API AUTOMATION - RESTASSURED

Quick Learning Guide

🧮 API Automation – Day 1

Topic: REST API Basics (GET, POST, PUT, DELETE)

📌 Day 1 Agenda:

1 Introduction to APIs

- What is an API?
- What is REST?
- REST vs. SOAP
- Key HTTP Methods

2 Understanding HTTP Methods

- GET
- POST
- PUT
- DELETE

3 Working with REST APIs using Postman

- Setting Up Postman
- Sending GET, POST, PUT, DELETE Requests

4 Introduction to REST Assured for API Testing

- Setting Up RestAssured
- Writing Basic API Test Cases

5 Day 1 Exercises
6 Best Practices
7 Possible Interview Questions

---

1 Introduction to APIs

📌 **What is an API?**
API (Application Programming Interface) allows two systems to communicate with each other.

📌 **What is REST?**

REST (Representational State Transfer) is an architectural style for building APIs.

📌 **REST vs. SOAP:**

Feature	REST	SOAP
Protocol	HTTP	HTTP, SMTP, TCP
Format	JSON, XML	XML
Performance	Faster	Slower
Security	OAuth, JWT	WS-Security
Usage	Web Services, Microservices	Legacy systems, Banking apps

📌 **Key HTTP Methods:**

Method	Purpose
GET	Retrieve data
POST	Create new resource
PUT	Update existing resource
DELETE	Remove resource

---

2 Understanding HTTP Methods

📌 **2.1 GET Request (Fetching Data)**

- Used to retrieve data from a server
- No request body required

Example API Endpoint (Fetching Users)

```
GET https://reqres.in/api/users?page=2
```

Example Response (JSON)

```
{
  "page": 2,
  "data": [
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
      "first_name": "Michael",
      "last_name": "Lawson"
    }
  ]
}
```

📌 2.2 POST Request (Creating Data)

- Used to create a new resource
- Requires a **request body**

Example API Endpoint (Creating User)

```
POST https://reqres.in/api/users
```

Request Body

```
{
  "name": "John Doe",
  "job": "QA Engineer"
}
```

Example Response


```
{
  "name": "John Doe",
  "job": "QA Engineer",
  "id": "101",
  "createdAt": "2024-02-13T12:34:56Z"
}
```

📌 2.3 PUT Request (Updating Data)

- Used to update an existing resource
- Requires both **ID** and **request body**

Example API Endpoint (Updating User)

```
PUT https://reqres.in/api/users/101
```

Request Body

```
{
  "name": "John Doe",
  "job": "Senior QA Engineer"
}
```

Example Response

```
{
  "name": "John Doe",
  "job": "Senior QA Engineer",
  "updatedAt": "2024-02-13T12:35:20Z"
}
```

📌 2.4 DELETE Request (Removing Data)

- Used to delete a resource
- No request body required

Example API Endpoint (Deleting User)

```
DELETE https://reqres.in/api/users/101
```

Example Response

```
Status Code: 204 No Content
```

---

3 Working with REST APIs using Postman

📌 Steps to Send Requests in Postman:

1 Open **Postman**
2 Select **GET, POST, PUT, or DELETE** from the dropdown
3 Enter the API URL
4 For **POST & PUT**, go to **Body** → Select **raw** → Enter JSON data
5 Click **Send**

📌 Verifying Response in Postman:

✓ Check Status Code (200, 201, 204, etc.)
✓ Validate Response Body
✓ Verify Headers

---

4 Introduction to REST Assured for API Testing

📌 What is REST Assured?

