com.jayway.restassured.specification

**Interface ResponseSpecification**

* **All Superinterfaces:**

[RequestSender](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSender.html), [RequestSenderOptions](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html)<[Response](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/response/Response.html)>

**All Known Subinterfaces:**

[FilterableResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/FilterableResponseSpecification.html)

**All Known Implementing Classes:**

[ResponseSpecificationImpl](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/internal/ResponseSpecificationImpl.html)

public interface **ResponseSpecification** extends [RequestSender](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSender.html)

Allows you to specify how the expected response must look like in order for a test to pass.

* + ***Method Summary***

|  |  |
| --- | --- |
| **All Methods**[**Instance Methods**](javascript:show(2);)[**Abstract Methods**](javascript:show(4);) | |
| **Modifier and Type** | **Method and Description** |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**and**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#and--)()  Syntactic sugar, e.g. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**appendRoot**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#appendRoot-java.lang.String-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) pathToAppend)  Append the given path to the root path of the response body so that you don't need to write the entire path for each expectation. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**appendRoot**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#appendRoot-java.lang.String-java.util.List-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) pathToAppend, [**List**](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[**Argument**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments)  Append the given path to the root path with arguments supplied of the response body so that you don't need to write the entire path for each expectation. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**body**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.util.List-org.hamcrest.Matcher-java.lang.Object...-)([**List**](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[**Argument**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments, org.hamcrest.Matcher matcher, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)  This as special kind of expectation that is mainly useful when you've specified a root path with an argument placeholder. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**body**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-org.hamcrest.Matcher-org.hamcrest.Matcher...-)(org.hamcrest.Matcher<?> matcher, org.hamcrest.Matcher<?>... additionalMatchers)  Expect that the response body conforms to one or more Hamcrest matchers. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**body**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.lang.String-java.util.List-org.hamcrest.Matcher-java.lang.Object...-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) key, [**List**](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[**Argument**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments, org.hamcrest.Matcher matcher, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)  Same as [**body(String, org.hamcrest.Matcher, Object...)**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.lang.String-org.hamcrest.Matcher-java.lang.Object...-) expect that you can pass arguments to the key. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**body**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.lang.String-org.hamcrest.Matcher-java.lang.Object...-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) path, org.hamcrest.Matcher<?> matcher, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)  Expect that the JSON or XML response body conforms to one or more Hamcrest matchers. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**content**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#content-java.util.List-org.hamcrest.Matcher-java.lang.Object...-)([**List**](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[**Argument**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments, org.hamcrest.Matcher matcher, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)  This as special kind of expectation that is mainly useful when you've specified a root path with an argument placeholder. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**content**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#content-org.hamcrest.Matcher-org.hamcrest.Matcher...-)(org.hamcrest.Matcher<?> matcher, org.hamcrest.Matcher<?>... additionalMatchers)  Expect that the response content conforms to one or more Hamcrest matchers. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**content**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#content-java.lang.String-java.util.List-org.hamcrest.Matcher-java.lang.Object...-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) path, [**List**](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[**Argument**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments, org.hamcrest.Matcher matcher, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)  Same as [**body(String, java.util.List, org.hamcrest.Matcher, Object...)**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.lang.String-java.util.List-org.hamcrest.Matcher-java.lang.Object...-) expect that you can pass arguments to the path. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**content**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#content-java.lang.String-org.hamcrest.Matcher-java.lang.Object...-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) key, org.hamcrest.Matcher<?> matcher, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)  Expect that the JSON or XML response content conforms to one or more Hamcrest matchers. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**contentType**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#contentType-com.jayway.restassured.http.ContentType-)([**ContentType**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/http/ContentType.html) contentType)  Set the response content type to be contentType. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**contentType**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#contentType-org.hamcrest.Matcher-)(org.hamcrest.Matcher<? super [**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true)> contentType)  Expect the response content type to be contentType. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**contentType**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#contentType-java.lang.String-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) contentType)  Set the response content type to be contentType. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**cookie**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#cookie-java.lang.String-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) cookieName)  Expect that a cookie exist in the response, regardless of value (it may have no value at all). |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**cookie**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#cookie-java.lang.String-org.hamcrest.Matcher-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) cookieName, org.hamcrest.Matcher<?> expectedValueMatcher)  Expect that a response cookie matches the supplied cookie name and hamcrest matcher. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**cookie**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#cookie-java.lang.String-java.lang.Object-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) cookieName, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true) expectedValue)  Expect that a response cookie matches the supplied name and value. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**cookies**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#cookies-java.util.Map-)([**Map**](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/Map.html?is-external=true)<[**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true),?> expectedCookies)  Expect that response cookies matches those specified in a Map. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**cookies**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#cookies-java.lang.String-java.lang.Object-java.lang.Object...-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) firstExpectedCookieName, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true) firstExpectedCookieValue, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... expectedCookieNameValuePairs)  Expect that response cookies matches the supplied cookie names and values. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**defaultParser**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#defaultParser-com.jayway.restassured.parsing.Parser-)([**Parser**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/parsing/Parser.html) parser)  Register a default predefined parser that will be used if no other parser (registered or pre-defined) matches the response content-type. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**detachRoot**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#detachRoot-java.lang.String-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) pathToDetach)  Detach the given path from the root path. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**expect**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#expect--)()  Syntactic sugar, e.g. |
| [**RequestSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSpecification.html) | [**given**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#given--)()  Returns the request com.jayway.restassured.specification so that you can define the properties of the request. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**header**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#header-java.lang.String-org.hamcrest.Matcher-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) headerName, org.hamcrest.Matcher<?> expectedValueMatcher)  Expect that a response header matches the supplied header name and hamcrest matcher. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**header**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#header-java.lang.String-java.lang.String-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) headerName, [**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) expectedValue)  Expect that a response header matches the supplied name and value. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**headers**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#headers-java.util.Map-)([**Map**](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/Map.html?is-external=true)<[**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true),?> expectedHeaders)  Expect that response headers matches those specified in a Map. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**headers**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#headers-java.lang.String-java.lang.Object-java.lang.Object...-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) firstExpectedHeaderName, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true) firstExpectedHeaderValue, [**Object**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... expectedHeaders)  Expect that response headers matches the supplied headers and values. |
| [**ResponseLogSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseLogSpecification.html) | [**log**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#log--)()  Returns the [**ResponseLogSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseLogSpecification.html) that allows you to log different parts of the [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html). |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**noRoot**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#noRoot--)()  Reset the root path of the response body so that you don't need to write the entire path for each expectation. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**noRootPath**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#noRootPath--)()  Reset the root path of the response body so that you don't need to write the entire path for each expectation. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**parser**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#parser-java.lang.String-com.jayway.restassured.parsing.Parser-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) contentType, [**Parser**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/parsing/Parser.html) parser)  Register a content-type to be parsed using a predefined parser. |
| [**RequestSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSpecification.html) | [**request**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#request--)()  Returns the request com.jayway.restassured.specification so that you can define the properties of the request. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**response**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#response--)()  Syntactic sugar, e.g. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**root**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#root-java.lang.String-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) rootPath)  Set the root path of the response body so that you don't need to write the entire path for each expectation. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**root**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#root-java.lang.String-java.util.List-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) rootPath, [**List**](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[**Argument**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments)  Set the root path with arguments of the response body so that you don't need to write the entire path for each expectation. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**rootPath**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#rootPath-java.lang.String-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) rootPath)  Set the root path of the response body so that you don't need to write the entire path for each expectation. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**rootPath**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#rootPath-java.lang.String-java.util.List-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) rootPath, [**List**](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[**Argument**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments)  Set the root path with arguments of the response body so that you don't need to write the entire path for each expectation. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**spec**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#spec-com.jayway.restassured.specification.ResponseSpecification-)([**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) responseSpecificationToMerge)  Expect that the response matches an entire specification. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**specification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#specification-com.jayway.restassured.specification.ResponseSpecification-)([**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) responseSpecificationToMerge)  Expect that the response matches an entire specification. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**statusCode**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#statusCode-int-)(int expectedStatusCode)  Expect that the response status code matches an integer. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**statusCode**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#statusCode-org.hamcrest.Matcher-)(org.hamcrest.Matcher<? super [**Integer**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Integer.html?is-external=true)> expectedStatusCode)  Expect that the response status code matches the given Hamcrest matcher. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**statusLine**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#statusLine-org.hamcrest.Matcher-)(org.hamcrest.Matcher<? super [**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true)> expectedStatusLine)  Expect that the response status line matches the given Hamcrest matcher. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**statusLine**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#statusLine-java.lang.String-)([**String**](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) expectedStatusLine)  Expect that the response status line matches the given String. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**that**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#that--)()  Syntactic sugar, e.g. |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**then**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#then--)()  Syntactic sugar, e.g. |
| [**Response**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/response/Response.html) | [**validate**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#validate-com.jayway.restassured.response.Response-)([**Response**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/response/Response.html) response)  Validates the specified response against this ResponseSpecification |
| [**ResponseSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) | [**when**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#when--)()  Syntactic sugar, e.g. |
| [**RequestSpecification**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSpecification.html) | [**with**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#with--)()  Returns the request com.jayway.restassured.specification so that you can define the properties of the request. |

