

Aim: To perform to send XML payload in REST assured, parse XML response in REST assured, parse JSON response in REST assured, and explore native logging of REST assured.

CODE:

Code for JSON

```
package restAssurescripts;

import org.json.JSONObject;
import org.testng.annotations.Test;

import io.restassured.RestAssured;
import io.restassured.http.ContentType;
import static org.hamcrest.Matchers.equalTo;

public class Script004PayloadJSON {

    @Test
    public void postusingJSON()
    {

        JSONObject body = new JSONObject();
        body.put("name", "Siva");
        body.put("salary", "4567");
        body.put("age", "23");
```

```

    RestAssured.given()

        .baseUrl("https://dummy.restapiexample.com/api/v1")

        .contentType(ContentType.JSON)

        .body(body.toString())

        .when().post("/create")

        .then()

        .statusCode(200)

        .log().all()

        .body("data.name", equalTo("Siva"))

        .extract().path("data.id");

    }

}

```

Code for XML

```

package restAssurescripts;

import org.hamcrest.Matchers;
import org.testng.annotations.Test;

import io.restassured.RestAssured;
import io.restassured.http.ContentType;

public class Script005PayloadXML {

    @Test

    public void payloadXML()

```

```
{
```

```
String xmlrequestBody = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\r\n"
    + "<Pet>\r\n"
    + "    <id>0</id>\r\n"
    + "    <Category>\r\n"
    + "        <id>0</id>\r\n"
    + "        <name>string</name>\r\n"
    + "    </Category>\r\n"
    + "    <name>doggie</name>\r\n"
    + "    <photoUrls>\r\n"
    + "        <photoUrl>string</photoUrl>\r\n"
    + "    </photoUrls>\r\n"
    + "    <tags>\r\n"
    + "        <Tag>\r\n"
    + "            <id>0</id>\r\n"
    + "            <name>string</name>\r\n"
    + "        </Tag>\r\n"
    + "    </tags>\r\n"
    + "    <status>available</status>\r\n"
    + "</Pet>";
```

```
RestAssured.given()
    .baseUrl("https://petstore.swagger.io")
    .basePath("/v2/pet")
    .contentType(ContentType.XML)
    .accept("application/xml")
    .body(xmlrequestBody)
    .when().post() // execute post request
    .then()
    .statusCode(200)
```

```
.body("Pet.name", Matchers.equalTo("doggie"))  
.log().all();
```

```
}
```

```
}
```

OUTPUTS:





