

Web 2.0

A short guide to fall into the client-side



Esteban Lorenzano

(by now, you already listen too much about that name)



because not everything is Seaside



Context

- ✦ Well, I suppose you all know, but web application development moved to a more responsive UI.
- ✦ But computations still happen in server side, so we need to find a bridge.



Reef

- ✦ A Seaside extension to manage AJAX/Javascript
- ✦ Exists since years, but not publicised enough
- ✦ Evolved according to needs (my needs)



Reef design

- ✧ Object model (no DSL)
- ✧ All interaction is a Block (but there are different kind of interactions)
 - ✧ Client side computation
 - ✧ AJAX request (several types)
 - ✧ Full page request



Reef components

- ✦ Many Bootstrap components wrapped
- ✦ Some others in the form of Plugins
- ✦ Magritte-Reef



Status

- ✦ Works fine :)
- ✦ Needs more people using it
- ✦ Needs more people caring about it



Install Reef

```
Gofer it  
  smalltalkhubUser: 'estebanlm' project: 'Reef';  
  configuration;  
  loadStable.
```



Tide

- ✦ Amber-to-Pharo communication bridge
- ✦ Keeps that “Seaside feeling”
 - ✦ Provides a way to map objects from the server
 - ✦ Provides a solution for managing async calls



Tide design

- Client side
 - Proxies
 - Futures (kinda)
- Server side
 - Objects are exposed as you need, starting by a root
 - Pragmas to “mark” exposition: <action>, <state>
 - Presenters for: Literals, collections, models



Status

- ✦ Works fine :)
- ✦ Needs more people using it
- ✦ Needs more people caring about it



Install Tide

Pre-requisites: node, npm, bower, pharo

```
$ git clone git@github.com:tide-framework/tide.git
$ cd tide && bower install && cd ..
$ pharo Pharo.image eval -save “
    Gofer it
      url: ‘filetree://./tide’;
      package: ‘BaselineOfTide’;
      load.
    BaselineOfTide load”
```

TDServer startOn: 7777.



There is no silver bullet!

(This will never be stressed enough)



Choose the one you need

