bidij

Behavior Driven Infrastructure support for Java

Table of content

1. Behavior Driven Infrastructure

> *When you write code, you always use unit testing & integration testing to verify that the application is working as expected, but why don't we use that when we install a system?*
>
> *What are you using to verify that your system is correctly configured and behaves the way you want?*

1.1. References

- Behavior Driven Infrastructure - Martin Englund
- Behaviour Driven Infrastructure through Cucumber - Lindsay Holmwood
- Yes Mum, I'll Behave: Beginning Behaviour Driven Infrastructure - James Turnbull

1.2. Grammar

```
Given ^an host set to "([^"]*)"$
```

```
Given ^the request's header "([^"]*)" has been set to "([^"]*)"$
```

```
Given ^(?:a|the) "([^"]*)" header set to "([^"]*)"$
```

```
Given ^the following request's header:$
```

```
Given ^the request method has been set to (.*)$
```

```
Given ^the request has no cookie named (.*)$
```

```
Given ^the request cookie "([^"]*)" has been set to "([^"]*)"$
```

```
Given ^basic auth credentials set to "([^"]*)" and "([^"]*)"$
```

Defined in the HTTP/1.1 Standard, section 14.1, the Accept: header lists the MIME Types of the media that the agent is willing to process. It is comma-separated lists of MIME type, each combined with a quality factor, as parameters giving the relative degree of preference between the different MIME Types lists.

- Content negotiation - MDN
- rfc2616 section 12

```
Given ^a content format negotiation set to "([^"]*)"$
```

When ^a request is made to "([^"]+)"$

When ^a ([a-zA-Z]+) request is made to "([^"]+)"$

When ^a ([a-zA-Z]+) request is made to "([^"]+)" with the following parameters:$

When ^a ([a-zA-Z]+) request is made to "([^"]+)" with content type "([^"]+)" with:$

When ^a json ([a-zA-Z]+) request is made to "([^"]+)" with:$

When ^I follow redirect$

Then ^the response status code should be (\d+)$

Then ^the response status code should be (\d+) \((.+)\)$

Then ^the response should have the cookie (.*)$

Then ^the response's cookie "([^"]*)" should be set to "([^"]*)"$

Then ^the response's body should (start with|end with|match|contain|be) "(.*)"$

Then ^the response's body should (match|contain|be):$

Then ^the response's body should not (match|contain|be) "([^"]*)"$

Then ^the json response's body should (contain|be):$

Then ^the response's header "([^"]*)" should be set to "([^"]*)"$

Then ^the response's header "([^"]*)" should not be set to "([^"]*)"$

Then ^the response should indicate a redirect$

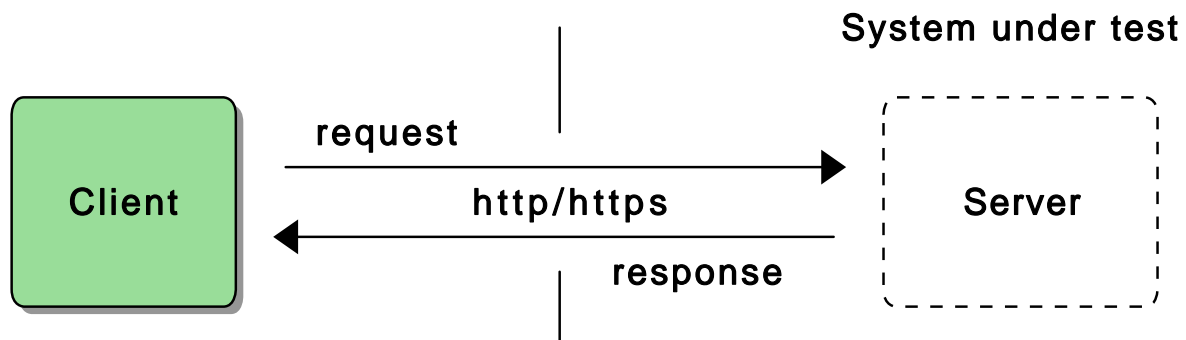Then ^the response should not indicate a redirect$

Given ^the "([^"]*)" jdbc configuration has been applied$

`Given` ^the following jdbc configurations:$

`When` ^a query is made on table "([^"]*)"$

`Then` ^the number of rows returned should be (greater than|greater than or equal to|equal to|lesser than|lesser than or equal to) (\d+)$

---

2. HTTP

In order to ease the testing of HTTP server

As an automation tester

I want some predefined steps to write and to check HTTP request and HTTP response.

2.1. Check the HTTP steps

Uri: bdi/glue/http/testdefs/01-get.feature

2.1.1. Simple GET using host

✅ **Given** an host set to "http://localhost:8080"

✅ **When** a GET request is made to "/users"

✅ **Then** the response status code should be 200

2.1.2. Simple GET using complete uri

✅ **When** a GET request is made to "http://localhost:8080/users"

✅ **Then** the response status code should be 200

2.1.3. Supported Verb: GET

Tags: @http

✅ **When** a GET request is made to "http://localhost:8080/users"

✅ **Then** the response status code should be 200

2.1.4. Supported Verb: PUT

Tags: @http

✅ **When** a PUT request is made to "http://localhost:8080/users"

✅ **Then** the response status code should be 405

2.1.5. Supported Verb: POST

- ✅   When a `POST` request is made to "`http://localhost:8080/users`"
- ✅   Then the response status code should be `400`

2.1.6. Supported Verb: DELETE

- ✅   When a `DELETE` request is made to "`http://localhost:8080/users`"
- ✅   Then the response status code should be `405`

2.1.7. Supported Verb: HEAD

- ✅   When a `HEAD` request is made to "`http://localhost:8080/users`"
- ✅   Then the response status code should be `200`

