1. Interactive document is shared – Functionality, Explore
2. Service to test - [https://bitly.com](https://bitly.com/)
3. Link to API documentation - <https://dev.bitly.com/v4/>
4. Tests to check the following methods:

GET /groups/{group\_guid}  
GET /groups/{group\_guid}/bitlinks/{sort}  
POST /bitlinks

----Tried via Postman for hands-on

---- When method was different 403 forbidden

--- when current method is given but not much specified what is the exact issue {

    "message": "FORBIDDEN",

    "resource": "bitlinks",

    "description": "You are currently forbidden to access this resource."

}

----- POST shorten: <https://api-ssl.bitly.com/v4/shorten>

When only long\_URL is given 201 Created message displayed

1. To prepare Test cases(Excel format set), Test Data sheet also set

GET /groups/{group\_guid}:

---Calls can be made at Individual or group level. This returns details of specified group guid

GET /groups/{group\_guid}/bitlinks/{sort}

* To sort
* 403 received when different guid was used
* 404 not found when wrong Get request URI was sent

POST /bitlinks

Converts a long url to a Bitlink and sets additional parameters.

400

402

403

404

422Unprocessable Entity (WebDAV) (RFC 4918) – when tags not sent

My collection link(initial): <https://www.getpostman.com/collections/cd1314ee48a7281f838e>

My collection V0.2 : https://www.getpostman.com/collections/cd1314ee48a7281f838e

My collection V0.3 : <https://www.getpostman.com/collections/cd1314ee48a7281f838e>

Bitly API Testing :

<https://bit.ly/2WUpYw7>

Testcase:

Feature:

|  |
| --- |
| Scenario: the user makes a successful get transaction request |
|  |

|  |
| --- |
| Given the user authenticates using "<login\_request>" |
|  |

|  |
| --- |
| And the user connects to the "transaction" service |
|  |

|  |
| --- |
| When the user sends a "get transactions" request using "<transaction\_request>" |
|  |

|  |
| --- |
| Then I should get a "<response\_status>" response |
|  |

|  |
| --- |
| And the response should match "<transaction\_response>" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Examples: |
|  |

|  |
| --- |
| | login\_request | transaction\_request | response\_status | transaction\_response | |
|  |

|  |
| --- |
| | login\_request\_admin.json | get\_transactions\_request1.json | successful | transaction\_response1.json | |
|  |

|  |
| --- |
| | login\_request\_user.json | invalid\_transation\_req.json | failure | general\_server\_error.json | |
|  |

| login\_request\_super\_user.json | invalid\_token\_request.json | failure | authentication\_error.json |

Authorization:

BitlySM

<https://bitly.com/bitlySM>.

BitlySM

Client ID

70579779768db41cde732636b57b518597cb532c

Client Secret

04079ee36af9583ad0e30242651c24793a59a98d

GET testcase 1:

**package** Bitly\_APIProject;

**import** **static** org.hamcrest.Matchers.*equalTo*;

**import** org.testng.annotations.Test;

**import** io.restassured.RestAssured;

**import** io.restassured.response.ValidatableResponse;

**import** io.restassured.specification.RequestSpecification;

**public** **class** Bitlytestcase {

RequestSpecification request = RestAssured.*given*();

String tokenGenerated = "c08549d81bd4b8de53451c54ce64f22c7f134bcd";

@Test

**public** **void** test\_getgroupdetails() {

request.header("Authorization","Bearer "+tokenGenerated );

ValidatableResponse res =

request.header("Content-Type","application/json")

.when()

.get("https://api-ssl.bitly.com/v4/groups/Bl83agh17lO")

.then()

.statusCode(200)

.body("guid", *equalTo*("Bl83agh17lO"))

.body("name", *equalTo*("smrithi099"));

System.***out***.println(res.log().all());

}

**public** **void** test\_getgroupdetailserror() {

request.header("Authorization","Bearer "+tokenGenerated );

request.header("Content-Type","application/json")

.when()

.get("https://api-ssl.bitly.com/v4/groups/Ba1bc23dE4F")

.then()

.statusCode(403)

.body("message", *equalTo*("FORBIDDEN"))

.body("description",*equalTo*("You are currently forbidden to access this resource"));

}

}

GET Testcase sort:

**package** Bitly\_APIProject;

**import** org.testng.annotations.Test;

**import** io.restassured.RestAssured;

**import** io.restassured.specification.RequestSpecification;

**public** **class** Bitlytestcase2\_GET\_Sort {

RequestSpecification request = RestAssured.*given*();

String tokenGenerated = "c08549d81bd4b8de53451c54ce64f22c7f134bcd";

@Test

**public** **void** test\_getsort\_clicks() {

request.header("Authorization","Bearer "+tokenGenerated );

request.header("Content-Type","application/json")

.when()

.get("https://api-ssl.bitly.com/v4/groups/Bl83agh17lO/bitlinks/clicks")

.then()

.statusCode(200);

}

**public** **void** test\_getsort\_unitsoftime() {

request.header("Authorization","Bearer "+tokenGenerated );

request.header("Content-Type","application/json")

.when()

.get("https://api-ssl.bitly.com/v4/groups/Bl83agh17lO/bitlinks/clicks?unit=month&units=5&unit\_reference=2021-09-02T15%3A04%3A05-0700&size=10")

.then()

.statusCode(200);

}

**public** **void** test\_get\_404pagenotfound() {

request.header("Authorization","Bearer "+tokenGenerated );

request.header("Content-Type","application/json")

.when()

.get("https://api-ssl.bitly.com/v4/groups/Bl83agh17lO/bitlink/clicks?unit=month&units=5&unit\_reference=2021-09-02T15%3A04%3A05-0700&size=10")

.then()

.statusCode(404);

}

}

POST testcase:

**package** Bitly\_APIProject;

**import** **static** org.hamcrest.Matchers.*equalTo*;

**import** java.io.File;

**import** org.testng.annotations.Test;

**import** io.restassured.RestAssured;

**import** io.restassured.specification.RequestSpecification;

**public** **class** BitlyPostTestcase3File {

RequestSpecification request = RestAssured.*given*();

String tokenGenerated = "c08549d81bd4b8de53451c54ce64f22c7f134bcd";

@Test

**public** **void** test\_PostlongURLtoBitlinksuccess() {

request.header("Authorization","Bearer "+tokenGenerated );

request.header("Content-Type","application/json");

request.body(**new** File("./Payload.json")).log().all()

.when()

.post("https://api-ssl.bitly.com/v4/bitlinks")

.then()

.statusCode(200)

.body("long\_url", *equalTo*("https://dev.bitly.com/"));

}

}