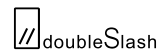




Maximilian Göke | 24.04.2023

doubleSlash

HATEOAS – Können REST APIs Geschichten erzählen?





Block 2
11:00 -
12:00

Die nachhaltige Präsentation - best practises für den bleibenden Vortrag
TD-158

Data visualization for runaways
TD-175

Big Data Technologieübersicht anhand Hortonworks Data Plattform und Streaming
TD-186





Picture: Patrick Gruban, Algerien 5 0049, CC BY-SA 2.0, cropped

Swagger Petstore 1.0.6

[Base URL: petstore.swagger.io/v2]
<https://petstore.swagger.io/v2/swagger.json>

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [#swagger](http://irc.freenode.net). For this sample, you can use the api key `special-key` to test the authorization filters.

- [Terms of service](#)
- [Contact the developer](#)
- [Apache 2.0](#)
- [Find out more about Swagger](#)

Schemes


HTTPS ▾

Authorize 


pet Everything about your Pets Find out more ^

- POST


/pet/{petId}/uploadImage uploads an image

▾ 
- POST


/pet Add a new pet to the store

▾ 
- PUT

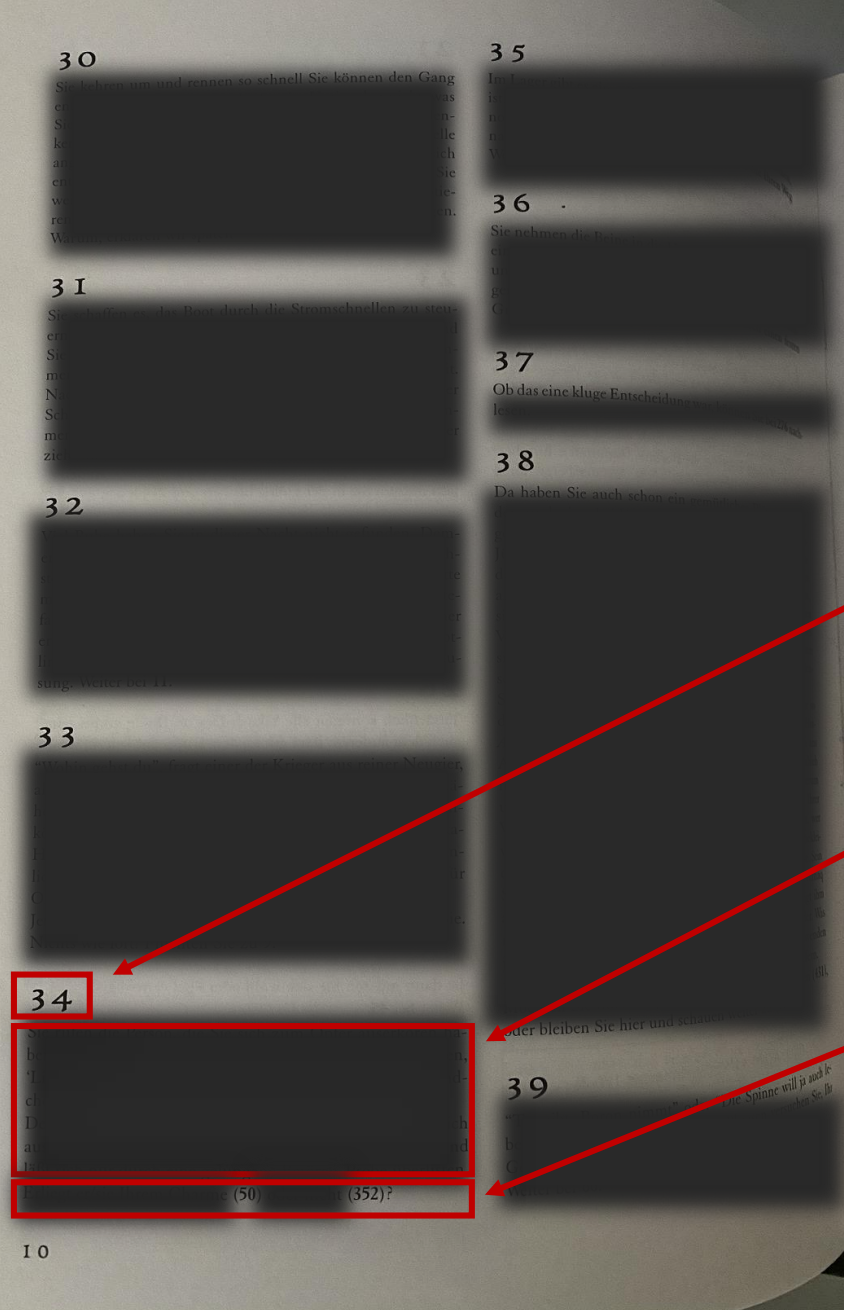
/pet Update an existing pet

▾ 
- GET

/pet/findByStatus Finds Pets by status

▾ 



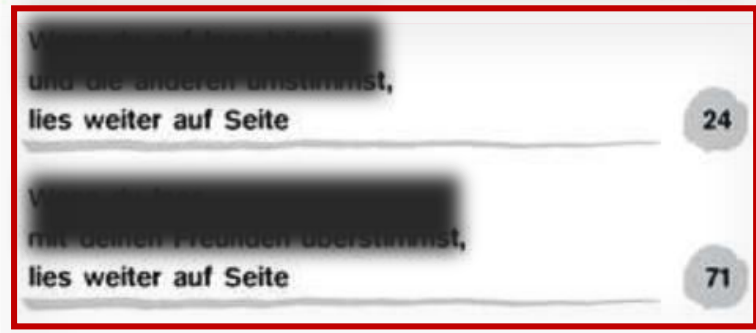
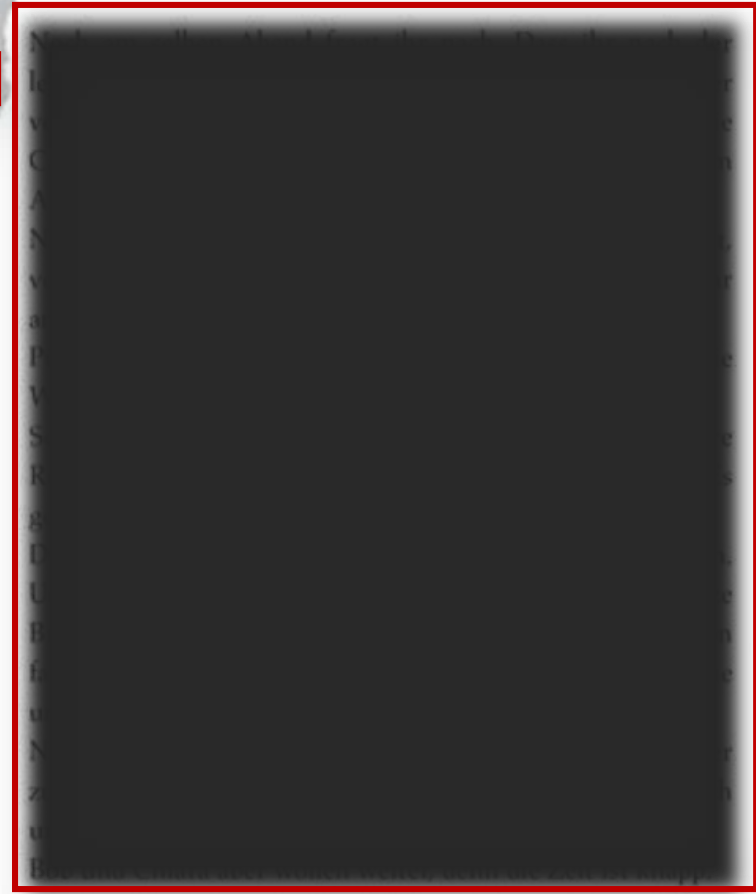


16

ID

Inhalt

Wie geht es weiter



Hypermedia As The Engine Of Application State

```
{  
  "id": "hal-cookbook",  
  "name": "HAL Cookbook,,,  
  "_links": {  
    "self": {  
      "href": "/api/book/hal-cookbook"  
    }  
  }  
}
```


GET /api/book/**hal-cookbook**

```
{  
  "id": "hal-cookbook",  
  "name": "HAL Cookbook,,,  
  "_links": {  
    "self": {  
      "href": "/api/book/hal-cookbook"  
    }  
  }  
}
```