* + - **Methods inherited from interface com.jayway.restassured.specification.**[**RequestSenderOptions**](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html)

[delete](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#delete--), [delete](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#delete-java.lang.String-java.util.Map-), [delete](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#delete-java.lang.String-java.lang.Object...-), [delete](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#delete-java.net.URI-), [delete](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#delete-java.net.URL-), [get](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#get--), [get](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#get-java.lang.String-java.util.Map-), [get](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#get-java.lang.String-java.lang.Object...-), [get](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#get-java.net.URI-), [get](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#get-java.net.URL-), [head](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#head--), [head](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#head-java.lang.String-java.util.Map-), [head](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#head-java.lang.String-java.lang.Object...-), [head](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#head-java.net.URI-), [head](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#head-java.net.URL-), [options](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#options--), [options](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#options-java.lang.String-java.util.Map-), [options](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#options-java.lang.String-java.lang.Object...-), [options](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#options-java.net.URI-), [options](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#options-java.net.URL-), [patch](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#patch--), [patch](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#patch-java.lang.String-java.util.Map-), [patch](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#patch-java.lang.String-java.lang.Object...-), [patch](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#patch-java.net.URI-), [patch](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#patch-java.net.URL-), [post](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#post--), [post](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#post-java.lang.String-java.util.Map-), [post](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#post-java.lang.String-java.lang.Object...-), [post](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#post-java.net.URI-), [post](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#post-java.net.URL-), [put](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#put--), [put](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#put-java.lang.String-java.util.Map-), [put](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#put-java.lang.String-java.lang.Object...-), [put](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#put-java.net.URI-), [put](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSenderOptions.html#put-java.net.URL-)

* + ***Method Detail***
    - **content**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) content(org.hamcrest.Matcher<?> matcher,

org.hamcrest.Matcher<?>... additionalMatchers)

Expect that the response content conforms to one or more Hamcrest matchers. E.g.

