Assignment Task 3: Api testing in Jmeter

For this task the website "https://www.videogamedb.uk/swagger-ui/index.html#/" was provided. The swagger showed the following api and some of the possible response message and status codes

1. api/authenticate → used to provide auth token that we are going to pass in other api
   Expected Status Codes:
   200 → OK
   400 → Invalid Username or Password

2. v1 api

   1. GET /api/videogame → provides all the list of videogames
      Expected Status Codes:
      200 → OK

   2. POST /api/videogame → creates a new videogame
      200 → OK
      400 → Invalid Request

   3. GET /api/videogame/{id} → get single videogame based on ID provided
      200 → OK
      404 → Video game not found

   4. PUT /api/videogame/{id} → update the videgame whose ID is provided
      200 → OK
      400 → Invalid Request
      404 → Video game not found
   5. DELETE /api/videogame/{id} → Delet the video game whose ID is provided
      200 → OK
      404 → Video game not found
3. v2 api

   1. GET /api/v2/videogame → provides all the list of videogames
      Expected Status Codes:
      200 → OK

   2. POST /api/v2/videogame → creates a new videogame
      200 → OK
      400 → Invalid Request

   3. GET /api/v2/videogame/{id} → get single videogame based on ID provided
      200 → OK
      404 → Video game not found

   4. PUT /api/v2/videogame/{id} → update the videgame whose ID is provided
      200 → OK
      400 → Invalid Request

404 → Video game not found

5. DELETE /api/v2/videogame/{id} → Delet the video game whose ID is provided
   200 → OK
   404 → Video game not found

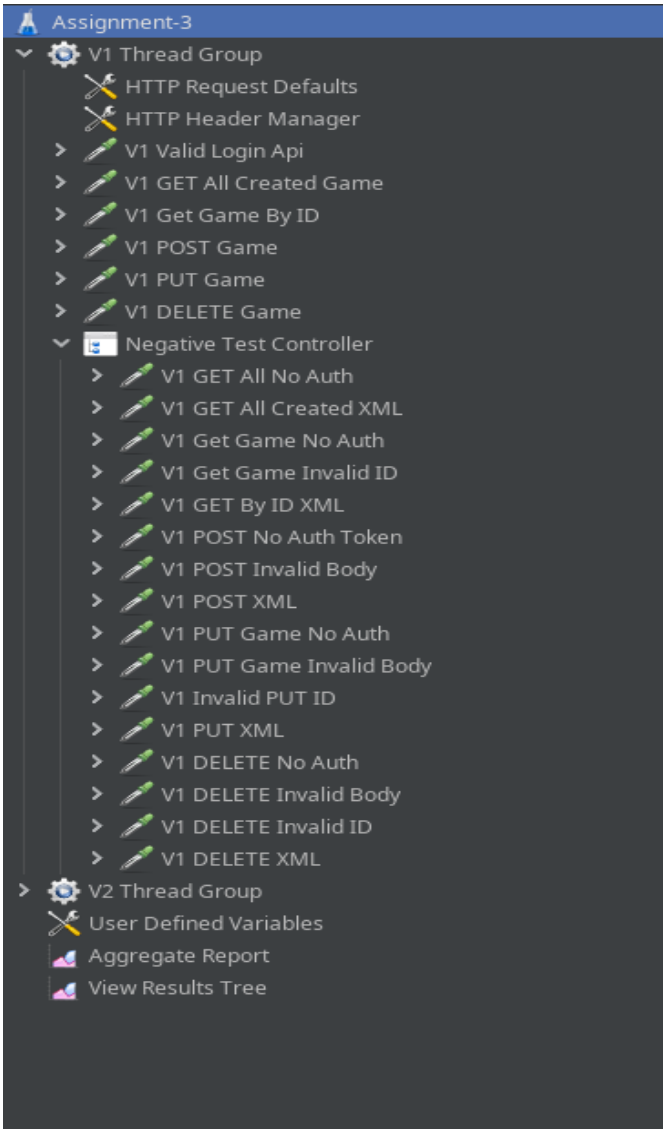
In addition to application/json v2 also accepts application/xml