ID

16

Inhalt

Wie geht es weiter

W...
und die anderen umstimmt,
lies weiter auf Seite 24

W...
mit deinen Freunden überstimmt,
lies weiter auf Seite 71

HATEOAS Standards

```
{
  "_links": {
    "self": {
      "href": "/api/book/hal-cookbook"
    }
  },
  "id": "hal-cookbook",
  "name": "HAL Cookbook"
}
```

HAL - Hypertext Application Language

```
{
  "collection" :
  {
    "version" : "1.0",
    "href" : "http://example.org/friends/",

    "links" : [
      { "rel" : "feed", "href" : "http://example.org/friends/rss" },
    ],

    "items" : [
      {
        "href" : "http://example.org/friends/jdoe",
        "data" : [
          { "name" : "full-name", "value" : "J. Doe", "prompt" : "Full Name" },
          { "name" : "email", "value" : "jdoe@example.org", "prompt" : "Email" }
        ],
        "links" : [
          { "rel" : "blog", "href" : "http://examples.org/blogs/jdoe", "prompt" :
"Blog" }
        ]
      }
    ]
  }
}
```

Collection+JSON

ZORK I: The Great Underground Empire
Copyright (c) 1981, 1982, 1983 Infocom, Inc. All rights reserved.
ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726

West of House

You are standing in an open field west of a white house, with a boarded
front door.

There is a small mailbox here.

>

ZORK I: The Great Underground Empire
Copyright (c) 1981, 1982, 1983 Infocom, Inc. All rights reserved.
ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726

West of House
You are standing in an open field west of a white house, with a boarded
front door.
There is a small mailbox here.
>

GET /house/west

```
{
  "data": {
    "text": "West of House. You are standing
in an open field west of a white house, with a
boarded front door. There is a small mailbox
here."
  },
  "_links": {
    "self": {
      "href": "/house/west"
    },
    "mailbox": {
      "href": "/house/west/mailbox"
    },
    ...
  }
}
```


ZORK I: The Great Underground Empire
Copyright (c) 1981, 1982, 1983 Infocom, Inc. All rights reserved.
ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726

West of House

You are standing in an open field west of a white house, with a boarded front door.

There is a small mailbox here.

>look at mailbox

The small mailbox is closed.

>

POST /house/west/mailbox

```
{
  "action": "look"
}

{
  "data": {
    "text": "The small mailbox is closed."
  },
  "_links": {
    "self": {
      "href": "/house/west/mailbox"
    },
    "look around": {
      "href": "/house/west"
    },
    "mailbox": {
      "href": "/house/west/mailbox"
    }
  }
}
```

ZORK I: The Great Underground Empire
Copyright (c) 1981, 1982, 1983 Infocom, Inc. All rights reserved.
ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726

West of House

You are standing in an open field west of a white house, with a boarded front door.

There is a small mailbox here.

>look at mailbox

The small mailbox is closed.

>open mailbox

Opening the small mailbox reveals a leaflet.

>_

POST /house/west/mailbox

```
{
  "action": "open"
}

{
  "data": {
    "text": "The small mailbox is closed."
  },
  "_links": {
    "self": {
      "href": "/house/west/mailbox"
    },
    "look around": {
      "href": "/house/west"
    },
    "mailbox": {
      "href": "/house/west/mailbox"
    }
  }
}
```