// Expect that the response content (body) contains the string "winning-numbers"

expect().content(containsString("winning-numbers")).when().get("/lotto");

// Expect that the response content (body) contains the string "winning-numbers" and "winners"

expect().content(containsString("winning-numbers"), containsString("winners")).when().get("/lotto");

**Parameters:**

matcher - The hamcrest matcher that must response content must match.

additionalMatchers - Optionally additional hamcrest matchers that must return true.

**Returns:**

the response specification

* + - **content**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) content([List](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[Argument](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments,
    - org.hamcrest.Matcher matcher,

[Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)

This as special kind of expectation that is mainly useful when you've specified a root path with an argument placeholder. For example:

expect().

root("x.%s"). // Root path with a placeholder

content(withArgs("firstName"), equalTo(..)).

content(withArgs("lastName"), equalTo(..)).

when().

get(..);

Note that this is the same as doing:

expect().

root("x.%s"). // Root path with a placeholder

content(withArgs("firstName"), equalTo(..)).

content(withArgs("lastName"), equalTo(..)).

when().

get(..);

Note that this method is the same as [body(java.util.List, org.hamcrest.Matcher, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.util.List-org.hamcrest.Matcher-java.lang.Object...-) but with a method name.

**Parameters:**

arguments - The arguments to apply to the root path.

matcher - The hamcrest matcher that must response body must match.

additionalKeyMatcherPairs - Optionally additional hamcrest matchers that must return true.

**Returns:**

the response specification

**See Also:**

[body(String, org.hamcrest.Matcher, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.lang.String-org.hamcrest.Matcher-java.lang.Object...-)

* + - **validate**

[Response](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/response/Response.html) validate([Response](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/response/Response.html) response)

Validates the specified response against this ResponseSpecification

**Parameters:**

response - The response to validate

**Returns:**

The same response

* + - **content**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) content([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) key,
    - org.hamcrest.Matcher<?> matcher,

[Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)

Expect that the JSON or XML response content conforms to one or more Hamcrest matchers.

**JSON example**

Assume that a GET request to "/lotto" returns a JSON response containing:

{ "lotto":{

"lottoId":5,

"winning-numbers":[2,45,34,23,7,5,3],

"winners":[{

"winnerId":23,

"numbers":[2,45,34,23,3,5]

},{

"winnerId":54,

"numbers":[52,3,12,11,18,22]

}]

}}

You can verify that the lottoId is equal to 5 like this:

expect().content("lotto.lottoId", equalTo(5)).when().get("/lotto");

You can also verify that e.g. one of the the winning numbers is 45.

expect().content("lotto.winning-numbers", hasItem(45)).when().get("/lotto");

Or both at the same time:

expect().content("lotto.lottoId", equalTo(5)).and().content("lotto.winning-numbers", hasItem(45)).when().get("/lotto");

or a slightly short version:

expect().content("lotto.lottoId", equalTo(5), "lotto.winning-numbers", hasItem(45)).when().get("/lotto");

**XML example**

Assume that a GET request to "/xml" returns a XML response containing:

<greeting>

<firstName>John</firstName>

<lastName>Doe</lastName>

</greeting>

You can now verify that the firstName is equal to "John" like this:

expect().content("greeting.firstName", equalTo("John")).when().get("/xml");

To verify both the first name and last name you can do like this:

expect().content("greeting.firstName", equalTo("John")).and().content("greeting.lastName", equalTo("Doe")).when().get("/xml");

Or the slightly shorter version of:

expect().content("greeting.firstName", equalTo("John"), "greeting.lastName", equalTo("Doe")).when().get("/xml");

**Notes**

Note that if the response content type is not of type application/xml or application/json you *cannot* use this verification.

The only difference between the content and body methods are of syntactic nature.

**Parameters:**

matcher - The hamcrest matcher that must response content must match.

additionalKeyMatcherPairs - Optionally additional hamcrest matchers that must return true.

**Returns:**

the response specification

* + - **body**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) body([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) key,
    - [List](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[Argument](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments,
    - org.hamcrest.Matcher matcher,

[Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)

Same as [body(String, org.hamcrest.Matcher, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.lang.String-org.hamcrest.Matcher-java.lang.Object...-) expect that you can pass arguments to the key. This is useful in situations where you have e.g. pre-defined variables that constitutes the key:

String someSubPath = "else";

int index = 1;

expect().body("something.%s[%d]", withArgs(someSubPath, index), equalTo("some value")). ..

or if you have complex root paths and don't wish to duplicate the path for small variations:

expect().

root("filters.filterConfig[%d].filterConfigGroups.find { it.name == 'Gold' }.includes").

body(withArgs(0), hasItem("first")).

body(withArgs(1), hasItem("second")).

..

