

Log Request

```
given().log().all(). .. // Log all
given().log().parameters(). .. // Log only the
parameters of the request
given().log().body(). .. // Log only the
request body
given().log().headers(). .. // Log only the
request headers
given().log().cookies(). .. // Log only the
request cookies
given().log().method(). .. // Log only the
request method
given().log().path(). .. // Log only the
request path
// Log if test fail
given().log().ifValidationFails(). ..
```

Log Response

```
get("/x").then().log().all(). .. // Log all
get("/x ").then().log().body(). .. // Log
body
get("/x ").then().log().ifError(). .. // Log
body if error
get("/x ").then().log().statusLine(). .. //
Only log the status line
get("/x ").then().log().headers(). .. //
Only log the response headers
get("/x ").then().log().cookies(). .. //
Only log the response cookies
// Only log if the status code is equal to 302
get("/x ").then().log().ifStatusCodeIsEqualTo(302). ..
// Only log if the status code matches the supplied
Hamcrest matcher
get("/x ").then().log().ifStatusCodeMatches(matcher). ..
// Log if test fail
.then().log().ifValidationFails(). ..
```

Config Log

```
// Log both request and response when test fail
RestAssured.enableLoggingOfRequestAndResponseIfValidationFails();
// Log both request and response.
RestAssured.filters(new RequestLoggingFilter(), new ResponseLoggingFilter());
// Log when using allure -report
given().filter(new AllureRestAssured())
```

Others

```
//compare schema
then().body(matchesJsonSchemaInClassPath("bookstore-schema.json"))
//Compare content of 2 files
File file1 = new File("src/test/resources/ -
books.json");
File file2 = new File("src/test/resources/ -
books-simple.json");
assertThat(contentOf(file1).isEqualTo(contentOf(file2));
//json_to_pojo
var mouseAction = then().extract()
.jsonPath().get(
Object("", MouseAction.class);
//json_to_list_of_pojo
var listMouseAction = then().extract()
.jsonPath().get(
List("", MouseAction.class);
MouseAction[] listMouseAction2 = jsonPath(
th().getOrDefault("", MouseAction[].class);
```

Jackson Annotation

```
//json_to_object_ignore_keys
@JsonIgnore(ignoreKeys = {
    "name", "id"
})

public class Foo {}

//One item as array
@JsonFormat(with = JsonFormat.Value.ACCEPT_SINGLE_VALUE_AS_ARRAY)
public List<Result> result;

//Set order of json key
@JsonPropertyOrder({ "name", "id" })

//Set property name for a field
@JsonProperty("name")
public String f_name

public class Contact {
    @JsonFormat(pattern = "yyyy-MM-dd")
    private LocalDate birthday;

    @JsonFormat(pattern = "yyyy-MM-dd HH:mm:ss")
    private LocalDateTime lastUpdate;
}

<dependency>
    <groupId>com.fasterxml.jackson.dataformat</groupId>
    <artifactId>jackson-dataformat-xml</artifactId>
    <version>2.9.7</version>
</dependency>
```

