# Rest Assured - Sample JSON and JsonPath Usage

## Sample JSON Response (sample.json)

{  
 "status": "success",  
 "data": {  
 "id": 101,  
 "name": "John Doe",  
 "email": "john.doe@example.com",  
 "roles": ["admin", "editor"],  
 "address": {  
 "city": "New York",  
 "zipcode": "10001"  
 },  
 "projects": [  
 {  
 "projectId": 1,  
 "projectName": "Alpha"  
 },  
 {  
 "projectId": 2,  
 "projectName": "Beta"  
 }  
 ]  
 }  
}

## Maven Dependency

<dependency>  
 <groupId>io.rest-assured</groupId>  
 <artifactId>rest-assured</artifactId>  
 <version>5.4.0</version>  
 <scope>test</scope>  
</dependency>

## Java Code Example Using JsonPath

import io.restassured.path.json.JsonPath;  
import org.testng.annotations.Test;  
  
import java.io.File;  
import java.util.List;  
  
public class JsonPathExample {  
  
 @Test  
 public void readJsonData() {  
 File file = new File("src/test/resources/sample.json");  
 JsonPath jsonPath = new JsonPath(file);  
  
 String status = jsonPath.getString("status");  
 int id = jsonPath.getInt("data.id");  
 String email = jsonPath.getString("data.email");  
 List<String> roles = jsonPath.getList("data.roles");  
 String city = jsonPath.getString("data.address.city");  
 String firstProjectName = jsonPath.getString("data.projects[0].projectName");  
 int secondProjectId = jsonPath.getInt("data.projects[1].projectId");  
  
 System.out.println("Status: " + status);  
 System.out.println("ID: " + id);  
 System.out.println("Email: " + email);  
 System.out.println("Roles: " + roles);  
 System.out.println("City: " + city);  
 System.out.println("First Project Name: " + firstProjectName);  
 System.out.println("Second Project ID: " + secondProjectId);  
 }  
}

## JsonPath Expressions Summary

|  |  |
| --- | --- |
| Expression | Meaning |
| status | Root level string |
| data.id | Nested key |
| data.roles[0] | First item in roles array |
| data.projects[1].projectId | Second object’s projectId |
| data.address.city | City inside nested object |