The key and arguments follows the standard [formatting syntax](https://download.oracle.com/javase/1,5.0/docs/api/java/util/Formatter.html#syntax) of Java.

Note that withArgs can be statically imported from the com.jayway.restassured.RestAssured class.

**Parameters:**

key - The body key

arguments - The arguments to apply to the key

matcher - The hamcrest matcher that must response body must match.

additionalKeyMatcherPairs - Optionally additional hamcrest matchers that must return true.

**Returns:**

the response specification

**See Also:**

[body(String, org.hamcrest.Matcher, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.lang.String-org.hamcrest.Matcher-java.lang.Object...-)

* + - **body**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) body([List](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[Argument](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments,
    - org.hamcrest.Matcher matcher,

[Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)

This as special kind of expectation that is mainly useful when you've specified a root path with an argument placeholder. For example:

expect().

root("x.%s"). // Root path with a placeholder

body(withArgs("firstName"), equalTo(..)).

body(withArgs("lastName"), equalTo(..)).

when().

get(..);

Note that this is the same as doing:

expect().

root("x.%s"). // Root path with a placeholder

body(withArgs("firstName"), equalTo(..)).

body(withArgs("lastName"), equalTo(..)).

when().

get(..);

Note that this method is the same as [content(java.util.List, org.hamcrest.Matcher, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#content-java.util.List-org.hamcrest.Matcher-java.lang.Object...-) but with a method name.

**Parameters:**

arguments - The arguments to apply to the root path.

matcher - The hamcrest matcher that must response body must match.

additionalKeyMatcherPairs - Optionally additional hamcrest matchers that must return true.

**Returns:**

the response specification

**See Also:**

[body(String, org.hamcrest.Matcher, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.lang.String-org.hamcrest.Matcher-java.lang.Object...-)

* + - **statusCode**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) statusCode(org.hamcrest.Matcher<? super [Integer](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Integer.html?is-external=true)> expectedStatusCode)

Expect that the response status code matches the given Hamcrest matcher. E.g.

expect().statusCode(equalTo(200)).when().get("/something");

**Parameters:**

expectedStatusCode - The expected status code matcher.

**Returns:**

the response specification

* + - **statusCode**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) statusCode(int expectedStatusCode)

Expect that the response status code matches an integer. E.g.

expect().statusCode(200).when().get("/something");

This is the same as:

expect().statusCode(equalTo(200)).when().get("/something");

**Parameters:**

expectedStatusCode - The expected status code.

**Returns:**

the response specification

* + - **statusLine**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) statusLine(org.hamcrest.Matcher<? super [String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true)> expectedStatusLine)

Expect that the response status line matches the given Hamcrest matcher. E.g.

expect().statusLine(equalTo("No Content")).when().get("/something");

**Parameters:**

expectedStatusLine - The expected status line matcher.

**Returns:**

the response specification

* + - **statusLine**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) statusLine([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) expectedStatusLine)

Expect that the response status line matches the given String. E.g.

expect().statusLine("No Content").when().get("/something");

This is the same as:

expect().statusLine(equalTo("No Content")).when().get("/something");

**Parameters:**

expectedStatusLine - The expected status line.

**Returns:**

the response specification

* + - **headers**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) headers([Map](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/Map.html?is-external=true)<[String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true),?> expectedHeaders)

Expect that response headers matches those specified in a Map.

E.g. expect that the response of the GET request to "/something" contains header headerName1=headerValue1 and headerName2=headerValue2:

Map expectedHeaders = new HashMap();

expectedHeaders.put("headerName1", "headerValue1"));

expectedHeaders.put("headerName2", "headerValue2");

expect().response().headers(expectedHeaders).when().get("/something");

You can also use Hamcrest matchers:

Map expectedHeaders = new HashMap();

expectedHeaders.put("Content-Type", containsString("charset=UTF-8"));

expectedHeaders.put("Content-Length", "160");

expect().headers(expectedHeaders).when().get("/something");

**Parameters:**

expectedHeaders - The Map of expected response headers

**Returns:**

the response specification

* + - **headers**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) headers([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) firstExpectedHeaderName,
    - [Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true) firstExpectedHeaderValue,

[Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... expectedHeaders)

Expect that response headers matches the supplied headers and values.

E.g. expect that the response of the GET request to "/something" contains header Pragma=no-cache and Content-Encoding=gzip:

expect().headers("Pragma", "no-cache", "Content-Encoding", "gzip").when().get("/something");

You can also use Hamcrest matchers:

expect().response().headers("Content-Type", containsString("application/json"), "Pragma", equalsTo("no-cache")).when().get("/something");

and you can even mix string matching and hamcrest matching:

expect().headers("Content-Type", containsString("application/json"), "Pragma", "no-cache").when().get("/something");

**Parameters:**

firstExpectedHeaderName - The name of the first header

firstExpectedHeaderValue - The value of the first header

expectedHeaders - A list of expected "header name" - "header value" pairs.

**Returns:**

the response specification

* + - **header**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) header([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) headerName,

org.hamcrest.Matcher<?> expectedValueMatcher)

Expect that a response header matches the supplied header name and hamcrest matcher.

