



ESUG 2023, Lyon (France)



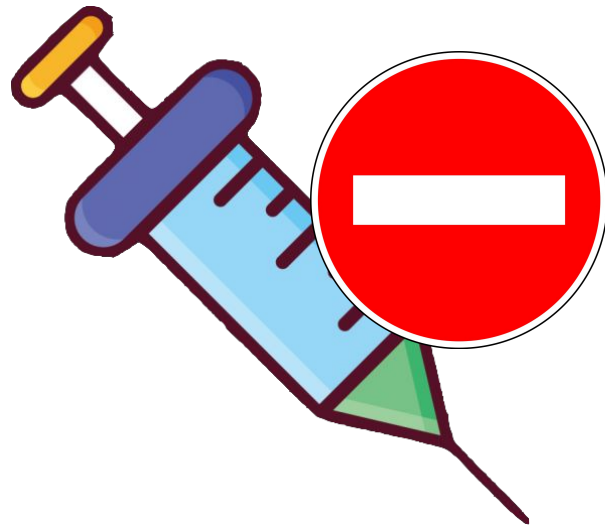
Hijack the JavaScript Ecosystem

Noury Bouraqadi & Dave Mason



I am an **Addict!**

Not That Kind of Addict!



I am a Smalltalk Addict!

ME AT HOME



I am a Smalltalk Addict!

ME AT HOME



ME AT WORK



I am a Smalltalk Addict!

ME AT HOME



ME AT WORK



ME ON VACATION



I am **Lazy!**

I am a Lazy Developer!



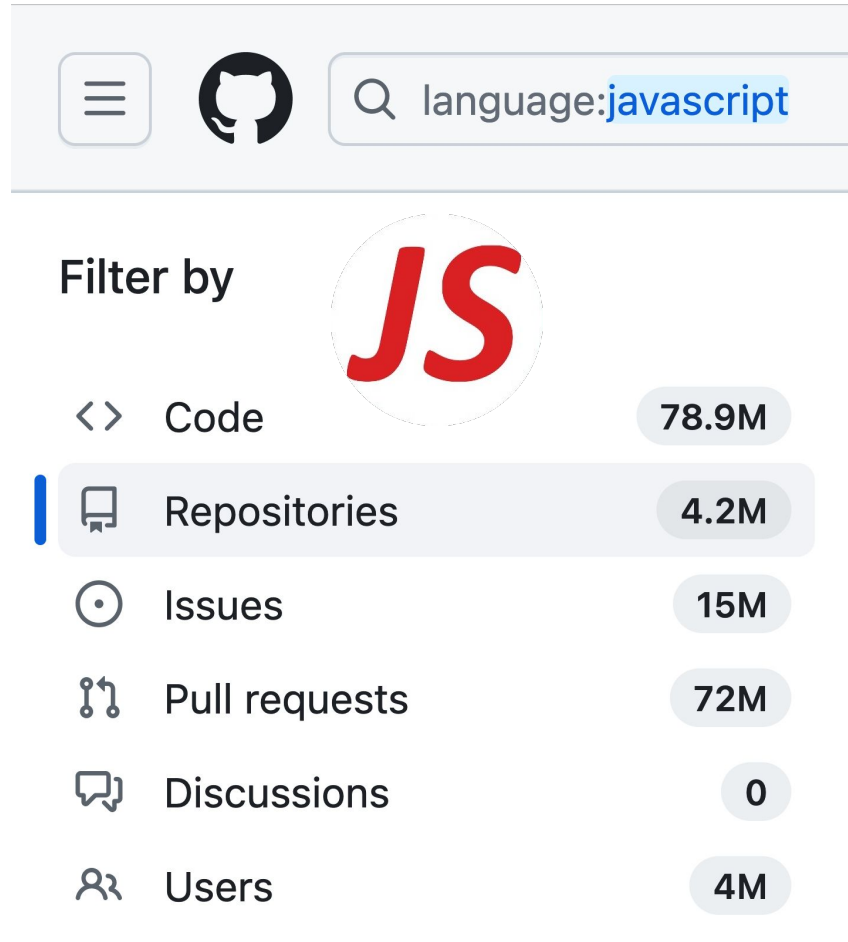
pharo-open-documentation /
awesome-pharo

A categorized community-driven collection of awesome Pharo libraries, tools, frameworks and software.








Thank You
Cyril Ferlicot

I am a Lazy Developer!



