

GettingStarted

Johan Haleby edited this page on 23 Nov · 37 revisions

Contents


► Pages 28

1. [Maven / Gradle](#)
2. [REST Assured](#)
3. [JsonPath](#)
4. [XmlPath](#)
5. [JSON Schema Validation](#)
6. [Spring Mock MVC](#)
7. [Scala Module](#)
8. [Non-maven users](#)
9. [Static Imports](#)
10. [Version 2.x](#)
11. [Documentation](#)

- [Getting Started](#)
- [Downloads](#)
- [Usage Guide \(Legacy\)](#)
- [Snapshot dependencies](#)
- [Release Notes](#)
- [FAQ](#)
- [Support](#)

Clone this wiki locally

<https://github.com/rest-ass> 

 Clone in Desktop

Maven / Gradle Users

Add the following dependency to your pom.xml:

REST Assured

Includes [JsonPath](#) and [XmlPath](#)

Maven:

```
<dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>rest-assured</artifactId>
  <version>3.0.6</version>
  <scope>test</scope>
</dependency>
```

Gradle:

```
testCompile 'io.rest-assured:rest-assured:3.0.6'
```

Notes

1. You should place rest-assured before the JUnit dependency declaration in your pom.xml / build.gradle in order to make sure that the correct version of Hamcrest is used.
2. REST Assured includes JsonPath and XmlPath as transitive dependencies

JsonPath

Standalone JsonPath (included if you depend on the `rest-assured` artifact). Makes it easy to parse JSON documents. Note that this JsonPath implementation uses [Groovy's GPath](#) syntax and is not to be confused with Kalle Stenflo's [JsonPath](#) implementation.

Maven:

```
<dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>json-path</artifactId>
```

```
<version>3.0.6</version>
</dependency>
```

Gradle:

```
compile 'io.rest-assured:json-path:3.0.6'
```

XmlPath

Stand-alone XmlPath (included if you depend on the `rest-assured` artifact). Makes it easy to parse XML documents.

Maven:

```
<dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>xml-path</artifactId>
  <version>3.0.6</version>
</dependency>
```

Gradle:

```
compile 'io.rest-assured:xml-path:3.0.6'
```

JSON Schema Validation

If you want to validate that a JSON response conforms to a [Json Schema](#) you can use the `json-schema-validator` module:

Maven:

```
<dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>json-schema-validator</artifactId>
  <version>3.0.6</version>
  <scope>test</scope>
</dependency>
```

Gradle:

```
testCompile 'io.rest-assured:json-schema-validator:3.0.6'
```

Refer to the [documentation](#) for more info.

Spring Mock Mvc

If you're using Spring Mvc you can now unit test your controllers using the [RestAssuredMockMvc](#) API in the `spring-mock-mvc` module. For this to work you need to depend on the `spring-mock-mvc` module:

Maven:

```
<dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>spring-mock-mvc</artifactId>
  <version>3.0.6</version>
  <scope>test</scope>
</dependency>
```

Gradle:

```
testCompile 'io.rest-assured:spring-mock-mvc:3.0.6'
```

Scala Support

If you're using Scala you may leverage the [scala-support](#) module. For this to work you need to depend on the `scala-support` module:

SBT:

```
libraryDependencies += "io.rest-assured" % "scala-support" % "3.0.6"
```

Maven:

```
<dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>scala-support</artifactId>
  <version>3.0.6</version>
  <scope>test</scope>
</dependency>
```

Gradle:

```
testCompile 'io.rest-assured:scala-support:3.0.6'
```

Non-maven users

Download [REST Assured](#) and [Json Schema Validator](#) (optional). You can also download [XmlPath](#) and/or [JsonPath](#) separately if you don't need REST Assured. If you're using Spring Mvc then you can download the [spring-mock-mvc](#) module as well. If you're using Scala you may optionally download the [scala-support](#) module. Extract the distribution zip file and put the jar files in your class-path.

Static imports

In order to use REST assured effectively it's recommended to statically import methods from the following classes:

```
io.restassured.RestAssured.*
io.restassured.matcher.RestAssuredMatchers.*
org.hamcrest.Matchers.*
```

If you want to use [Json Schema](#) validation you should also statically import these methods:

```
io.restassured.module.json.JsonSchemaValidator.*
```

Refer to [Json Schema Validation](#) section for more info.

If you're using Spring MVC you can use the [spring-mock-mvc](#) module to unit test your Spring Controllers using the Rest Assured DSL. To do this statically import the methods from [RestAssuredMockMvc](#) *instead* of importing the methods from `io.rest-assured.RestAssured` and `io.rest-assured.matcher.RestAssuredMatchers`:

```
io.restassured.module.mockmvc.RestAssuredMockMvc.*
io.restassured.matcher.RestAssuredMatchers.*
```

Version 2.x

If you need to depend on an older version replace `groupId io.rest-assured` with `com.jayway.restassured`.

Documentation

When you've successfully downloaded and configured REST Assured in your classpath please refer to the [usage guide](#) for examples.