E.g. expect that the response of the GET request to "/something" contains header Pragma=no-cache:

expect().header("Pragma", containsString("no")).when().get("/something");

You can also expect several headers:

expect().header("Pragma", equalsTo("no-cache")),and().header("Content-Encoding", containsString("zip")).when().get("/something");

Also take a look at [headers(String, Object, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#headers-java.lang.String-java.lang.Object-java.lang.Object...-) )} for a short version of passing multiple headers.

**Parameters:**

headerName - The name of the expected header

expectedValueMatcher - The Hamcrest matcher that must conform to the value

**Returns:**

the response specification

* + - **header**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) header([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) headerName,

[String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) expectedValue)

Expect that a response header matches the supplied name and value.

E.g. expect that the response of the GET request to "/something" contains header Pragma=no-cache:

expect().header("Pragma", "no-cache").when().get("/something");

You can also expect several headers:

expect().header("Pragma", "no-cache"),and().header("Content-Encoding", "gzip").when().get("/something");

Also take a look at [headers(String, Object, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#headers-java.lang.String-java.lang.Object-java.lang.Object...-) for a short version of passing multiple headers.

**Parameters:**

headerName - The name of the expected header

expectedValue - The value of the expected header

**Returns:**

the response specification

* + - **cookies**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) cookies([Map](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/Map.html?is-external=true)<[String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true),?> expectedCookies)

Expect that response cookies matches those specified in a Map.

E.g. expect that the response of the GET request to "/something" contains cookies cookieName1=cookieValue1 and cookieName2=cookieValue2:

Map expectedCookies = new HashMap();

expectedCookies.put("cookieName1", "cookieValue1"));

expectedCookies.put("cookieName2", "cookieValue2");

expect().response().cookies(expectedCookies).when().get("/something");

You can also use Hamcrest matchers:

Map expectedCookies = new HashMap();

expectedCookies.put("cookieName1", containsString("Value1"));

expectedCookies.put("cookieName2", "cookieValue2");

expect().cookies(expectedCookies).when().get("/something");

**Parameters:**

expectedCookies - A Map of expected response cookies

**Returns:**

the response specification

* + - **cookie**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) cookie([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) cookieName)

Expect that a cookie exist in the response, regardless of value (it may have no value at all).

**Parameters:**

cookieName - the cookie to validate that it exists

**Returns:**

the response specification

* + - **cookies**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) cookies([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) firstExpectedCookieName,
    - [Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true) firstExpectedCookieValue,

[Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... expectedCookieNameValuePairs)

Expect that response cookies matches the supplied cookie names and values.

E.g. expect that the response of the GET request to "/something" contains cookies cookieName1=cookieValue1 and cookieName2=cookieValue2:

expect().cookies("cookieName1", "cookieValue1", "cookieName2", "cookieValue2").when().get("/something");

You can also use Hamcrest matchers:

expect().response().cookies("cookieName1", containsString("Value1"), "cookieName2", equalsTo("cookieValue2")).when().get("/something");

and you can even mix string matching and hamcrest matching:

expect().cookies("cookieName1", containsString("Value1"), "cookieName2", "cookieValue2").when().get("/something");

**Parameters:**

firstExpectedCookieName - The name of the first cookie

firstExpectedCookieValue - The value of the first cookie

expectedCookieNameValuePairs - A list of expected "cookie name" - "cookie value" pairs.

**Returns:**

the response specification

* + - **cookie**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) cookie([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) cookieName,

org.hamcrest.Matcher<?> expectedValueMatcher)

Expect that a response cookie matches the supplied cookie name and hamcrest matcher.

E.g. expect that the response of the GET request to "/something" contain cookie cookieName1=cookieValue1

expect().cookie("cookieName1", containsString("Value1")).when().get("/something");

You can also expect several cookies:

expect().cookie("cookieName1", equalsTo("cookieValue1")),and().cookie("cookieName2", containsString("Value2")).when().get("/something");

Also take a look at [cookies(String, Object, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#cookies-java.lang.String-java.lang.Object-java.lang.Object...-) for a short version of passing multiple cookies.

**Parameters:**

cookieName - The name of the expected cookie

expectedValueMatcher - The Hamcrest matcher that must conform to the value

**Returns:**

the response specification

* + - **cookie**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) cookie([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) cookieName,

[Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true) expectedValue)

Expect that a response cookie matches the supplied name and value.

E.g. expect that the response of the GET request to "/something" contain cookie cookieName1=cookieValue1:

expect().cookie("cookieName1", "cookieValue1").when().get("/something");

You can also expect several cookies:

expect().cookie("cookieName1", "cookieValue1"),and().cookie("cookieName2", "cookieValue2").when().get("/something");

Also take a look at [cookies(String, Object, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#cookies-java.lang.String-java.lang.Object-java.lang.Object...-) for a short version of passing multiple cookies.

**Parameters:**

cookieName - The name of the expected cookie

expectedValue - The value of the expected cookie

**Returns:**

the response specification

* + - **log**

[ResponseLogSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseLogSpecification.html) log()

Returns the [ResponseLogSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseLogSpecification.html) that allows you to log different parts of the [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html). This is mainly useful for debug purposes when writing your tests. It's a shortcut for:

given().filter(ResponseLoggingFilter.responseLogger()). ..

