A Simple HTTP Application
As a pretext to revisit Pharo Syntax
Damien Cassou, Stéphane Ducasse and Luc Fabresse

http://www.pharo.org
A Tiny Book Server

A simple book library server
Getting a Book

```javascript
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

```plaintext
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
formAt:put('author', 'van Caekenberghe et al')
```
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:IsInteger>'
    -> [: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 := Dictionary new.
teapot := Teapot configure: { #defaultOutput -> #json. #port -> 8181. #debugMode -> true }.

- `| books teapot |` delimits local variable definition
- `books := Dictionary new.` := assignment
- `#port` is a symbol (aka unique string)
- `configure: { #defaultOutput -> #json. #port -> 8181. #debugMode -> true }` is a message sent to the class Teapot
- `configure:` 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:IsInteger>'
   --> [: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

```
http://localhost:8181/books/1
```

teatop

GET: '/books/<id:IsInteger>'

→ [: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

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/