

# A Simple HTTP Application

As a pretext to revisit Pharo Syntax

Damien Cassou, Stéphane Ducasse and Luc Fabresse

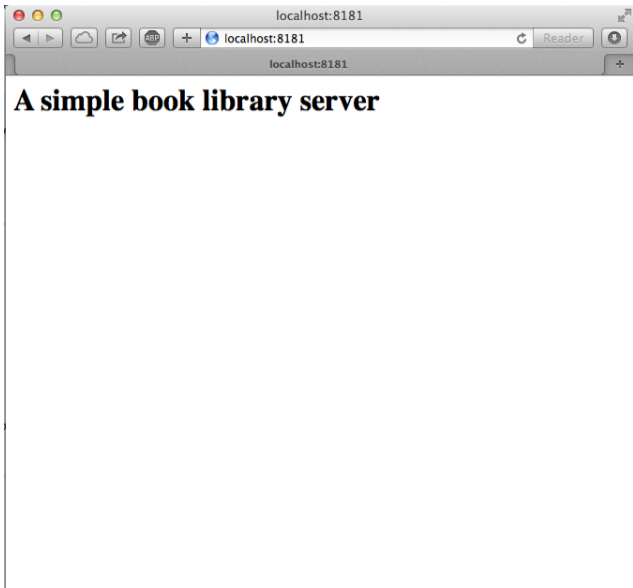
W3S04



<http://www.pharo.org>



# A Tiny Book Server

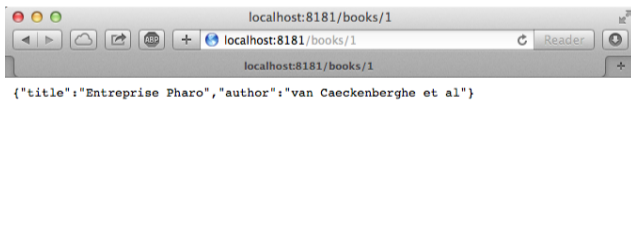


# Getting a Book

ZnClient new

```
url: 'http://localhost:8181/books/1';  
get
```

- Class name starts with an uppercase
- new is a message sent to the class ZnClient
- url: and get are both sent to the instance of ZnClient



# Adding a Book

```
ZnClient new  
url: 'http://localhost:8181/books/1';  
formAt: 'author' put: 'van Caekenberghe et al';  
formAt: 'title' put: 'Entreprise Pharo';  
post
```

formAt: 'author' put: 'van Caekenberghe et al'  
is equivalent to  
formAtput('author' , 'van Caekenberghe et al')



# The Full Server

```
books := Dictionary new.  
teapot := Teapot configure: {  
  #defaultOutput -> #json. #port -> 8181 }.  
teapot  
GET: '/' -> '<h1>A simple book server</h1>'; output: #html;  
GET: '/books' -> books;  
GET: '/books/<id:lsInteger>'  
  -> [ :request | books at: (request at: #id) asString];  
POST: '/books/<id>'  
  -> [ :request || book |  
    book := { 'author' -> (request at: #author).  
      'title' -> (request at: #title) } asDictionary.  
    books at: (request at: #id) put: book ];  
start.
```

# Configuring a Server

```
| books teapot |  
books := Dictionary new.  
teapot := Teapot configure: { #defaultOutput -> #json. #port  
    -> 8181 . #debugMode -> true }.
```

- || delimits local variable definition
- := assignment
- #port is a symbol (aka unique string)
- configure: is a message sent to the class Teapot
- In configure:, : means that the message is expecting an argument
- { . . } is an array of three elements
- -> creates a key-value pair



# Defining The Server Routes

```
teapot
GET: '/'
  -> '<h1>A simple book server</h1>'; output: #html;
GET: '/books'
  -> books;
GET: '/books/<id:lsInteger>'
  -> [ :request | books at: (request at: #id) asString];
POST: '/books/<id>'
  -> [ :request || book |
      book := { 'author' -> (request at: #author) .
                'title' -> (request at: #title) } asDictionary.
      books at: (request at: #id) put: book ];
start.
```



# More About Syntax

Processing URIs such as: `http://localhost:8181/books/1`

```
teapot
```

```
GET: '/books/<id:Integer>'
```

```
→ [ :request | books at: (request at: #id) asString];
```

- [ :request | books at: (request at: #id) asString ] is a block
- acts as an anonymous method
  - :request is an argument
  - at: is a message accepting one argument





# Conclusion

- Teapot is a cool package  
<http://smalltalkhub.com/#!/~zeroflag/Teapot>
- A web server in one page
- Based on Zinc a really strong and well designed HTTP client/server



A course by



and



in collaboration with



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/>