**Returns:**

the response log specification

* + - **rootPath**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) rootPath([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) rootPath)

Set the root path of the response body so that you don't need to write the entire path for each expectation. E.g. instead of writing:

expect().

body("x.y.firstName", is(..)).

body("x.y.lastName", is(..)).

body("x.y.age", is(..)).

body("x.y.gender", is(..)).

when().

get(..);

you can use a root path and do:

expect().

rootPath("x.y").

body("firstName", is(..)).

body("lastName", is(..)).

body("age", is(..)).

body("gender", is(..)).

when().

get(..);

Note that this method is exactly the same as [root(String)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#root-java.lang.String-).

**Parameters:**

rootPath - The root path to use.

* + - **rootPath**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) rootPath([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) rootPath,

[List](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[Argument](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments)

Set the root path with arguments of the response body so that you don't need to write the entire path for each expectation.

Note that this method is exactly the same as [root(String, java.util.List)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#root-java.lang.String-java.util.List-).

**Parameters:**

rootPath - The root path to use.

arguments - A list of arguments. The path and arguments follows the standard [formatting syntax](https://download.oracle.com/javase/1,5.0/docs/api/java/util/Formatter.html#syntax) of Java.

**See Also:**

[rootPath(String)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#rootPath-java.lang.String-)

* + - **root**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) root([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) rootPath,

[List](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[Argument](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments)

Set the root path with arguments of the response body so that you don't need to write the entire path for each expectation.

Note that this method is exactly the same as [rootPath(String, java.util.List)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#rootPath-java.lang.String-java.util.List-).

**Parameters:**

rootPath - The root path to use.

arguments - The list of substitution arguments. The path and arguments follows the standard [formatting syntax](https://download.oracle.com/javase/1,5.0/docs/api/java/util/Formatter.html#syntax) of Java..

**See Also:**

[rootPath(String)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#rootPath-java.lang.String-)

* + - **root**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) root([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) rootPath)

Set the root path of the response body so that you don't need to write the entire path for each expectation. E.g. instead of writing:

expect().

body("x.y.firstName", is(..)).

body("x.y.lastName", is(..)).

body("x.y.age", is(..)).

body("x.y.gender", is(..)).

when().

get(..);

you can use a root and do:

expect().root("x.y").

body("firstName", is(..)).

body("lastName", is(..)).

body("age", is(..)).

body("gender", is(..)).

when().

get(..);

Note that this method is exactly the same as [rootPath(String)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#rootPath-java.lang.String-) but slightly shorter.

**Parameters:**

rootPath - The root path to use.

* + - **noRoot**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) noRoot()

Reset the root path of the response body so that you don't need to write the entire path for each expectation. For example:

expect().

root("x.y").

body("firstName", is(..)).

body("lastName", is(..)).

noRoot()

body("z.something1", is(..)).

body("w.something2", is(..)).

when().

get(..);

This is the same as calling rootPath("") but more expressive. Note that this method is exactly the same as [noRootPath()](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#noRootPath--) but slightly shorter.

**See Also:**

[root(String)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#root-java.lang.String-)

* + - **noRootPath**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) noRootPath()

Reset the root path of the response body so that you don't need to write the entire path for each expectation. For example:

expect().

root("x.y").

body("firstName", is(..)).

body("lastName", is(..)).

noRoot()

body("z.something1", is(..)).

body("w.something2", is(..)).

when().

get(..);

This is the same as calling rootPath("") but more expressive. Note that this method is exactly the same as [noRoot()](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#noRoot--) but slightly more expressive.

**See Also:**

[root(String)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#root-java.lang.String-)

* + - **appendRoot**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) appendRoot([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) pathToAppend)

Append the given path to the root path of the response body so that you don't need to write the entire path for each expectation. E.g. instead of writing:

expect().

root("x.y").

body("age", is(..)).

body("gender", is(..)).

body("name.firstName", is(..)).

body("name.lastName", is(..)).

when().

get(..);

you can use a append root and do:

expect().

root("x.y").

body("age", is(..)).

body("gender", is(..)).

appendRoot("name").

body("firstName", is(..)).

body("lastName", is(..)).

when().

get(..);

**Parameters:**

pathToAppend - The root path to append.

* + - **appendRoot**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) appendRoot([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) pathToAppend,

[List](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[Argument](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments)

Append the given path to the root path with arguments supplied of the response body so that you don't need to write the entire path for each expectation. This is mainly useful when you have parts of the path defined in variables. E.g. instead of writing:

String namePath = "name";

expect().

root("x.y").

body("age", is(..)).

body("gender", is(..)).

body(namePath + "first", is(..)).

body(namePath + "last", is(..)).

when().

get(..);

you can use a append root and do:

String namePath = "name";

expect().

root("x.y").

body("age", is(..)).

body("gender", is(..)).

appendRoot("%s", withArgs(namePath)).

body("first", is(..)).

body("last", is(..)).

when().

get(..);

**Parameters:**

pathToAppend - The root path to use. The path and arguments follows the standard [formatting syntax](https://download.oracle.com/javase/1,5.0/docs/api/java/util/Formatter.html#syntax) of Java.

* + - **detachRoot**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) detachRoot([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) pathToDetach)

Detach the given path from the root path. E.g. instead of writing:

when().

get(..);

then().

root("x.y").

body("age", is(..)).

body("gender", is(..)).

root("x").

body("firstName", is(..)).

body("lastName", is(..)).

you can use a append root and do:

when().

get(..);

then().

root("x.y").

body("age", is(..)).

body("gender", is(..)).

detachRoot("y").

body("firstName", is(..)).

body("lastName", is(..)).

