What is Matchers in Rest Assured?

In Rest Assured, the Matchers class is used to perform assertions on the response received from an API call. It provides a set of predefined matchers that can be used to validate various aspects of the response, such as the content, headers, status codes, etc. These matchers are typically used in combination with the then() method to verify that the API response meets the expected criteria.

Rest Assured leverages the Matchers from the Hamcrest library, which is widely used for writing flexible and readable assertions in test code.

Methods in the Matchers Class in Rest Assured

The Matchers class in Rest Assured provides a variety of methods for performing assertions. Below are the commonly used methods, along with their return types:

1. **equalTo(Object operand)**
    - **Return Type:** Matcher<T>
    - **Description:** Checks if the actual value is equal to the given operand.
2. **hasSize(int size)**
    - **Return Type:** Matcher<Collection<T>>
    - **Description:** Verifies if the collection has the specified size.
3. **containsString(String substring)**
    - **Return Type:** Matcher<String>
    - **Description:** Checks if the string contains the specified substring.
4. **startsWith(String prefix)**
    - **Return Type:** Matcher<String>
    - **Description:** Checks if the string starts with the given prefix.
5. **endsWith(String suffix)**
    - **Return Type:** Matcher<String>
    - **Description:** Checks if the string ends with the specified suffix.
6. **hasItem(T item)**
    - **Return Type:** Matcher<Iterable<? super T>>
    - **Description:** Verifies if the collection contains the specified item.

7. **greaterThan(T value)**
    - **Return Type:** Matcher<T>
    - **Description:** Checks if the actual value is greater than the given value.
8. **lessThan(T value)**
    - **Return Type:** Matcher<T>
    - **Description:** Checks if the actual value is less than the given value.
9. **not(Matcher<T> matcher)**
    - **Return Type:** Matcher<T>
    - **Description:** Asserts that the actual value does not match the specified matcher.
10. **hasEntry(K key, V value)**
    - **Return Type:** Matcher<Map<? extends K,? extends V>>
    - **Description:** Verifies if the map contains the specified key-value pair.
11. **allOf(Matcher<? super T>... matchers)**
    - **Return Type:** Matcher<T>
    - **Description:** Asserts that all the specified matchers are true.
12. **anyOf(Matcher<? super T>... matchers)**
    - **Return Type:** Matcher<T>
    - **Description:** Asserts that at least one of the specified matchers is true.

Let's explore 5 simple examples using the Matchers class in the BDD pattern:

1. Equal To Example

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


public class ApiTest {
   public void validateResponse() {
      given()
      .when()
```

```
        .get("https://api.example.com/users/1")
    .then()
        .assertThat()
        .body("name", equalTo("Sweta"));
  }
}
```

Explanation: This example checks if the name field in the response body is equal to "Sweta".

2. Contains String Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.containsString;

public class ApiTest {
  public void validateResponse() {
    given()
    .when()
        .get("https://api.example.com/users/1")
    .then()
        .assertThat()
        .body("email", containsString("mike"));
  }
}
```

Explanation: This example checks if the email field in the response body contains the substring "mike".

3. Starts With Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.startsWith;
```

```
public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/users/1")
        .then()
            .assertThat()
            .body("username", startsWith("Mahi"));
    }
}
```

Explanation: This example checks if the username field in the response body starts with "Mahi".

4. Has Size Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.hasSize;

public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/users")
        .then()
            .assertThat()
            .body("users", hasSize(10));
    }
}
```

Explanation: This example checks if the users array in the response body has a size of 10.

5. Greater Than Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.greaterThan;

public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/orders/total")
        .then()
            .assertThat()
            .body("totalAmount", greaterThan(100));
    }
}
```

Explanation: This example checks if the totalAmount field in the response body is greater than 100.

Let's explore 5 moderate examples using the Matchers class in the BDD pattern:

6. Ends With Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.endsWith;

public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/users/1")
        .then()
            .assertThat()
```

```
        .body("email", endsWith("@example.com"));
    }
}
```

Explanation: This example checks if the email field in the response body ends with "@example.com".

7. Has Item Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.hasItem;

public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/users")
        .then()
            .assertThat()
            .body("users.names", hasItem("Sweta"));
    }
}
```

Explanation: This example checks if the names array in the response body contains the name "Sweta".

8. Less Than Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.lessThan;

public class ApiTest {
    public void validateResponse() {
        given()
```

```
        .when()
            .get("https://api.example.com/orders/total")
        .then()
            .assertThat()
            .body("totalAmount", lessThan(500));
    }
}
```

Explanation: This example checks if the totalAmount field in the response body is less than 500.

9. Has Entry Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.hasEntry;

public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/orders/1")
        .then()
            .assertThat()
            .body("orderDetails", hasEntry("status", "shipped"));
    }
}
```

Explanation: This example checks if the orderDetails map in the response body contains a key-value pair with "status": "shipped".

10. Not Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.not;
```

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


public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/users/1")
        .then()
            .assertThat()
            .body("username", not(equalTo("admin")));
    }
}
```

Explanation: This example checks if the username field in the response body is not equal to "admin".

Let's explore 5 hard examples using the Matchers class in the BDD pattern:

11. All Of Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.allOf;
import static org.hamcrest.Matchers.startsWith;
import static org.hamcrest.Matchers.containsString;


public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/users/1")
        .then()
```

```
        .assertThat()
        .body("username", allOf(startsWith("M"), containsString("ike")));
    }
}
```

Explanation: This example checks if the username field in the response body starts with "M" and contains the substring "ike".

12. Any Of Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.anyOf;
import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.startsWith;

public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/users/1")
        .then()
            .assertThat()
            .body("role", anyOf(equalTo("admin"), startsWith("user")));
    }
}
```

Explanation: This example checks if the role field in the response body is either "admin" or starts with "user".

13. Has Size with All Of Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.allOf;
import static org.hamcrest.Matchers.hasSize;
```

```
import static org.hamcrest.Matchers.greaterThan;


public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/users")
        .then()
            .assertThat()
            .body("users", allOf(hasSize(10), hasSize(greaterThan(5))));
    }
}
```

Explanation: This example checks if the users array in the response body has a size of 10 and that the size is greater than 5.

14. Complex Has Entry Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.hasEntry;
import static org.hamcrest.Matchers.hasItem;


public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/orders/1")
        .then()
            .assertThat()
            .body("orderDetails", hasEntry("items", hasItem("laptop")));
```

```
    }
}
```

Explanation: This example checks if the orderDetails map in the response body contains an entry with "items": ["laptop"].

15. Combined Assertions Example

```
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.allOf;
import static org.hamcrest.Matchers.hasSize;
import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.hasItem;

public class ApiTest {
    public void validateResponse() {
        given()
        .when()
            .get("https://api.example.com/users")
        .then()
            .assertThat()
            .body("users", allOf(
                hasSize(10),
                hasItem(equalTo("Mahi")),
                hasItem(equalTo("Sweta"))
            ));
    }
}
```

Explanation: This example checks if the users array in the response body has a size of 10 and contains the names "Mahi" and "Sweta".

These examples demonstrate the flexibility and power of the Matchers class in Rest Assured for performing various types of assertions on API responses. By combining different matchers, you can create complex validation logic that ensures your API behaves as expected.