Seaside Hotwired

Johan Brichau
johan@yesplan.be
• Server-side web application framework
• Page-based navigation
• jQuery-Javascript integration for “SPA” behaviour

https://github.com/SeasideSt/Seaside
https://seaside.st

• Client-side framework to augment server-side rendered pages
• Implement “SPA” behaviour without JS*
• Origins in Ruby-on-Rails but server-framework agnostic
• Turbo & Stimulus

https://hotwired.dev/
• Each action triggers a full page rendering
• Each page is a continuation in Seaside
• Server-side implementation only
jQuery for seaside

todos

- What needs to be done?
  - Task 1
  - Task 7
  - Task 8
  - Task 9
  - Task 10
  - new todo

WATodo

- WAItemEditor
- WATodoltem
- WATodoltem
- WATodoltem
Each action triggers a XMLHttpRequest (JQuery AJAX)

One page, one continuation in Seaside

Server side (Smalltalk) and client-side (Javascript) implementation
Turbo Frames are independent pieces of a web page that can be appended, prepended, replaced, or removed without a complete page refresh.
• Call/answer replaces the receiving component with the called component

• Action callback changes state of one or more components

• Action link inside a turboframe replaces that frame

• Turbostream of actions:
  • Replace / Update
  • Remove
  • Append / Prepend
  • Before / After
In summary:

- Divide a page in Turbo Frames by decorating existing components
- Adapt Links & Forms to do partial page updates with `turboCall: & turboStreamCallback:`

Advantages:

- No jQuery/Javascript (or less :-))
- Optimisation of Seaside rendering: partial rendering on the server side
- Remove unneeded callbacks from session state [WIP]
• Currently in alpha preview mode
  • API is not settled
    • E.g. WATurboFrame decoration vs WATurboFrameComponent subclass?
  • Limitation: Turbo links with ‘replace’ rather than ‘advance’ action
  • Turbo Streams via Websockets [WIP]
  • Error handling needs improvements
Thank you for listening!

Seaside Hotwired

Johan Brichau
johan@yesplan.be

Yesplan