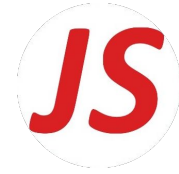
The image shows a screenshot of the GitHub search interface. At the top, there is a search bar with the text "language:javascript" and a magnifying glass icon. To the left of the search bar is the GitHub logo and a menu icon. Below the search bar, the text "Filter by" is followed by a large red "JS" logo. A list of filter categories is shown, each with an icon and a count in a rounded rectangle. The "Repositories" category is highlighted with a blue vertical bar on the left.

| | | |
|---|---------------|-------|
| <> | Code | 78.9M |
|  | Repositories | 4.2M |
|  | Issues | 15M |
|  | Pull requests | 72M |
|  | Discussions | 0 |
|  | Users | 4M |

Addict + Lazy = *Phar***JS**



+



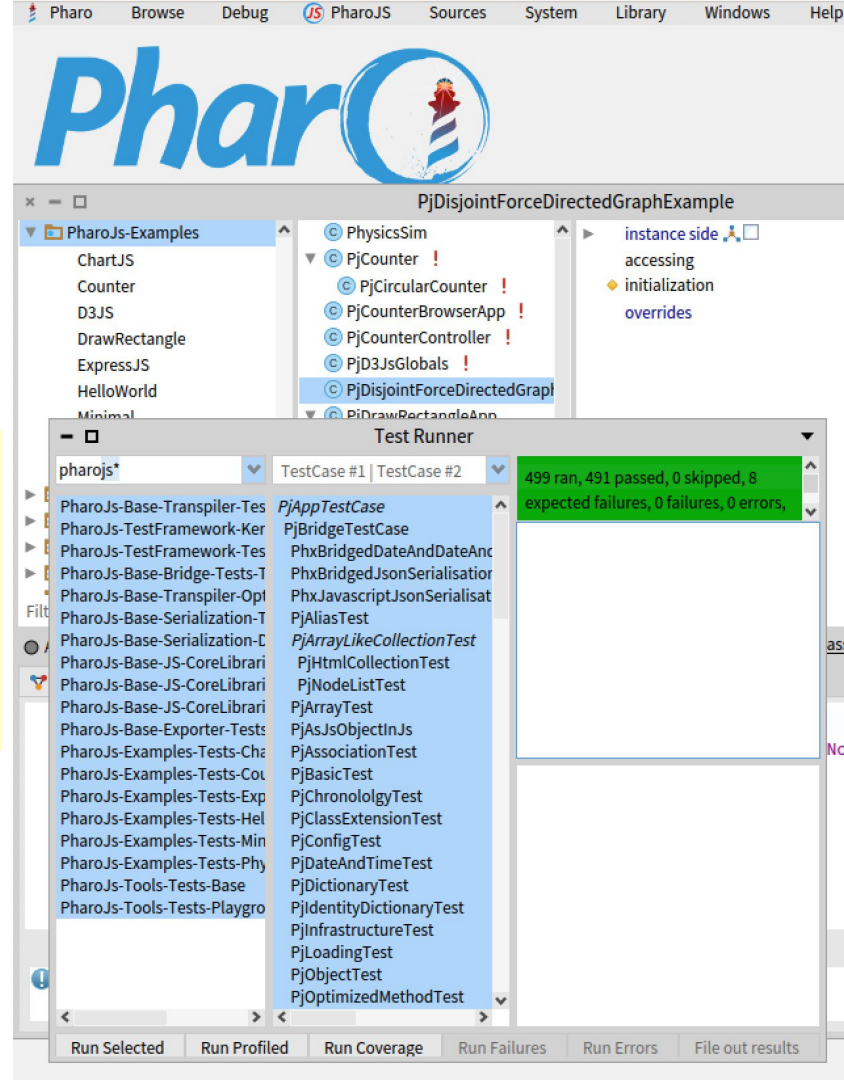
Development

**Libraries
&
Run-Time**



- **Transpiler:** Converts Pharo Code to JavaScript
- **Framework:** Develop JS applications in Pharo
- **Libraries:** Extend JS Objects with Pharo's Behavior
- **Tools:** Playground + Inspector for JS Objects
- **Test Framework:** Test JS Code