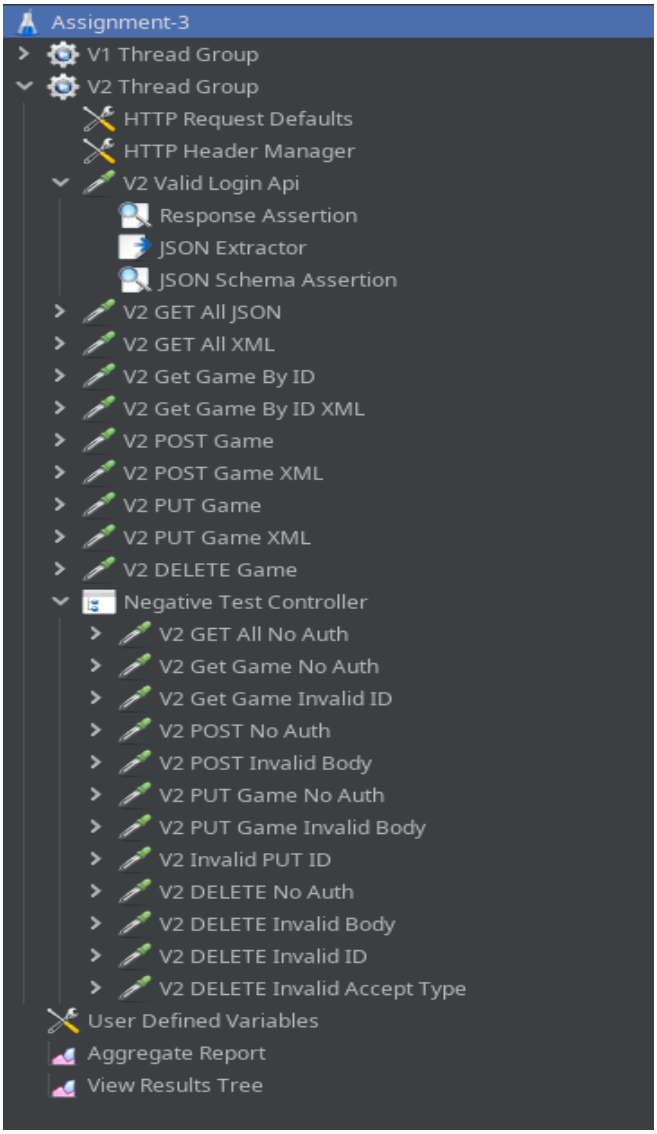
Test Plan

For testing V1 and V2 I created two thread groups(1 threads and 1 second Ramp up time with no loop). Then each thread group had global HTTP Request Defaults and HTTP Header Manager

Also User Defined Variables was used for:
baseUrl → baseUrl for v1 and v2 → www.videogamedb.uk
gameId → to be used in put post and get by Id methods → 1
schemaPath → for JSON schema validation → path where schema folder is located

Overall Structure





Cases:

V1 Thread Group

HTTP Request Defaults has following properties
protocol: https
server name: ${baseUrl}// User Defined Variable
port: 443

HTTP Header Manager has:
Content-Type and Accept both set to application/json as per the documentation

API Requests
Valid login api to fetch token then use **JSON Extractor** to assign it to ${token} variable

All following requests use the HTTP header **Authorization** with value Beare ${token}

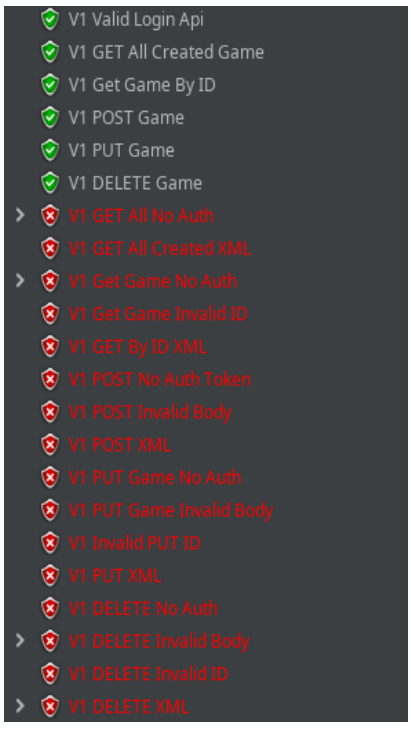
Then all the valid requests utilize the JSON schema **assertion**
(Plugin: **https://github.com/kde-intro/jmeter-json-schema-assertion-plugin**)

It also uses Response Assertion with status code and the DELETE request uses Response Assertion with text response.