Zork REST-API Source Code

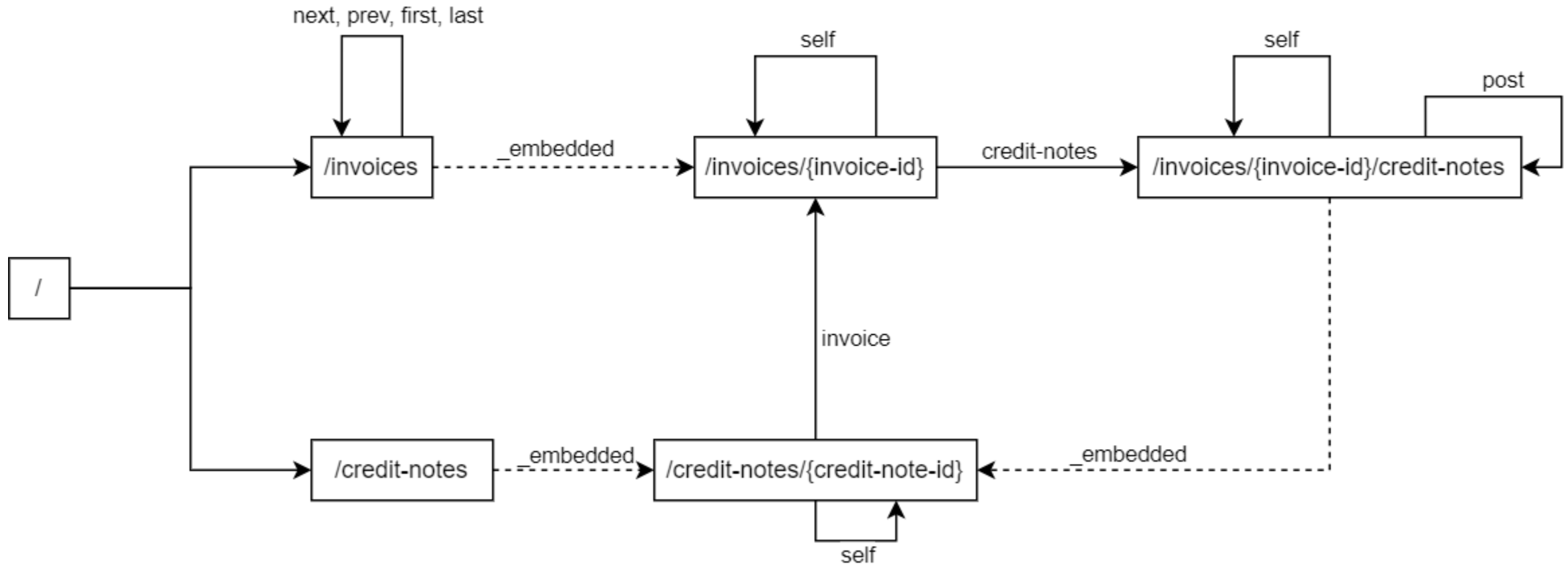
```
5 usages
7 public class HouseEntity extends RepresentationModel<HouseEntity> {
8
9     3 usages
10     ResponseData data;
11
12     no usages
13     public HouseEntity() {}
14
15     1 usage
16     public HouseEntity(final ResponseData data) { this.data = data; }
17
18     no usages
19     public ResponseData getData() { return data; }
20
21
22     no usages
23     public void setData(final ResponseData data) { this.data = data; }
24
25
26 }
```

Zork REST-API Source Code

4 usages

```
43 @GetMapping("/house/west")
44 public ResponseEntity<HouseEntity> westSideOfHouse() {
45     final HouseEntity houseEntity = new HouseEntity(new ResponseData(
46         text: "West of House. You are standing in an open field west of a white house, with a
            boarded front door. There is a small mailbox here.");
47
48     houseEntity.add(linkTo(methodOn(ZorkController.class).westSideOfHouse()).withSelfRel());
49     houseEntity.add(linkTo(methodOn(ZorkController.class).lookAtMailbox()).withRel("look at mailbox"));
50     houseEntity.add(linkTo(methodOn(ZorkController.class).performActionOnMailbox(mailboxAction: null))
        .withRel("mailbox action"));
51     return ResponseEntity.status(HttpStatus.OK).body(houseEntity);
52 }
53
```


Invoice API - HATEOAS



Edit Headers

http://localhost:8080/api/messages

Go!

JSON Properties

```
{
  "page": {
    "size": 20,
    "totalElements": 4,
    "totalPages": 1,
    "number": 0
  }
}
```

Links

Relation	Name	Title	HTTP Request	Doc
self			<div><+>>✖</div>	
profile			<div><+>>✖</div>	
curies [0]	chatty		<div><+>>✖</div>	

Embedded Resources

chatty:messages	
chatty:messages [0]	
chatty:messages [1]	
chatty:messages [2]	

Response Status

200 (OK)

Response Headers

content-length	2503
content-type	application/hal+json
date	Sun, 01 Jan 2023 14:21:22 GMT
etag	"0eed8e2df4ab1bb01e39dea0d99e7f933"

Response Body

```
{
  "_embedded": {
    "chatty:messages": [
      {
        "text": "hello all!",
        "timeStamp": "2023-01-01T14:07:25.942238Z",
        "_links": {
          "self": {
            "href": "http://localhost:8080/api/messages/1"
          },
          "chatty:chatMessage": {
            "href": "http://localhost:8080/api/messages/1{?projection}",
            "templated": true
          },
          "author": {
            "href": "http://localhost:8080/api/messages/1/author{?projection}",
            "templated": true
          }
        }
      }
    ]
  }
}
```

Swagger Petstore 1.0.6

[Base URL: petstore.swagger.io/v2]
<https://petstore.swagger.io/v2/swagger.json>

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [#swagger](http://irc.freenode.net). For this sample, you can use the api key `special-key` to test the authorization filters.