2.1.8. Supported Verb: OPTIONS

- ✅   When a `OPTIONS` request is made to "`http://localhost:8080/users`"
- ✅   Then the response status code should be `200`

2.1.9. Supported Verb: TRACE

- ✅   When a `TRACE` request is made to "`http://localhost:8080/users`"
- ✅   Then the response status code should be `200`

2.1.10. Supported Verb: PATCH

- ✅   When a `PATCH` request is made to "`http://localhost:8080/users`"
- ✅   Then the response status code should be `501`

2.2. HTTPS steps

2.2.1. GET on app with https configuration set

- ✅   Given a sample server running on port `8080` and on secure port `8083`
- ✅   Given an host set to "`http://localhost:8080`"
- ✅   When a `GET` request is made to "`/users`"
- ✅   Then the response status code should be `200`

2.2.2. GET on https

✅   Given a sample server running on port 8080 and on secure port 8083

✅   And an host set to "https://localhost:8083"

✅   When a GET request is made to "/users"

✅   Then the response status code should be 200

2.2.3. GET on https with basic auth no credential provided

✅   Given a sample server running on port 8080 and on secure port 8083

✅   And an host set to "https://localhost:8083"

✅   When a GET request is made to "/auth/users"

✅   Then the response status code should be 401 (Unauthorized)

2.2.4. GET on https with basic auth using default credentials

✅   Given a sample server running on port 8080 and on secure port 8083

✅   Given an host set to "https://localhost:8083"

✅   And default basic auth credentials set

✅   When a GET request is made to "/auth/users"

✅   Then the response status code should be 200 (OK)

2.2.5. GET on https with basic auth specifying credentials

✅   Given a sample server running on port 8080 and on secure port 8083

✅   Given an host set to "https://localhost:8083"

✅   And basic auth credentials set to "carmen" and "mccallum"

✅   When a GET request is made to "/auth/users"

✅   Then the response status code should be 200 (OK)

2.2.6. GET on https with basic auth specifying credentials

✅   Given a sample server running on port 8080 and on secure port 8083

✅   Given an host set to "https://localhost:8083"

✅   And basic auth credentials set to "carmen" and "mccallum"

✅   When a GET request is made to "/auth/users" with the following parameters:

parameter name	parameter value
offset	50
limit	10
filter	{ :name "/john (.*)/"}

✅   Then the response status code should be 200 (OK)

2.3. Headers and Content Negociation

Uri: bdi/glue/http/testdefs/03-content-negociation.feature

2.3.1. GET with json content negotiation

- ✅ **Given** a sample server running on port `8080`
- ✅ **Given** an host set to "`http://localhost:8080`"
- ✅ **And** a content format negotiation set to "`application/json`"
- ✅ **When** a `GET` request is made to "`/about`"
- ✅ **Then** the response status code should be `200` (`OK`)
- ✅ **And** the json response's body should `contain`:

```
{ "version" : "1.2.0" }
```

2.3.2. GET with html content negocation

- ✅ **Given** a sample server running on port `8080`
- ✅ **Given** an host set to "`http://localhost:8080`"
- ✅ **And** a content format negotiation set to "`text/html`"
- ✅ **When** a `GET` request is made to "`/about`"
- ✅ **Then** the response status code should be `200` (`OK`)
- ✅ **And** the response's body should `start with` "`<html`"

## 2.4. POST request

Uri: bdi/glue/http/testdefs/04-post.feature

2.4.1. A POST with json response

- ✅ **Given** a sample server running on port `8080`
- ✅ **And** an host set to "`http://localhost:8080`"
- ✅ **When** a `POST` request is made to "`/users`" with content type "`application/json`" with:

```
{
  "firstname":"Carmen",
  "lastname":"Mccallum"
}
```

- ✅ **Then** the response status code should be `201`
- ✅ **And** the json response's body should `contain`:

```
{
  "status":"ok"
}
```

3. Jdbc
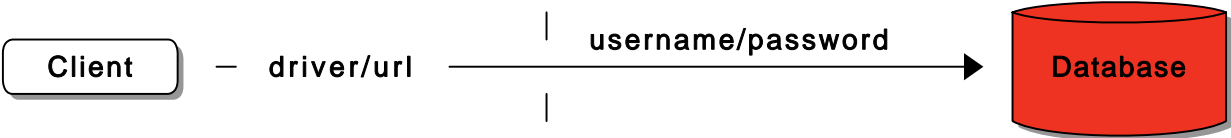
In order to ease the testing of Database

As an automation tester

I want some predefined steps to access and query Database

3.1. Check the JDBC steps
Uri: bdi/glue/jdbc/testdefs/01-jdbc.feature

3.1.1. Simple Access

✅ **Given** a sample database running and defining "default" jdbc configuration

✅ **Given** the "default" jdbc configuration has been applied

✅ **When** a query is made on table "user"

✅ **Then** the number of rows returned should be greater than 0

3.1.2. Jdbc Configuration
Tags: @wip

✅ **Given** the following jdbc configurations:

configuration name	driver	url	username	password
server_mode	org.h2.Driver	jdbc:h2:tcp://localhost/~/test	pif	pifp
in_memory	org.h2.Driver	jdbc:h2:mem:test	sa	sa
file	org.h2.Driver	jdbc:h2:${workingDir}/db_${idgen}	sa	sa

✅ **And** a sample database running using configuration "file"

✅ **Given** the "file" jdbc configuration has been applied

✅ **When** a query is made on table "user"

✅ **Then** the number of rows returned should be greater than 0

4. Overview

4.1. Consolidated tag views

Jdbc	1/1
Http	19/19

Sample Steps

✅ Given a passed step

🚫 And a failed step

⚙️ When a pending step

❓ But an undefined step

⚠️ Then a skipped step