The Negative Test Controller contains all API request that are bound to fail by logic comprising of no auth header, invalid body or invalid ID.
Also it validates if application/xml is a valid response or not

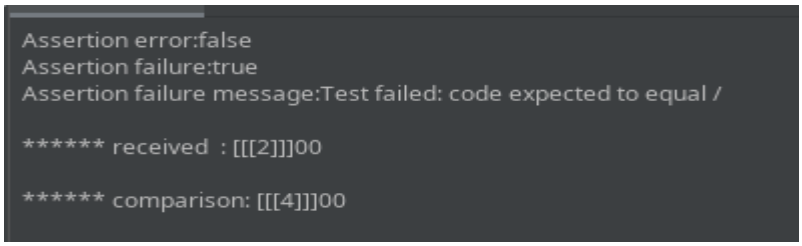
Similar approach is used for Thread Group V2

Result and Analysis

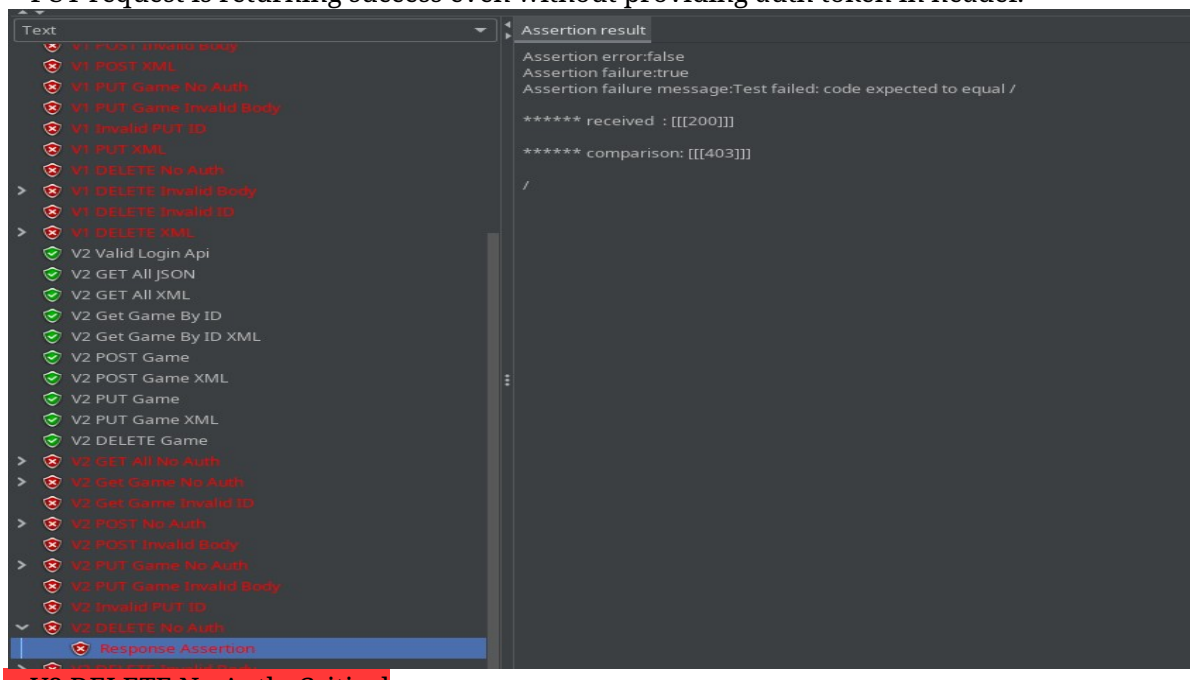


Failures:

1. **V1 Get All No Auth**: The api document explicitly does not include anywhere that GET api can return success without providing auth header so in order to test that GET all games request was sent without authorization header and expected Error:403 forbidden but received 200



2. Same case for **GET with ID**
3. **V1 Delete Invalid Body**
   The DELETE request does not require body parameter from the api documentaion. However even on sending a body it send success so there is a possibility that the api request is ignoring the body or this might not be the case so it is flagged.
4. The api v1 doesnot provide info on it accepting application/xml so **DELETE** was expected to return 406 as in other requests. However it sent success 200.
5. V2 GET all No Auth and V2 Get by ID
   Same reason as for no auth in v1
6. V2 POST No Auth: Critical
   POST request is returning success even without providing auth token in header.
7. V2 PUT No Auth: Critical
   PUT request is returning success even without providing auth token in header.



8. V2 DELETE No Auth: Critical
   DELETE request is returning success even without providing auth token in header.

9. **V2 Delete Invalid Body**