

Besser Schlafen mit RestAssured

Hypermedia

Hypermedia

REST

Hypermedia

REST

- REpresentational State Transfer

Hypermedia

REST

- REpresentational State Transfer

HATEOAS

Hypermedia

REST

- REpresentational State Transfer

HATEOAS

- Hypertext As The Enginge Of Application State

Hypermedia

REST

- REpresentational State Transfer

HATEOAS

- Hypertext As The Enginge Of Application State

HTTP

Hypermedia

REST

- REpresentational State Transfer

HATEOAS

- Hypertext As The Enginge Of Application State

HTTP

- Hypertext Transfer Protocol

Hypermedia

REST

- REpresentational State Transfer

HATEOAS

- Hypertext As The Enginge Of Application State

HTTP

- Hypertext Transfer Protocol

Kennt jeder

Hypermedia

REST

- REpresentational State Transfer

HATEOAS

- Hypertext As The Enginge Of Application State

HTTP

- Hypertext Transfer Protocol

Kennt jeder

- Web-Browser

Einfach zu testen?

```
$ curl -i http://localhost:8080/rest-endpoint
```

Einfach zu testen?

```
$ curl -i http://localhost:8080/rest-endpoint
```

```
HTTP/1.1 401 Forbidden
```

Neuer Versuch

```
$ curl -i -u rhqadmin:rhqadmin http://localhost:7080/rest
```

Neuer Versuch

```
$ curl -i -u rhqadmin:rhqadmin http://localhost:7080/rest
```

```
HTTP/1.1 503 Service Unavailable  
Content-Type: */*
```

Nun mit Media Type

```
$ curl -i -u rhqadmin:rhqadmin http://localhost:7080/rest -  
HAccept:application/json
```

Nun mit Media Type

```
$ curl -i -u rhqadmin:rhqadmin http://localhost:7080/rest -  
HAccept:application/json
```

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Date: Thu, 06 Jun 2013 20:31:21 GMT
```


Nun mit Media Type

```
$ curl -i -u rhqadmin:rhqadmin http://localhost:7080/rest -  
HAccept:application/json
```

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Date: Thu, 06 Jun 2013 20:31:21 GMT
```

```
[{"platforms":{"href":"resource/platforms.json"}}, {"groups":  
{"href":"group.json"}}, {"dynaGroups":{"href":"group/definitions.json"}},  
{"alerts":{"href":"alert.json"}}, {"status":{"href":"status.json"}},  
{"favoriteResources":{"href":"user/favorites/resource.json"}}]
```

Auswertung

Ist "status.json" im Ergebnis?

```
$curl ... -o - | grep "status.json" && echo "Ist gut"
```

Besser in Java ?

```
URLConnection con = "http://...".toUrl().getConnection();
con.open()
InputStream is = con.getInputStream();
while(is) {
    readInput();
}
boolean gut = Compare(input, "status.json")
if (gut) {
    System.out.println("gut");
}
```

Das muss besser gehen ...

So?

```
given()
  .header("Accept", "application/json")
.expect()
  .statusCode(200)
  .body("", containsString("status.json"))
  .log()
  .ifError()
.when()
  .get("http://localhost:7080/rest");
```

So?

```
given()
  .header("Accept", "application/json")
.expect()
  .statusCode(200)
  .body("", containsString("status.json"))
  .log()
  .ifError()
.when()
  .get("http://localhost:7080/rest");
```

Willkommen bei RestAssured!

RestAssured

Johan Haleby

- PowerMock

<http://code.google.com/p/rest-assured/>

<https://github.com/jayway/rest-assured>

ASL 2.0

Kurzdurchlauf

Kurzdurchlauf

```
given()  
  .header("Accept", "application/json")  
  .queryParams("city", "Karlsruhe")
```

Kurzdurchlauf

```
given()  
  .header("Accept", "application/json")  
  .queryParams("city", "Karlsruhe")
```

```
.expect()  
  .statusCode(200)  
  .body("", containsString("status.json"))
```

Kurzdurchlauf

```
given()  
  .header("Accept", "application/json")  
  .queryParams("city", "Karlsruhe")
```

```
.expect()  
  .statusCode(200)  
  .body("", containsString("status.json"))
```

```
.when()  
  .get("http://localhost:7080/rest/{city}");
```

Begleitprojekt: JsonPath

```
[{"platforms":{"href":"resource/platforms.json"}}, {"groups":{"href":"group.json"}}, {"dynaGroups":{"href":"group/definitions.json"}}, {"alerts":{"href":"alert.json"}}, {"status":{"href":"status.json"}}, {"favoriteResources":{"href":"user/favorites/resource.json"}}]
```

```
.body("*.*.href", hasItem("group.json"))
```

Begleitprojekt: XmlPath

```
<data>  
  <person><name>heiko</name>...</person>  
  <person><name>Horschd</name>...</person>  
</data>
```

```
.body("data.person.name", containsString("Horschd"))
```

Matcher

Hamcrest & eigen

```
.body(..., containsString("123"))  
.header("Content-Type", is("application/json"))
```

Schauen wir doch mal Code an ...

Schauen wir doch mal Code an ...
