (Negative Case) - PolicyTypeId Not Supported	Known non-existent policyTypeId	GET	/policytypes/:policyTypeId/policies	Sending HTTP GET into the URI with policyTypeId	404 Not Found	- 200 OK - Empty Array	FAIL
6.2.3.3	Query Single Policy (Positive Case)	Known policyTypeId and policyId	GET	/policytypes/:policyTypeId/policies/:policyId	Sending HTTP GET into the URI with policyTypeId and policyId	- 200 OK - PolicyObject	- 200 OK - PolicyObject	PASS
6.2.3.4	Query Single Policy (Negative Case) - Policy Does Not Exist	Known policyTypeId and non-existent policyId	GET	/policytypes/:policyTypeId/policies/:policyId	Sending HTTP GET into the URI with policyTypeId and policyId	404 Not Found	503 Service Unavailable	FAIL
6.2.3.5	Query Policy Status (Positive Case)	Known policyTypeId and policyId	GET	/policytypes/:policyTypeId/policies/:policyId/status	Sending HTTP GET into the URI with policyTypeId and policyId	- 200 OK - PolicyStatusObject	- 200 OK - PolicyStatusObject	PASS
6.2.3.6	Query Policy Status (Negative Case) - Policy Does Not Exist	Known policyTypeId and non-existent policyId	GET	/policytypes/:policyTypeId/policies/:policyId/status	Sending HTTP GET into the URI with policyTypeId and policyId	404 Not Found	404 Not Found	PASS
6.2.4.1	Update Single Policy (Positive Case)	Agreed policy type to use with corresponding PolicyObject JSON, policyId	PUT	/policytypes/:policyTypeId/policies/:policyId	Sending HTTP PUT into the URI with PolicyObject JSON in Body	- 200 OK - Response message contains PolicyObject	202 Accepted	FAIL
6.2.4.2	Update Single Policy (Negative Case) – Schema Validation Failure	Agreed policy type to use with mismatched JSON, policyId	PUT	/policytypes/:policyTypeId/policies/:policyId	Sending HTTP PUT into the URI with PolicyObject JSON in Body	400 Bad Request	400 Bad Request	PASS
6.2.5.1	Delete Single Policy (Positive Case)	Known policyTypeId and policyId	DELETE	/policytypes/:policyTypeId/policies/:policyId	Sending HTTP DELETE into the URI with policyTypeId and policyId	- 204 No Content -Empty response body	- 202 Accepted -Empty response body	FAIL
6.2.5.2	Delete Single Policy (Negative Case) – Policy Does Not Exist	Known policyTypeId and non-existent policyId	DELETE	/policytypes/:policyTypeId/policies/:policyId	Sending HTTP GET into the URI with policyTypeId and policyId	404 Not Found	503 Service Unavailable	FAIL
6.2.6.1	Notify Policy Status (Positive Case)							FAIL

https://univindonesia-my.sharepoint.com/:x:/g/personal/ibrahim_rijal_office_ui_ac_id/EXQZNXe_kdJGn0vjwijiZZEBgJq9ve6YZFQ4dHkIWXSsnQ?e=rkhuTA