**Parameters:**

pathToDetach - The root path to detach.

* + - **contentType**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) contentType([ContentType](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/http/ContentType.html) contentType)

Set the response content type to be contentType.

**Parameters:**

contentType - The content type of the response.

**Returns:**

the response specification

* + - **contentType**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) contentType([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) contentType)

Set the response content type to be contentType.

**Parameters:**

contentType - The content type of the response.

**Returns:**

the response specification

* + - **contentType**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) contentType(org.hamcrest.Matcher<? super [String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true)> contentType)

Expect the response content type to be contentType.

**Parameters:**

contentType - The expected content type of the response.

**Returns:**

the response specification

* + - **body**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) body(org.hamcrest.Matcher<?> matcher,

org.hamcrest.Matcher<?>... additionalMatchers)

Expect that the response body conforms to one or more Hamcrest matchers. E.g.

// Expect that the response body (content) contains the string "winning-numbers"

expect().body(containsString("winning-numbers")).when().get("/lotto");

// Expect that the response body (content) contains the string "winning-numbers" and "winners"

expect().body(containsString("winning-numbers"), containsString("winners")).when().get("/lotto");

**Parameters:**

matcher - The hamcrest matcher that must response body must match.

additionalMatchers - Optionally additional hamcrest matchers that must return true.

**Returns:**

the response specification

* + - **body**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) body([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) path,
    - org.hamcrest.Matcher<?> matcher,

[Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)

Expect that the JSON or XML response body conforms to one or more Hamcrest matchers.

**JSON example**

Assume that a GET request to "/lotto" returns a JSON response containing:

{ "lotto":{

"lottoId":5,

"winning-numbers":[2,45,34,23,7,5,3],

"winners":[{

"winnerId":23,

"numbers":[2,45,34,23,3,5]

},{

"winnerId":54,

"numbers":[52,3,12,11,18,22]

}]

}}

You can verify that the lottoId is equal to 5 like this:

expect().body("lotto.lottoId", equalTo(5)).when().get("/lotto");

You can also verify that e.g. one of the the winning numbers is 45.

expect().body("lotto.winning-numbers", hasItem(45)).when().get("/lotto");

Or both at the same time:

expect().body("lotto.lottoId", equalTo(5)).and().body("lotto.winning-numbers", hasItem(45)).when().get("/lotto");

or a slightly short version:

expect().body("lotto.lottoId", equalTo(5), "lotto.winning-numbers", hasItem(45)).when().get("/lotto");

**XML example**

Assume that a GET request to "/xml" returns a XML response containing:

<greeting>

<firstName>John</firstName>

<lastName>Doe</lastName>

</greeting>

You can now verify that the firstName is equal to "John" like this:

expect().body("greeting.firstName", equalTo("John")).when().get("/xml");

To verify both the first name and last name you can do like this:

expect().body("greeting.firstName", equalTo("John")).and().body("greeting.lastName", equalTo("Doe")).when().get("/xml");

Or the slightly shorter version of:

expect().body("greeting.firstName", equalTo("John"), "greeting.lastName", equalTo("Doe")).when().get("/xml");

**Notes**

Note that if the response body type is not of type application/xml or application/json you *cannot* use this verification.

The only difference between the content and body methods are of syntactic nature.

**Parameters:**

path - The body path

matcher - The hamcrest matcher that must response body must match.

additionalKeyMatcherPairs - Optionally additional hamcrest matchers that must return true.

**Returns:**

the response specification

* + - **content**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) content([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) path,
    - [List](https://docs.oracle.com/javase/1.5.0/docs/api/java/util/List.html?is-external=true)<[Argument](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/Argument.html)> arguments,
    - org.hamcrest.Matcher matcher,

[Object](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html?is-external=true)... additionalKeyMatcherPairs)

Same as [body(String, java.util.List, org.hamcrest.Matcher, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#body-java.lang.String-java.util.List-org.hamcrest.Matcher-java.lang.Object...-) expect that you can pass arguments to the path. This is useful in situations where you have e.g. pre-defined variables that constitutes the path:

String someSubPath = "else";

int index = 1;

expect().body("something.%s[%d]", withArgs(someSubPath, index), equalTo("some value")). ..

or if you have complex root paths and don't wish to duplicate the path for small variations:

expect().

root("filters.filterConfig[%d].filterConfigGroups.find { it.name == 'Gold' }.includes").

body(withArgs(0), hasItem("first")).

body(withArgs(1), hasItem("second")).

..

The path and arguments follows the standard [formatting syntax](https://download.oracle.com/javase/1,5.0/docs/api/java/util/Formatter.html#syntax) of Java.

Note that withArgs can be statically imported from the com.jayway.restassured.RestAssured class.

**Parameters:**

path - The body path

matcher - The hamcrest matcher that must response body must match.

additionalKeyMatcherPairs - Optionally additional hamcrest matchers that must return true.

