

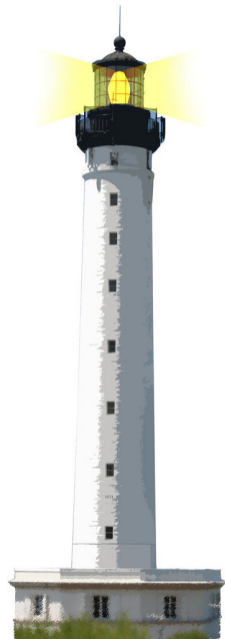
Seaside: A Glance at MetaData and REST

Damien Cassou, Stéphane Ducasse and Luc Fabresse

W4S10



<http://www.pharo.org>

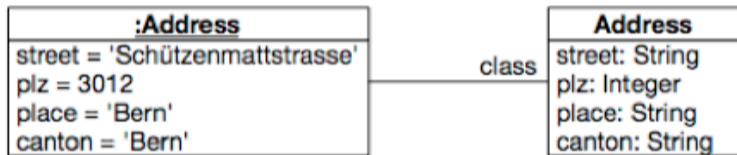


Magritte: Taking Advantages of MetaData

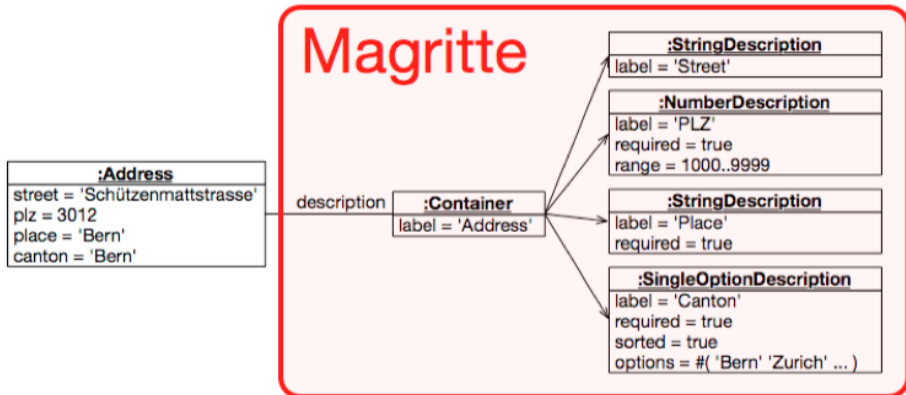
- Describe once
- Generate many times
- No more manually created forms
- Seaside component automatic generation



Address



Address Description



Describe

```
Address >> descriptionStreet  
  ^ MAStringDescription new  
    autoAccessor: #street;  
    label: 'Street';  
    priority: 100
```

```
Address >> descriptionPlz  
  ^ MANumberDescription new  
    autoAccessor: #plz;  
    priority: 200; label: 'PLZ';  
    beRequired;  
    min: 1000;  
    max: 9999
```



Interpret

```
anAddress description do: [ :description |
```

```
  Transcript
```

```
    show: description label; show: ':'; tab;
```

```
    show: (description toString: (anAddress readUsing:  
      description));
```

```
  cr ]
```

```
Street: Schutzenmattstrasse
```

```
PLZ: 3012
```

```
Place: Bern
```

```
Canton: Bern
```



Generate Forms

self call: (anAddress asComponent addValidatedForm; yourself)

Street:

PLZ: *

Place: *

Canton: ▾ *

Generate Automatic Reports

The screenshot displays the QUVE software interface, which is used for generating automatic reports. The interface is divided into several sections:

- Top Left:** A line chart showing data trends over time, with a legend and axes.
- Top Center:** A navigation menu with options like 'Home', 'Reports', 'Tools', and 'Help'. Below it, there are search and filter options.
- Top Right:** A 'Reports Summary' section with a 'Generate' button and a list of report categories.
- Middle Left:** A 'Data Summary' table with columns for 'Date', 'Value', and 'Unit'. It lists various financial metrics.
- Middle Center:** A large table with multiple columns, likely representing a detailed financial statement or portfolio overview. It includes columns for 'Date', 'Value', 'Unit', and 'Status'.
- Middle Right:** A 'Reports Summary' section with a 'Generate' button and a list of report categories.
- Bottom Left:** A 'Data Summary' table with columns for 'Date', 'Value', and 'Unit'. It lists various financial metrics.
- Bottom Center:** A 'Pie Chart' section showing data distribution across different categories. It includes a legend and a 'Generate' button.
- Bottom Right:** A 'Data Summary' table with columns for 'Date', 'Value', and 'Unit'. It lists various financial metrics.