Several Tests Fail

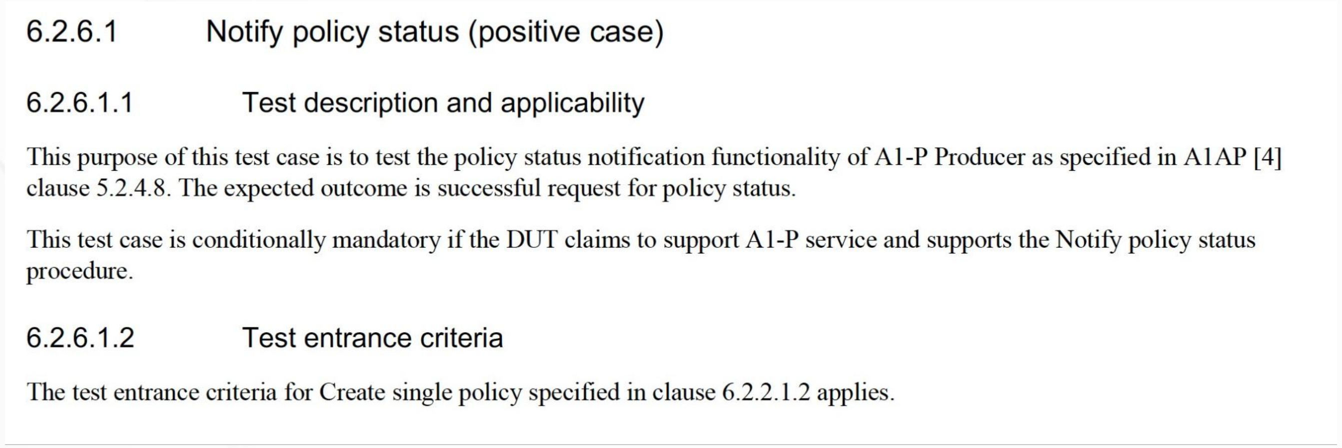


	202 Accepted	FAIL
400 Bad Request	400 Bad Request	PASS
404 Not Found	503 Service Unavailable	FAIL
- 200 OK - Array of policyIDs	- 200 OK - Array of policyIDs	PASS
404 Not Found	- 200 OK - Empty Array	FAIL
- 200 OK - PolicyObject	- 200 OK - PolicyObject	PASS
404 Not Found	503 Service Unavailable	FAIL
- 200 OK - PolicyStatusObject	- 200 OK - PolicyStatusObject	PASS
404 Not Found	404 Not Found	PASS
- 200 OK - Response message contains PolicyObject	202 Accepted	FAIL
400 Bad Request	400 Bad Request	PASS
- 204 No Content -Empty response body	- 202 Accepted -Empty response body	FAIL
404 Not Found	503 Service Unavailable	FAIL
		FAIL

- Fail mostly regarding to the Response code
- Did not notice any functionality issues

Problem

6.2.6.1 Problem



6.2.6.1 Notify policy status (positive case)

6.2.6.1.1 Test description and applicability

This purpose of this test case is to test the policy status notification functionality of A1-P Producer as specified in A1AP [4] clause 5.2.4.8. The expected outcome is successful request for policy status.

This test case is conditionally mandatory if the DUT claims to support A1-P service and supports the Notify policy status procedure.

6.2.6.1.2 Test entrance criteria

The test entrance criteria for Create single policy specified in clause 6.2.2.1.2 applies.

