REST-assured

Johan Haleby

HTTP Builder

```
def  http = new HTTPBuilder( 'http://ajax.googleapis.com' )
// perform a GET request, expecting JSON response data
 http.request( GET, JSON ) {
  uri.path = '/ajax/services/search/web'
  uri.query = [ v:'1.0', q: 'Calvin and Hobbes' ]
  headers.'User-Agent' = 'Mozilla/5.0 Ubuntu/8.10 Firefox/3.0.4'
  // response handler for a success response code:
  response.success = { resp, json ->
    println  resp.statusLine
    // parse the JSON response object:
    json.responseData.results.each {
      println  "  ${it.titleNoFormatting} : ${it.visibleUrl}"
    }
  }
  // handler for any failure status code:
  response.failure = { resp ->
    println  "Unexpected error: ${resp.statusLine.statusCode} : $
{resp.statusLine.reasonPhrase}"
  }
}
```

JAYWAY

POST with HTTP Builder

```
def  http = new HTTPBuilder('http://twitter.com/statuses/')
http.request( POST ) {
  uri.path = 'update.xml'
  body =  [ status : 'update!' , source : 'httpbuilder' ]
  requestContentType = ContentType.URLENC
  response.success = { resp ->
    println  "Tweet response status: ${resp.statusLine}"
    assert  resp.statusLine.statusCode == 200
  }
}
```

JSON Parsing in Groovy

```
{"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]
    }]
  }
}
```

```
json.lotto.lottoId // 5
json.lotto.'winning-numbers' // 2, 45, 24 ..
json.lotto.winners.size() // 2
json.lotto.winners[0] // “First winner”
```

REST Assured - Requests

```
get("/lotto");
post("/lotto");
delete("/lotto");
put("/lotto");
head("/lotto");
```

Defaults to localhost:8080

JAYWAY

REST Assured - JSON

```
{"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]
    }]
  }
}
```

```
expect().body("lotto.lottoId", equalTo(5)).when().get("/lotto");
expect().body("lotto.winners.winnerId", hasItems(23,54)).when().get("/lotto");
expect().body("lotto.winners.size()", is(2)).when().get("/lotto");
expect().body("lotto.winners.winnerId[0]", is(23)).when().get("/lotto");
expect().body("lotto.winner-numbers", hasItems(2,45, ..)).get("/lotto");
```

JAYWAY

REST Assured - JSON

```
expect().
        body("lotto.lottoId", equalTo(5)).
        body("lotto.winners.winnerId", hasItems(23,54)).
        body("lotto.winners.size()", is(2)).
        body("lotto.winners.winnerId[0]", is(23)).
        body("lotto.winners.winner-numbers", hasItems(2,45, ..)).
when().
        get("/lotto");
```

JAYWAY

REST Assured - XML

```
<records>
      <car name='HSV Maloo' make='Holden' year='2006'>
        <country>Australia</country>
        <record type='speed'>Pickup Truck with speed of 271kph</record>
      </car>
      <car name='P50' make='Peel' year='1962'>
        <country>Isle of Man</country>
        <record type='size'>Street-Legal Car at 99cm wide, 59kg</record>
      </car>
      <car name='Royale' make='Bugatti' year='1931'>
        <country>France</country>
        <record type='price'>Most Valuable Car at $15 million</record>
      </car>
</records>
```

```
expect().
        body("records.car[0].@name", equalTo("HSV Maloo")).
        body("records.car[-1].country", equalTo("France")).
        body("records.car[0..1].@year", hasItems("2006","1962")).
        body("records.car.country", hasItems("Australia","Isle ..","France")).
when().
        get("/xml");
```

JAYWAY

REST Assured – XML (X-Path)

```
<records>
      <car name='HSV Maloo' make='Holden' year='2006'>
        <country>Australia</country>
        <record type='speed'>Pickup Truck with speed of 271kph</record>
      </car>
      <car name='P50' make='Peel' year='1962'>
        <country>Isle of Man</country>
        <record type='size'>Street-Legal Car at 99cm wide, 59kg</record>
      </car>
      <car name='Royale' make='Bugatti' year='1931'>
        <country>France</country>
        <record type='price'>Most Valuable Car at $15 million</record>
      </car>
</records>
```

```
expect().
        body(hasXPath("/records/car[1]/@name"), equalTo("HSV Maloo")).
        body(hasXPath("/records/car[1]/country[text()='Australia'"")).
when().
        get("/xml");
```

JAYWAY

REST Assured – XSD & DTD

```
<records>
      <car name='HSV Maloo' make='Holden' year='2006'>
        <country>Australia</country>
        <record type='speed'>Pickup Truck with speed of 271kph</record>
      </car>
      <car name='P50' make='Peel' year='1962'>
        <country>Isle of Man</country>
        <record type='size'>Street-Legal Car at 99cm wide, 59kg</record>
      </car>
      <car name='Royale' make='Bugatti' year='1931'>
        <country>France</country>
        <record type='price'>Most Valuable Car at $15 million</record>
      </car>
</records>
```

```
expect().body(matchesDtd(dtd)).when().get("/xml");
expect().body(matchesXsd(xsd)).when().get("/xml");
```