**Returns:**

the response specification

**See Also:**

[content(String, org.hamcrest.Matcher, Object...)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#content-java.lang.String-org.hamcrest.Matcher-java.lang.Object...-)

* + - **when**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) when()

Syntactic sugar, e.g.

expect().body(containsString("OK")).when().get("/something");

is that same as:

expect().body(containsString("OK")).get("/something");

**Returns:**

the response specification

* + - **given**

[RequestSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSpecification.html) given()

Returns the request com.jayway.restassured.specification so that you can define the properties of the request.

expect().body(containsString("OK")).given().parameters("param1", "value1").when().get("/something");

**Returns:**

the request com.jayway.restassured.specification

* + - **that**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) that()

Syntactic sugar, e.g.

expect().that().body(containsString("OK")).when().get("/something");

is that same as:

expect().body(containsString("OK")).get("/something");

**Returns:**

the response specification

* + - **request**

[RequestSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSpecification.html) request()

Returns the request com.jayway.restassured.specification so that you can define the properties of the request.

expect().body(containsString("OK")).and().request().parameters("param1", "value1").when().get("/something");

**Returns:**

the request com.jayway.restassured.specification

* + - **response**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) response()

Syntactic sugar, e.g.

expect().response().body(containsString("OK")).when().get("/something");

is that same as:

expect().body(containsString("OK")).get("/something");

**Returns:**

the response specification

* + - **and**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) and()

Syntactic sugar, e.g.

expect().body(containsString("OK")).and().body(containsString("something else")).when().get("/something");

is that same as:

expect().body(containsString("OK")).body(containsString("something else")).when().get("/something");

**Returns:**

the response specification

* + - **with**

[RequestSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/RequestSpecification.html) with()

Returns the request com.jayway.restassured.specification so that you can define the properties of the request.

expect().body(containsString("OK")).and().with().request().parameters("param1", "value1").get("/something");

**Returns:**

the request com.jayway.restassured.specification

* + - **then**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) then()

Syntactic sugar, e.g.

expect().body(containsString("OK")).then().get("/something");

is that same as:

expect().body(containsString("OK")).get("/something");

**Returns:**

the response specification

* + - **expect**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) expect()

Syntactic sugar, e.g.

expect().body(containsString("OK")).and().expect().body(containsString("something else")).when().get("/something");

is that same as:

\* expect().body(containsString("OK")).and().body(containsString("something else")).when().get("/something");

**Returns:**

the response specification

* + - **spec**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) spec([ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) responseSpecificationToMerge)

Expect that the response matches an entire specification.

ResponseSpecification responseSpec = new ResponseSpecBuilder().expectStatusCode(200).build();

expect().

spec(responseSpec).

body("x.y.z", equalTo("something")).

when().

get("/something");

This is useful when you want to reuse multiple different expectations in several tests.

The specification passed to this method is merged with the current specification. Note that the supplied specification can overwrite data in the current specification. The following settings are overwritten:

* + - * Content type
      * Root pathStatus code
      * Status line

The following settings are merged:

* + - * Response body expectations
      * Cookies
      * Headers

This method is the same as [specification(ResponseSpecification)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#specification-com.jayway.restassured.specification.ResponseSpecification-) but the name is a bit shorter.

**Parameters:**

responseSpecificationToMerge - The specification to merge with.

**Returns:**

the response specification

* + - **specification**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) specification([ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) responseSpecificationToMerge)

Expect that the response matches an entire specification.

ResponseSpecification responseSpec = new ResponseSpecBuilder().expectStatusCode(200).build();

expect().

specification(responseSpec).

body("x.y.z", equalTo("something")).

when().

get("/something");

This is useful when you want to reuse multiple different expectations in several tests.

The specification passed to this method is merged with the current specification. Note that the supplied specification can overwrite data in the current specification. The following settings are overwritten:

* + - * Content type
      * Root pathStatus code
      * Status line

The following settings are merged:

* + - * Response body expectations
      * Cookies
      * Headers

This method is the same as [spec(ResponseSpecification)](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html#spec-com.jayway.restassured.specification.ResponseSpecification-) but the name is a bit longer.

**Parameters:**

responseSpecificationToMerge - The specification to merge with.

**Returns:**

the response specification

* + - **parser**
    - [ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) parser([String](https://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html?is-external=true) contentType,

[Parser](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/parsing/Parser.html) parser)

Register a content-type to be parsed using a predefined parser. E.g. let's say you want parse content-type application/vnd.uoml+xml with the XML parser to be able to verify the response using the XML dot notations:

expect().body("document.child", equalsTo("something"))..

Since application/vnd.uoml+xml is not registered to be processed by the XML parser by default you need to explicitly tell REST Assured to use this parser before making the request:

expect().parser("application/vnd.uoml+xml", Parser.XML).when(). ..;

You can also specify by it for every response by using:

RestAssured.registerParser("application/vnd.uoml+xml", Parser.XML);

**Parameters:**

contentType - The content-type to register

parser - The parser to use when verifying the response.

* + - **defaultParser**

[ResponseSpecification](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/specification/ResponseSpecification.html) defaultParser([Parser](https://www.javadoc.io/static/com.jayway.restassured/rest-assured/2.4.0/com/jayway/restassured/parsing/Parser.html) parser)

Register a default predefined parser that will be used if no other parser (registered or pre-defined) matches the response content-type. E.g. let's say that for some reason no content-type is defined in the response but the content is nevertheless JSON encoded. To be able to expect the content in REST Assured you need to set the default parser:

expect().defaultParser(Parser.JSON).when(). ..;

You can also specify it for every response by using:

RestAssured.defaultParser(Parser.JSON);

**Parameters:**

parser - The parser to use when verifying the response if no other parser is found for the response content-type.