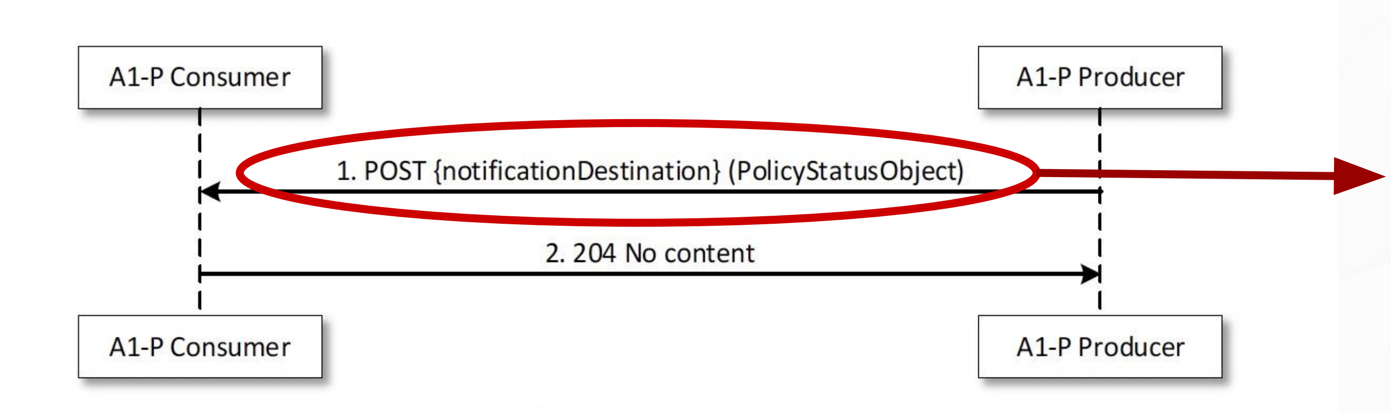
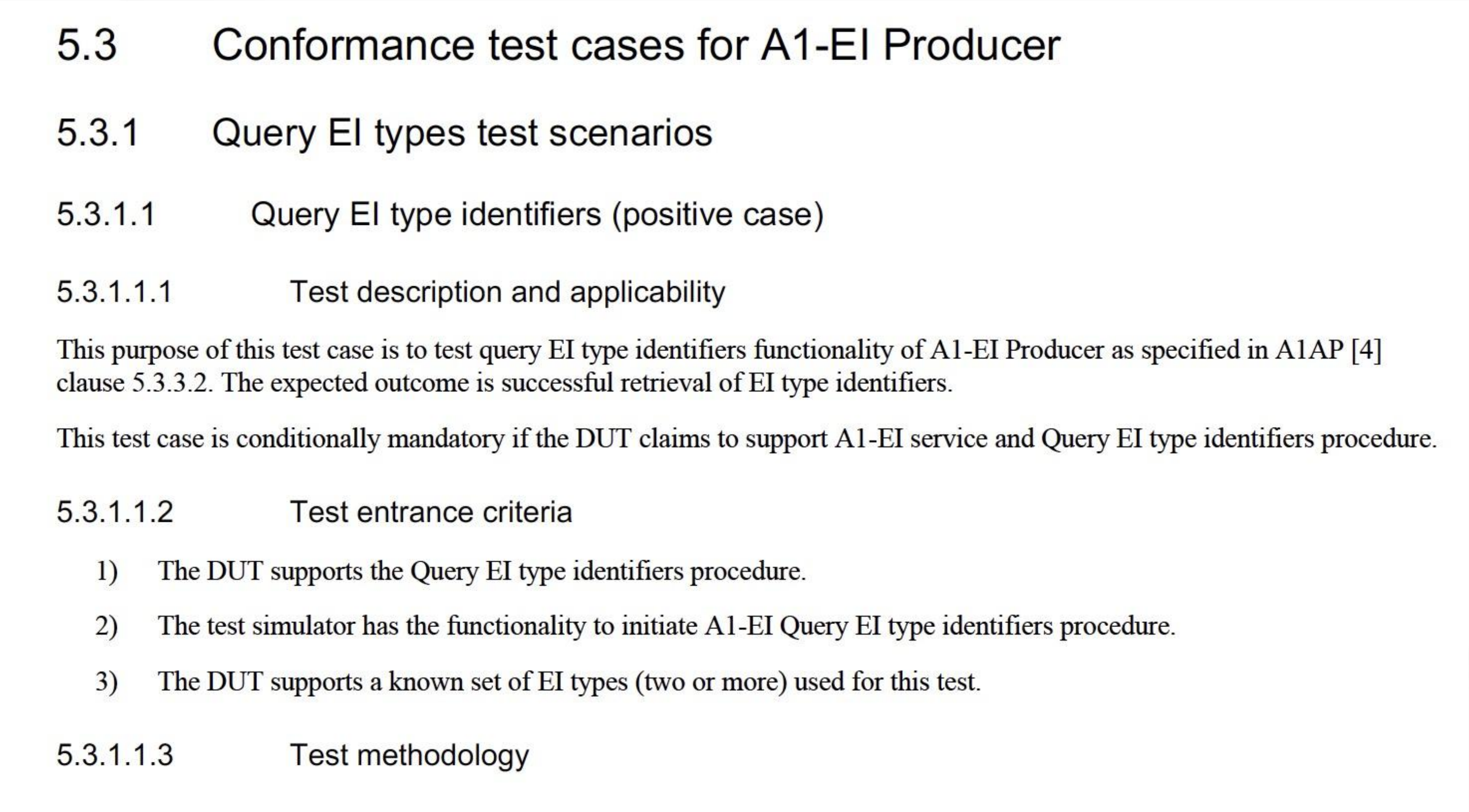


Figure 5.2.4.8.1-1 Notify policy status operation

Not Working

Next

Testing for the A1-EI Producer



5.3 Conformance test cases for A1-EI Producer

5.3.1 Query EI types test scenarios

5.3.1.1 Query EI type identifiers (positive case)

5.3.1.1.1 Test description and applicability

This purpose of this test case is to test query EI type identifiers functionality of A1-EI Producer as specified in A1AP [4] clause 5.3.3.2. The expected outcome is successful retrieval of EI type identifiers.

This test case is conditionally mandatory if the DUT claims to support A1-EI service and Query EI type identifiers procedure.

5.3.1.1.2 Test entrance criteria

1) The DUT supports the Query EI type identifiers procedure.
2) The test simulator has the functionality to initiate A1-EI Query EI type identifiers procedure.
3) The DUT supports a known set of EI types (two or more) used for this test.

5.3.1.1.3 Test methodology