Parameters

```
given().
        param("name1", "value1").
        param("name2", "value2").
expect().
        statusCode(200).
when().
        get("/something");
```

```
given().params("name1","value1","name2", "value2").when().get("/something");
```

```
get("/something?name1=value1&name2=value2");
```

```
Map<String, String> parameters = new HashMap<~>(..);
given().params(parameters).when().get("/something");
```

```
// URL encoding happens automatically
given().param("name", "value").when().post("/something");
```

JAYWAY

Request Headers & Cookies

```
given().
        header("headerName", "value").
        cookie("cookieName", "value").
expect().
        statusCode(200).
when().
        get("/something");
```

```
given().headers("name1","value1","name2", "value2").when().get("/something");
given().cookies("name1","value1","name2", "value2").when().get("/something");

Map<String, String> headers = new HashMap<~>(..);
given().headers(headers).when().get("/something");

Map<String, String> cookies = new HashMap<~>(..);
given().cookies(cookies).when().get("/something");
```

Validate Headers & Cookies

```
expect().header("name", "value").when().get("/something");
expect().cookie("name", containsString("value")).when().get("/something");

expect().headers("name1", "value1", “name2”,containsString("value"))..
expect().cookies("name1", "value1", “name2”,containsString("value"))..

Map<String, Object> expected = new HashMap<~>(..);
expected.put(“name1”, “value1”);
expected.put(“name1”, containsString(“something”);
expect().headers(expected).when().get(“/something”);
```

JAYWAY

REST Assured - Request Body

```
// String
given().body("{ \"message\" : \"helloworld\"}").when().post("/jsonBody");

// Binary
byte[] body = { 23, 42, 127, 123};
given().body(body).when().post("/binaryBody");
```

Request & Response Body

```
given().request().body(body).and().expect().response().body(containsString("so
mething") ).when().post("/header");
```

REST Assured - Authentication

- Basic
- Form
- Digest
- OAuth
- Certificate

```
given().auth().basic("jetty","jetty").when().get("/secured/basic");
given().auth().certificate(certURL,"password”).when().get("/secured/cert");
```

JAYWAY

REST Assured – Advanced validation

```
  <records>
        <car name='HSV Maloo' make='Holden' year='2006'>
          <country>Australia</country>
          <record type='speed'>Pickup Truck with speed of 271kph</record>
        </car>
        <car name='P50' make='Peel' year='1962'>
          <country>Isle of Man</country>
          <record type='size'>Street-Legal Car at 99cm wide, 59kg</record>
        </car>
        <car name='Royale' make='Bugatti' year='1931'>
          <country>France</country>
          <record type='price'>Most Valuable Car at $15 million</record>
        </car>
  </records>

expect().
         body("records.car.find { it.record.@type == 'speed' }.country",
equalTo("Australia")).
```

REST Assured - Defaults

```
RestAssured.baseURI = "http://myhost.org";
RestAssured.port = 80;
RestAssured.basePath = "/resource";
RestAssured.authentication = basic("username", "password");
```

```
get("/xml")
```

```
http://myhost.org:80/resource/xml with basic auth
```

```
RestAssured.reset();
```

JAYWAY

REST Assured – Response (1)

```
Response response = get("/lotto");

// Get all headers
Map<String, String> allHeaders = response.getHeaders();

// Get a single header value:
String headerName = response.getHeader("headerName");

// Get all cookies
Map<String, String> allCookies = response.getCookies();

// Get a single cookie value:
String cookieValue = response.getCookie("cookieName");

// Get status line
String statusLine = response.getStatusLine();

// Get status code
int statusCode = response.getStatusCode();
```

REST Assured – Response (2)

```
Response response = get("/lotto");
String responseBody = response.getBody().asString();

String body = get("/lotto").asString();
```

REST Assured – JSON Path

```
{"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]
    }]
  }
}
```

```
String json = get("/lotto").asString();

// Static import "with" from JsonPath
int lottoId = with(json).getInt("lotto.lottoId);
List<Integer> winnerIds = with(json).get("lotto.winners.winnerId");

// Or a bit more efficiently
JsonPath jsonPath = new JsonPath(json).setRoot("lotto");
int lottoId = jsonPath.getInt("lottoId");
List<Integer> winnerIds = jsonPath.get("winners.winnderId");
```

JAYWAY

REST Assured – XML Path

```
<records>
      <car name='HSV Maloo' make='Holden' year='2006'>
        <country>Australia</country>
        <record type='speed'>Pickup Truck with speed of 271kph</record>
      </car>
      <car name='P50' make='Peel' year='1962'>
        <country>Isle of Man</country>
        <record type='size'>Street-Legal Car at 99cm wide, 59kg</record>
      </car>
</records>
```

```
String xml = get("/xml").asString();

// Static import "with" from XmlPath
String country = with(xml).get("records.car[0].country");
List<String> carNames = with(xml).get("records.car.@name");

// Or a bit more efficiently
XmlPath xmlPath = new XmlPath(xml).setRoot("records");
String country = xmlPath.get("car[0].country");
List<String> carNames = xmlPath.get("car.@name");
```

REST Assured – more..

- Specify parser for mime-type
- Content type
- Status codes
- Status line
- ..

Web Page & DEMO


JAYWAY