- A Java library for API testing
- Supports **BDD-style** testing

📌 Setting Up REST Assured:

✓ Add the following dependency in **pom.xml** (Maven)

```
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured</artifactId>
    <version>5.3.0</version>
    <scope>test</scope>
</dependency>
```

📌 4.1 Writing Basic API Tests with REST Assured

GET Request in REST Assured

```
import io.restassured.RestAssured;
import io.restassured.response.Response;
import static org.hamcrest.Matchers.equalTo;

public class GetRequestExample {
    public static void main(String[] args) {
        RestAssured.baseURI = "https://reqres.in/api";

        Response response = RestAssured.given()
            .when().get("/users/2")
            .then().statusCode(200)
            .body("data.id", equalTo(2))
            .extract().response();

        System.out.println("Response: " + response.asPrettyString());
    }
}
```

POST Request in REST Assured

```java
import io.restassured.RestAssured;
import io.restassured.http.ContentType;
import static io.restassured.RestAssured.given;

public class PostRequestExample {
    public static void main(String[] args) {
        RestAssured.baseURI = "https://reqres.in/api";

        given()
            .contentType(ContentType.JSON)
            .body("{ \"name\": \"John\", \"job\": \"QA Engineer\" }")
        .when()
            .post("/users")
        .then()
            .statusCode(201);
    }
}
```

---

5 Day 1 Exercises

1 Send GET, POST, PUT, DELETE requests using Postman
2 Write a REST Assured test for a GET request
3 Create a POST request using REST Assured
4 Implement assertions for response validation
5 Validate response status codes and headers

---

6 Best Practices

✓ **Use meaningful assertions in API tests**
✓ **Validate status codes & response times**
✓ **Check both positive & negative scenarios**
✓ **Avoid hardcoded values** (use dynamic test data)

---

7 Possible Interview Questions & Evaluation Topics

📌 **Conceptual Questions**

- What is REST API?
- Difference between GET and POST?
- What are Status Codes in API responses?
- What is the difference between PUT and PATCH?
- Why do we use REST Assured?

📌 **Coding Questions**

- 🔹 Write a REST Assured test for a DELETE request.
- 🔹 Validate JSON response using REST Assured.
- 🔹 Handle authentication in API testing.

📌 **Summary of API Automation Day 1**

✓ Understanding REST APIs & HTTP Methods
✓ Working with APIs using Postman
✓ Introduction to REST Assured for API Testing
✓ Best Practices & Interview Questions

🧮 API Automation – Day 2

Topic: RestAssured Basics

📌 Day 2 Agenda:

1 Introduction to REST Assured

- What is REST Assured?
- Setting Up REST Assured in a Java Project

2 Writing Basic API Tests using REST Assured

- Performing **GET, POST, PUT, DELETE** requests

3 Understanding Response Handling

- Validating Status Codes
- Validating Response Body
- Validating Headers

4 Extracting Data from API Responses

5 Implementing Assertions in REST Assured

6 Day 2 Exercises

7 Best Practices

8 Possible Interview Questions

---

1 Introduction to REST Assured

📌 **What is REST Assured?**

- A Java-based library for API testing
- Supports **BDD-style syntax** for writing tests
- Works with **JUnit and TestNG**

📌 **Why Use REST Assured?**

✓ Automates API testing easily
✓ Simple & readable syntax
✓ Supports JSON & XML validation
✓ Integrates with Java testing frameworks

---

2 Setting Up REST Assured

📌 **Maven Dependency (Add to pom.xml)**

```
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured</artifactId>
    <version>5.3.0</version>
    <scope>test</scope>
</dependency>
```

📌 **Import Required Classes**

```
import io.restassured.RestAssured;
import io.restassured.response.Response;
import static io.restassured.RestAssured.*;
import static org.hamcrest.Matchers.*;
```

---

3 Writing Basic API Tests using REST Assured

📌 **3.1 GET Request (Fetching Data)**

```java
import io.restassured.RestAssured;
import io.restassured.response.Response;
import static io.restassured.RestAssured.*;
import static org.hamcrest.Matchers.*;

public class GetRequestExample {
    public static void main(String[] args) {
        RestAssured.baseURI = "https://reqres.in/api";

        given()
            .when()
                .get("/users/2")
            .then()
                .statusCode(200)
                .body("data.id", equalTo(2))
                .body("data.email", containsString("@reqres.in"))
                .log().all();
    }
}
```

Explanation:

✓ Uses **GET** request to fetch user details

✓ Validates **status code 200**

✓ Checks **JSON response body**

📌 3.2 POST Request (Creating Data)

```java
import io.restassured.http.ContentType;

public class PostRequestExample {
    public static void main(String[] args) {
        RestAssured.baseURI = "https://reqres.in/api";

        given()
            .contentType(ContentType.JSON)
            .body("{ \"name\": \"John\", \"job\": \"QA Engineer\" }")
        .when()
            .post("/users")
        .then()
            .statusCode(201)
            .body("name", equalTo("John"))
            .body("job", equalTo("QA Engineer"))
            .log().all();
    }
}
```

Explanation:

✓ Uses **POST** request to create a user

✓ Sends **JSON payload**

✓ Validates **status code 201**

✓ Asserts that response contains **correct name & job**

📌 3.3 PUT Request (Updating Data)

```java
public class PutRequestExample {
    public static void main(String[] args) {
        RestAssured.baseURI = "https://reqres.in/api";

        given()
            .contentType(ContentType.JSON)
            .body("{ \"name\": \"John\", \"job\": \"Senior QA Engineer\" }")
        .when()
            .put("/users/2")
        .then()
            .statusCode(200)
            .body("job", equalTo("Senior QA Engineer"))
            .log().all();
    }
}
```

Explanation:
✓ Uses **PUT** request to update user data
✓ Sends **updated JSON payload**

📌 3.4 DELETE Request (Removing Data)

```java
public class DeleteRequestExample {
    public static void main(String[] args) {
        RestAssured.baseURI = "https://reqres.in/api";

        given()
            .when()
                .delete("/users/2")
            .then()
                .statusCode(204)  // No Content
                .log().all();
    }
}
```

Explanation:
✓ Uses **DELETE** request to remove a user
✓ Asserts that **status code 204 (No Content)** is returned

---

4 Extracting Data from API Responses

📌 Extracting JSON Data from Response

```
Response response = given().when().get("/users/2").then().extract().response();

String userEmail = response.jsonPath().getString("data.email");
System.out.println("User Email: " + userEmail);
```

---

5 Implementing Assertions in REST Assured

📌 **Common Assertions**

✓ **Status Code Validation**

```
.then().statusCode(200);
```

✓ **Response Body Validation**

```
.then().body("data.id", equalTo(2));
```

✓ **Header Validation**

```
.then().header("Content-Type", "application/json; charset=utf-8");
```

---

6 Day 2 Exercises

📌 **Beginner:**

✅ Write a GET request to fetch user data.

✅ Write a POST request to create a new user.

📌 **Intermediate:**

✅ Validate response headers & body for a GET request.

✅ Write a PUT request to update user details.

📌 **Advanced:**

✅ Extract and store API response data in a variable.

✅ Write a DELETE request and validate response status.

---

7 Best Practices

✓ **Use Base URI & Path Parameters**

```
RestAssured.baseURI = "https://reqres.in/api";
```

✓ **Always Validate Status Codes & Responses**

✓ **Use Logs for Debugging**

```
.log().all();
```

✓ **Keep Test Data Dynamic (Avoid Hardcoding)**
✓ **Write Negative Test Cases (Invalid Data, 404 Errors, etc.)**

---

8 Possible Interview Questions & Evaluation Topics

📌 **Conceptual Questions**

- What is REST Assured?
- How do you validate API responses in REST Assured?
- What is the difference between PUT and PATCH?
- How do you handle authentication in API testing?

📌 **Coding Questions**

- Write a GET request test case using REST Assured.
- Implement assertions for status codes and headers.
- Extract and validate specific fields from API response.