The interface is designed to be user-friendly and provides a comprehensive overview of financial data and reports.

Seaside-REST

- Smooth and easy integration
- Annotation of domain objects
- Natural conversion

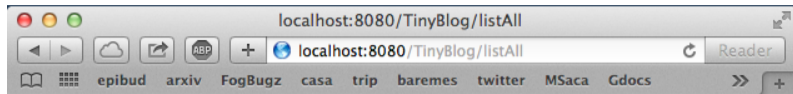


Filter Definition

```
WARestfulFilter subclass: #TBRestfulFilter
instanceVariableNames: ''
classVariableNames: ''
package: 'TinyBlog-REST'
```

```
TBApplicationRootComponent class >> initialize
"self initialize"
| app |
app := WAAdmin register: self asApplicationAt: 'TinyBlog'.
app
  addLibrary: JQDeploymentLibrary;
  addLibrary: JQUIDeploymentLibrary;
  addLibrary: TBSDeploymentLibrary.
app addFilter: TBRestfulFilter new.
```

Getting All Blogs



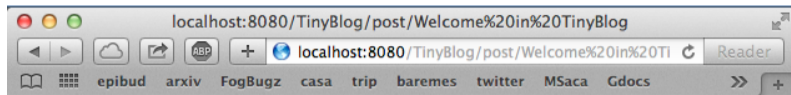
```
[{"title": "Welcome in TinyBlog", "category": "TinyBlog", "date": new Date(2015, 11, 17), "text": "TinyBlog is a small blog engine made with Pharo."}, {"title": "Report Pharo Sprint", "category": "Pharo", "date": new Date(2015, 11, 17), "text": "Friday, June 12 there was a Pharo sprint / Moose dojo. It was a nice event with more than 15 motivated sprinters. With the help of candies, cakes and chocolate, huge work has been done"}, {"title": "Brick on top of Bloc - Preview", "category": "Pharo", "date": new Date(2015, 11, 17), "text": "We are happy to announce the first preview version of Brick, a new widget set created from scratch on top of Bloc. Brick is being developed primarily by Alex Syrel (together with Alain Plantec, Andrei Chis and myself), and the work is sponsored by ESUG. \r\t\tBrick is part of the Glamorous Toolkit effort and will provide the basis for the new versions of the development tools."}, {"title": "The sad story of unclassified blog posts", "category": "Unclassified", "date": new Date(2015, 11, 17), "text": "So sad that I can read this."}, {"title": "Working with Pharo on the Raspberry Pi", "category": "Pharo", "date": new Date(2015, 11, 17), "text": "Hardware is getting cheaper and many new small devices like the famous Raspberry Pi provide new computation power that was one once only available on regular desktop computers."}]
```

Getting All blogs

```
TBRestfulFilter >> listAll
<get>
<produces: 'text/json'>

^ String streamContents: [ :astream |
  TBlog current allBlogPosts
  do: [ :each | each javascriptOn: stream ]
  separatedBy: [ stream << ',' ] ]
```

Finding a Blog



```
{"title": "Welcome in TinyBlog", "date": new  
Date(2015, 11, 17), "category": "TinyBlog"}
```

Finding a Blog

```
TBRestfulFilter >> post: title
<get>
<path: 'post/{title}'>
<produces: 'text/json'>
| post |
post := TBBlog current postWithTitle: title.
post ifNil: [ ^ self notFound ].
^ String streamContents: [ :astream |
    post javascriptOn: astream ]
```



With Search

```
TBRestfulFilter >> search: title
<get>
<path: 'search?title={title}'>
<produces: 'text/json'>
| post |
post := TBBlog current postWithTitle: title.
post ifNil: [ ^ self notFound ].
^ String streamContents: [ :astream |
  post javascriptOn: astream attributes: #(title category date
  text) ]
```

```
http://localhost:8080/TinyBlog/search?title=Welcome%20
in%20TinyBlog
```

Conclusion

- Seaside supports the other REST API (Delete, Post, Get,...)
- Seaside coupled with Magritte offers a really powerful combination
- REST is nicely integrated in Seaside-REST
- Have a look at Teapot for another REST solution



A course by



and



in collaboration with



Inria 2020

Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France

<https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>