Seaside: A Glance at MetaData and REST

Damien Cassou, Stéphane Ducasse and Luc Fabresse

http://www.pharo.org
Magritte: Taking Advantages of MetaData

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

:Address
- street = 'Schützenmattstrasse'
- plz = 3012
- place = 'Bern'
- canton = 'Bern'

:Container
- label = 'Address'

:Description
- label = 'Street'
- required = true
- range = 1000..9999

:Description
- label = 'Place'
- required = true

:SingleOptionDescription
- label = 'Canton'
- required = true
- sorted = true
- options = #('Bern' 'Zurich' ...)
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
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
self call: (anAddress asComponent addValidatedForm; yourself)
Generate Automatic Reports
Seaside-REST

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

WARestfulFilter subclass: #TBRestfulFilter
  instanceVariableNames: ''
  classVariableNames: ''
  package: 'TinyBlog−REST'

TBAplicationRootComponent class >> initialize
  "self initialize"
  | app |
  app := WAAadmin 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\n\rBrick 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>
<p produces: 'text/json'>

\[
\text{^ String streamContents: [ :astream | }
\text{TBBlog current allBlogPosts }
\text{do: [ :each | each javascriptOn: stream ]}
\text{separatedBy: [ stream << ',' ] ]}
\]

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) ]
```

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

Inria

and

UTOP

in collaboration with

Unisciel

Fondation unit

Université de Lille

INSTITUT Mines-Télécom

MINES Douai

Inria 2016

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/