📌 Summary of API Automation Day 2

✓ **Introduction to REST Assured**
✓ **Performing API Requests (GET, POST, PUT, DELETE)**
✓ **Extracting and Validating API Responses**
✓ **Best Practices & Interview Questions**

API Automation – Day 3

Topic: Authentication (OAuth, JWT, API Keys)

📌 Day 3 Agenda:

1 Introduction to Authentication in APIs

- Why is Authentication Needed?
- Common Authentication Mechanisms

2 Understanding API Key Authentication

- How API Keys Work
- Implementing API Key Authentication in RestAssured

3 Understanding OAuth 2.0 Authentication

- OAuth Flow (Client Credentials, Authorization Code)
- Implementing OAuth 2.0 in RestAssured

4 Understanding JWT Authentication

- How JWT Works
- Implementing JWT Authentication in RestAssured

5 Day 3 Exercises

6 Best Practices

7 Possible Interview Questions

---

1 Introduction to Authentication in APIs

📌 **Why is Authentication Needed?**

- Ensures **only authorized users** can access APIs
- Prevents **unauthorized access and data breaches**
- Helps in enforcing **role-based access control (RBAC)**

📌 **Common Authentication Mechanisms in APIs**

🔹 **API Key Authentication** – A secret key is passed in the request headers.
🔹 **OAuth 2.0** – Secure, token-based authentication for web & mobile apps.
🔹 **JWT (JSON Web Token)** – A stateless authentication mechanism with signed tokens.

---

2 API Key Authentication

📌 **How API Keys Work?**

- A unique key is issued per user/application.
- The key is sent in the **headers** or **query parameters** of API requests.
- Example: https://api.example.com/data?api_key=YOUR_SECRET_KEY

📌 **Example: API Key Authentication with RestAssured**

```java
import io.restassured.RestAssured;
import io.restassured.response.Response;
import static io.restassured.RestAssured.*;

public class APIKeyAuthExample {
    public static void main(String[] args) {
        RestAssured.baseURI = "https://api.example.com";

        Response response = given()
            .header("x-api-key", "YOUR_API_KEY")
            .when()
            .get("/data")
            .then()
            .statusCode(200)
            .log().all()
            .extract().response();
    }
}
```

✓ **Pass API Key in the request headers**
✓ **Validate response & status codes**

---

3 OAuth 2.0 Authentication

📌 **What is OAuth 2.0?**

- A secure authentication protocol using **access tokens**.
- Used by major platforms like **Google, Facebook, GitHub**.
- Common grant types:
  - **Client Credentials Flow** – Machine-to-machine authentication
  - **Authorization Code Flow** – User-based authentication

📌 **OAuth 2.0 Client Credentials Flow (RestAssured Example)**

```java
import io.restassured.RestAssured;
import io.restassured.response.Response;
import static io.restassured.RestAssured.*;

public class OAuth2Example {
    public static void main(String[] args) {
        RestAssured.baseURI = "https://auth.example.com";

        // Step 1: Get OAuth Token
        String accessToken = given()
            .contentType("application/x-www-form-urlencoded")
            .formParam("grant_type", "client_credentials")
            .formParam("client_id", "YOUR_CLIENT_ID")
            .formParam("client_secret", "YOUR_CLIENT_SECRET")
            .when()
            .post("/oauth/token")
            .then()
            .statusCode(200)
            .extract().path("access_token");

        // Step 2: Use OAuth Token in API Request
        given()
            .auth().oauth2(accessToken)
            .when()
            .get("/api/protected-resource")
            .then()
            .statusCode(200)
            .log().all();
    }
}
```

✓ Step 1: Get OAuth Token
✓ Step 2: Use Token to Access Protected APIs

---

4 JWT (JSON Web Token) Authentication

📌 What is JWT?

- A **self-contained token** that contains authentication details.
- Composed of **Header, Payload, Signature**.
- Used for **stateless authentication** in APIs.

📌 How JWT Works?

1 User logs in & gets a JWT token.

2 JWT is sent in API requests as **Authorization: Bearer <token>**.
3 Server verifies & grants access.

📌 **Example: JWT Authentication in RestAssured**

```java
import io.restassured.RestAssured;
import static io.restassured.RestAssured.*;

public class JWTAuthExample {
    public static void main(String[] args) {
        RestAssured.baseURI = "https://api.example.com";

        // Example JWT Token
        String jwtToken = "YOUR_JWT_TOKEN";

        given()
            .header("Authorization", "Bearer " + jwtToken)
            .when()
            .get("/protected-endpoint")
            .then()
            .statusCode(200)
            .log().all();
    }
}
```

✓ **JWT is sent in Authorization: Bearer <token> header**
✓ **Token is validated by the server before granting access**

---

5 Day 3 Exercises

📌 **Beginner:**
✅ Use API Key authentication to access an API.
✅ Validate response status codes.

