# REST Assured GET Request

In REST Assured, a **GET request** is used to retrieve data from a given endpoint. Here's a **step-by-step guide** to performing a GET request in REST Assured along with code and explanation.

---

## ✅ Basic Structure of a GET Request in REST Assured

### 💡 Dependency (Maven)

Make sure you have this in your `pom.xml`:

```
xml

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

---

## ✅ Simple GET Request Example

```
java

import io.restassured.RestAssured;
import io.restassured.response.Response;

public class GetExample {
    public static void main(String[] args) {
        // Base URI
        RestAssured.baseURI = "https://jsonplaceholder.typicode.com";

        // Sending GET request to /posts/1
        Response response = RestAssured
                                .given()
                                .when()
                                .get("/posts/1");

        // Print response body
        System.out.println("Response Body:");
        System.out.println(response.getBody().asString());

        // Status Code
        System.out.println("Status Code: " + response.getStatusCode());

        // Header
        System.out.println("Content-Type: " + response.getHeader("Content-Type"));
    }
}
```

---

## ✅ Explanation

| Part | Description |
|---|---|
| **`RestAssured.baseURI`** | Sets the base URL for all requests |
| **`.given()`** | Prepares the request specification |
| **`.when()`** | Triggers the actual HTTP request |
| **`.get("/endpoint")`** | Sends the GET request to the given endpoint |
| **`.getBody().asString()`** | Converts the response body to a string |
| **`.getStatusCode()`** | Gets the HTTP status code (e.g., 200) |
| **`.getHeader()`** | Fetches specific headers from the response |

## ✅ Validate GET Response with Assertions

You can use **Hamcrest Matchers** for validation:

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

public class GetWithAssertions {
    public static void main(String[] args) {
        baseURI = "https://jsonplaceholder.typicode.com";

        given().
        when().
            get("/posts/1").
        then().
            assertThat().
            statusCode(200).
            body("userId", equalTo(1)).
            body("id", equalTo(1));
    }
}
```

## ✅ Common Methods in GET

| Method | Description |
|---|---|
| **`getStatusCode()`** | Status code of response |
| **`getBody().asString()`** | Response body |
| **`getTime()`** | Response time in ms |
| **`jsonPath().get("key")`** | Parse specific JSON fields |

Would you like me to provide:

- GET request with query parameters?
- GET request with headers?
- Integration with TestNG or JUnit?

Let me know what you're building!

Great! Let's go through how to **validate the status code**, **headers**, and **response body** in a **REST Assured GET request** using assertions.

## ✅ Full Example with Validations

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

public class GetValidationExample {
    public static void main(String[] args) {

        // Base URI
        baseURI = "https://jsonplaceholder.typicode.com";

        // Validate status code, headers, and response body
        given().
        when().
            get("/posts/1").
        then().
            assertThat().
            statusCode(200). // ✅ Status Code Validation
            header("Content-Type", equalTo("application/json; charset=utf-8")). // ✅ Header Validation
            body("userId", equalTo(1)). // ✅ Response Body Field Validation
            body("id", equalTo(1)).
            body("title", not(empty())); // ✅ Field not empty check
    }
}
```

## ✅ Explanation of Validations

| Validation | Code | Meaning |
|---|---|---|
| **Status Code** | `statusCode(200)` | Checks if response returned HTTP 200 |
| **Header** | `header("Content-Type", equalTo("application/json; charset=utf-8"))` | Validates exact value of a response header |
| **Body Field Equals** | `body("userId", equalTo(1))` | Verifies if `userId` is 1 |
| **Field Not Empty** | `body("title", not(empty()))` | Ensures the `title` is not an empty string |

## ✅ Extract and Print for Debugging (Optional)

If you want to print and verify manually:

```java
Response response = get("/posts/1");

System.out.println("Status Code: " + response.getStatusCode());
```

```
  System.out.println("Header Content-Type: " + response.getHeader("Content-Type"));
  System.out.println("Body: " + response.getBody().asPrettyString());
```

---

Would you like to validate:

- A **nested JSON** field?
- A **list/array** in the response?
- Use **TestNG/JUnit** assertions instead?

Let me know how you'd like to proceed!