[Terms of service](#)

[Contact the developer](#)

[Apache 2.0](#)

[Find out more about Swagger](#)

Schemes

HTTPS

Authorize



pet Everything about your Pets

[Find out more](#)



POST **/pet/{petId}/uploadImage** uploads an image



POST **/pet** Add a new pet to the store



PUT **/pet** Update an existing pet



GET **/pet/findByStatus** Finds Pets by status



GET **/pet/findByTags** Finds Pets by tags



Table of Contents

- Request structure
- Request parameters
- Response fields
- Response links
- Example response
- CURL request

Invoice Manager

This endpoint returns all invoices. There is no way to change the order of the elements. It uses pagination by default. The default values are `1` for `offset` and `5` for `limit`. This gives you the first page which contains 5 invoices.

To navigate through the invoices you can use the provided HATEOAS links.

Request structure

```
GET /invoices HTTP/1.1
Host: localhost
```

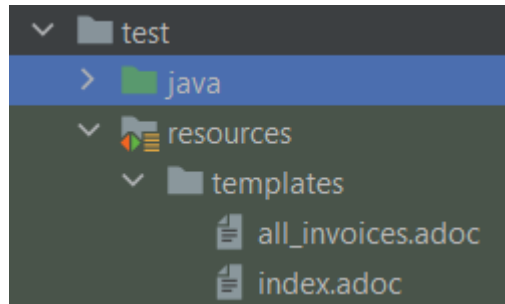
```
void should_return_invoices_with_links() throws Exception {
```

```
    this.mockMvc.perform(get( urlTemplate: "/invoices")
        .queryParam( name: "offset", ...values: "2")
        .queryParam( name: "limit", ...values: "1"))
        .andDo(print())
        .andExpect(status().isOk())
        .andDo(documentationHandler.document(
            links(
                hallinks(),
                linkWithRel("self").description("Link to this resource"),
                linkWithRel("next").description("Link to next page"),
                linkWithRel("previous").description("Link to previous page")
            ),
            queryParameters(
                parameterWithName("offset").description("Indicates the page. Starts with 1").optional(),
                parameterWithName("limit").description("Limits the amount of invoices per page").optional()
            ),
            responseFields(
                fieldWithPath("invoices").description("Array that contains all invoices"),
                fieldWithPath("invoices[].invoiceId").description("ID of an invoice"),
                fieldWithPath("invoices[].orderId").description("ID of the order that caused this invoice"),
                fieldWithPath("invoices[].totalNetAmount").description("Total net amount of invoice"),
                fieldWithPath("invoices[].totalTaxAmount").description("Total tax amount of invoice"),
                fieldWithPath("invoices[].totalGrossAmount").description("Total gross amount of invoice"),
                subsectionWithPath("invoices[].customer").description("Contains information about customer"),
                subsectionWithPath("invoices[]._links").description("Link to resources for specific invoice"),
                subsectionWithPath("_links").description("Link to resources")
            )
        ));
}
```

```

  ▾ target
    > classes
    > generated-docs
    ▾ generated-snippets
      > should_return_invoice_that_can_be_refunded
      > should_return_invoice_that_is_already_refunded
      > should_return_invoices
      ▾ should_return_invoices_with_links
        curl-request.adoc
        http-request.adoc
        http-response.adoc
        httpie-request.adoc
        links.adoc
        query-parameters.adoc
        request-body.adoc
        response-body.adoc
        response-fields.adoc

```



This endpoint returns all invoices. There is no way to change the order of the elements.
It uses pagination by default. The default values are `1` for `offset` and `5` for `limit`. This gives you the first page

To navigate through the invoices you can use the provided HATEOAS links.

=== Request structure

include::{snippets}/should_return_invoices/http-request.adoc[]

Was sagt ihr?
Können REST APIs Geschichten erzählen?