📌 **Intermediate:**
✅ Implement OAuth 2.0 authentication & fetch protected data.
✅ Validate response headers in OAuth API requests.

📌 **Advanced:**
✅ Use JWT authentication to access a protected API endpoint.
✅ Extract & decode JWT tokens in Java.

---

6 Best Practices

✓ **Never expose API Keys or OAuth Credentials in code**
✓ **Use Environment Variables for storing sensitive keys**
✓ **Always validate authentication responses properly**
✓ **Use Token Expiry & Refresh Tokens for OAuth authentication**
✓ **Implement Role-Based Access Control (RBAC) for different users**

---

7 Possible Interview Questions & Evaluation Topics

📌 **Conceptual Questions**

- What are the differences between API Key, OAuth 2.0, and JWT authentication?
- When should you use API Key authentication over OAuth 2.0?
- How does OAuth 2.0 Client Credentials flow work?
- What is a refresh token in OAuth 2.0?
- How does JWT authentication work, and what are its security risks?

📌 **Coding Questions**

- Write a RestAssured test using API Key authentication.
- Implement OAuth 2.0 authentication in RestAssured.
- Validate & extract JWT tokens from API responses.

📌 Summary of API Automation Day 3

✓ **API Authentication Types (API Key, OAuth 2.0, JWT)**
✓ **Implementing API Key Authentication in RestAssured**
✓ **Using OAuth 2.0 Client Credentials Flow**
✓ **Authenticating APIs with JWT Tokens**
✓ **Best Practices & Interview Questions**

🗓️ API Automation – Day 4

Topic: Assertions, Logging & Reporting + API & UI Test Integration

---

📌 Day 4 Agenda:

1 Understanding Assertions in API Testing

- Why Assertions Are Important
- Using RestAssured for Assertions
- Validating Response Status Codes, Headers, and Body

2 Logging and Debugging in RestAssured

- Using Logs to Debug API Requests & Responses

3 Reporting API Test Results

- Generating Reports using TestNG and ExtentReports

4 Integrating API and UI Tests

- When and Why to Integrate API & UI Tests
- How to Use Selenium and RestAssured Together

5 Day 4 Exercises

6 Best Practices

7 Possible Interview Questions

---

1 Understanding Assertions in API Testing

📌 **Why Are Assertions Important?**

- Ensure API responses meet **expected behavior**.
- Validate **response status codes, headers, and body content**.
- Help detect **unexpected API changes and failures**.

📌 **Types of Assertions in API Testing:**

✅ **Status Code Validation** – Ensures the API response has the expected status.
✅ **Response Time Assertion** – Ensures the API responds within the expected time.
✅ **Header Validation** – Verifies expected headers are present in the response.
✅ **Response Body Assertion** – Validates JSON/XML response data.

📌 **Example: Assertions in RestAssured**

```java
import io.restassured.RestAssured;
import static io.restassured.RestAssured.*;
import static org.hamcrest.Matchers.*;

public class APIAssertions {
    public static void main(String[] args) {
        RestAssured.baseURI = "https://api.example.com";

        given()
            .when()
            .get("/users/1")
            .then()
            .statusCode(200)  // ✅ Assert status code
            .header("Content-Type", equalTo("application/json"))  // ✅ Assert header
            .body("id", equalTo(1))  // ✅ Assert JSON response value
            .body("email", containsString("@example.com"))  // ✅ Assert email format
            .log().all(); // Log the request & response
    }
}
```

✓ Checks API Response Status Code
✓ Validates Content-Type in Headers
✓ Asserts JSON Body Values

---

2 Logging and Debugging in RestAssured

📌 **Why Logging is Important?**

- Helps debug API request and response failures.
- Provides insights into **request payloads, headers, and responses**.

📌 **Logging Options in RestAssured**

✅ .log().all() – Logs **everything (request & response)**
✅ .log().headers() – Logs only **headers**
✅ .log().body() – Logs **response body**
✅ .log().ifError() – Logs only when **errors occur**

📌 **Example: Logging API Requests & Responses**

```java
given()
    .log().all() // ✅ Logs Request
    .when()
    .get("/users/1")
    .then()
    .log().ifError(); // ✅ Logs Response only if there's an error
```

✓ Useful for Debugging API Failures
✓ Log selectively using .ifError()

3 Reporting API Test Results

📌 **Why Reporting is Essential?**

- Helps **track test execution results**.
- Generates reports for **stakeholders and debugging**.
- Supports **integration with CI/CD pipelines**.

📌 **Generating Reports with TestNG & ExtentReports**

✅ **TestNG** – Generates basic HTML test reports.

✅ **ExtentReports** – Creates detailed test execution reports with logs & screenshots.

📌 **Example: Integrating ExtentReports with API Tests**

```java
import com.aventstack.extentreports.*;
import com.aventstack.extentreports.reporter.ExtentHtmlReporter;
import io.restassured.RestAssured;
import static io.restassured.RestAssured.*;
import org.testng.annotations.*;

public class APIReportExample {
    ExtentReports extent;
    ExtentTest test;

    @BeforeClass
    public void setupReport() {
        ExtentHtmlReporter htmlReporter = new ExtentHtmlReporter("API_TestReport.html");
        extent = new ExtentReports();
        extent.attachReporter(htmlReporter);
    }

    @Test
    public void validateUserAPI() {
        test = extent.createTest("Validate User API");

        given()
            .when()
            .get("https://api.example.com/users/1")
            .then()
            .statusCode(200);

        test.pass("API Test Passed Successfully");
    }

    @AfterClass
    public void tearDown() {
        extent.flush(); // ✅ Generates report
    }
}

```

✓ Creates an HTML Report

✓ Logs API Test Execution Details

---

4 Integrating API & UI Tests

📌 **Why Integrate API and UI Tests?**

- Reduces **dependency on UI testing** alone.
- API Tests are **faster** and help find **backend issues early**.
- Ensures end-to-end **data flow validation** between UI & API.

📌 **Example: API & UI Test Integration (Selenium + RestAssured)**

```java
import io.restassured.RestAssured;
import io.restassured.response.Response;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import static io.restassured.RestAssured.*;

public class APIUIIntegrationTest {
    public static void main(String[] args) {
        // ✅ Step 1: Get user data from API
        RestAssured.baseURI = "https://api.example.com";
        Response response = given().when().get("/users/1");

        String expectedUsername = response.jsonPath().getString("username");

        // ✅ Step 2: Validate user details on UI
        System.setProperty("webdriver.chrome.driver", "path-to-chromedriver");
        WebDriver driver = new ChromeDriver();
        driver.get("https://example.com/login");

        WebElement usernameField = driver.findElement(By.id("username"));
        usernameField.sendKeys(expectedUsername);

        WebElement loginButton = driver.findElement(By.id("login"));
        loginButton.click();

        driver.quit();
    }
}
```

✓ Retrieves Data from API
✓ Uses API Data for UI Testing

---

5 Day 4 Exercises

1 **Write a RestAssured test to validate:**

- Response status

- Headers
- JSON body elements

2 **Log requests & responses for API calls**

3 **Generate ExtentReports for API tests**

4 **Integrate an API test with Selenium UI validation**

---

6 Best Practices

✅ Use assertions for **status codes, headers, and response data**
✅ Implement **logging** to capture API request-response details
✅ Maintain **separate test suites for API & UI tests**
✅ Use reporting tools like **ExtentReports, TestNG Reports**
✅ Follow **CI/CD integration for automated API tests**

---

7 Possible Interview Questions

1 What is the purpose of assertions in API testing?
2 How can you log API requests and responses using RestAssured?
3 What are the advantages of using ExtentReports for API test reporting?
4 How do you integrate API testing with Selenium UI tests?
5 What challenges can arise while integrating API and UI tests?