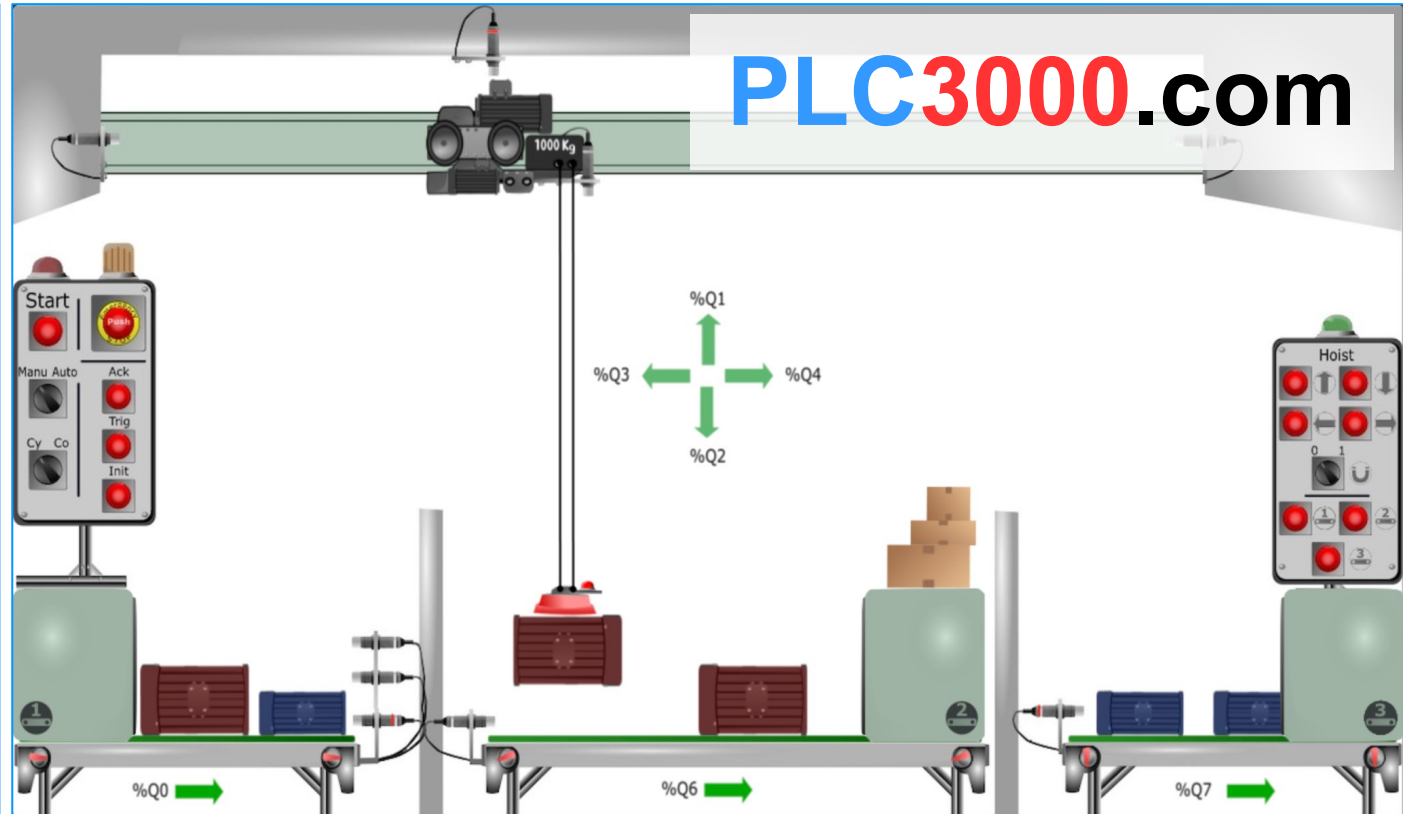
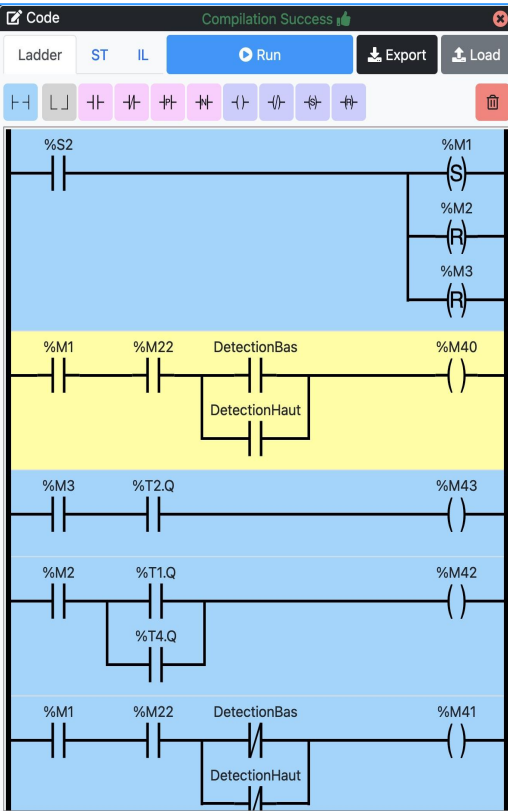
Mobile Apps with *Phar*JS



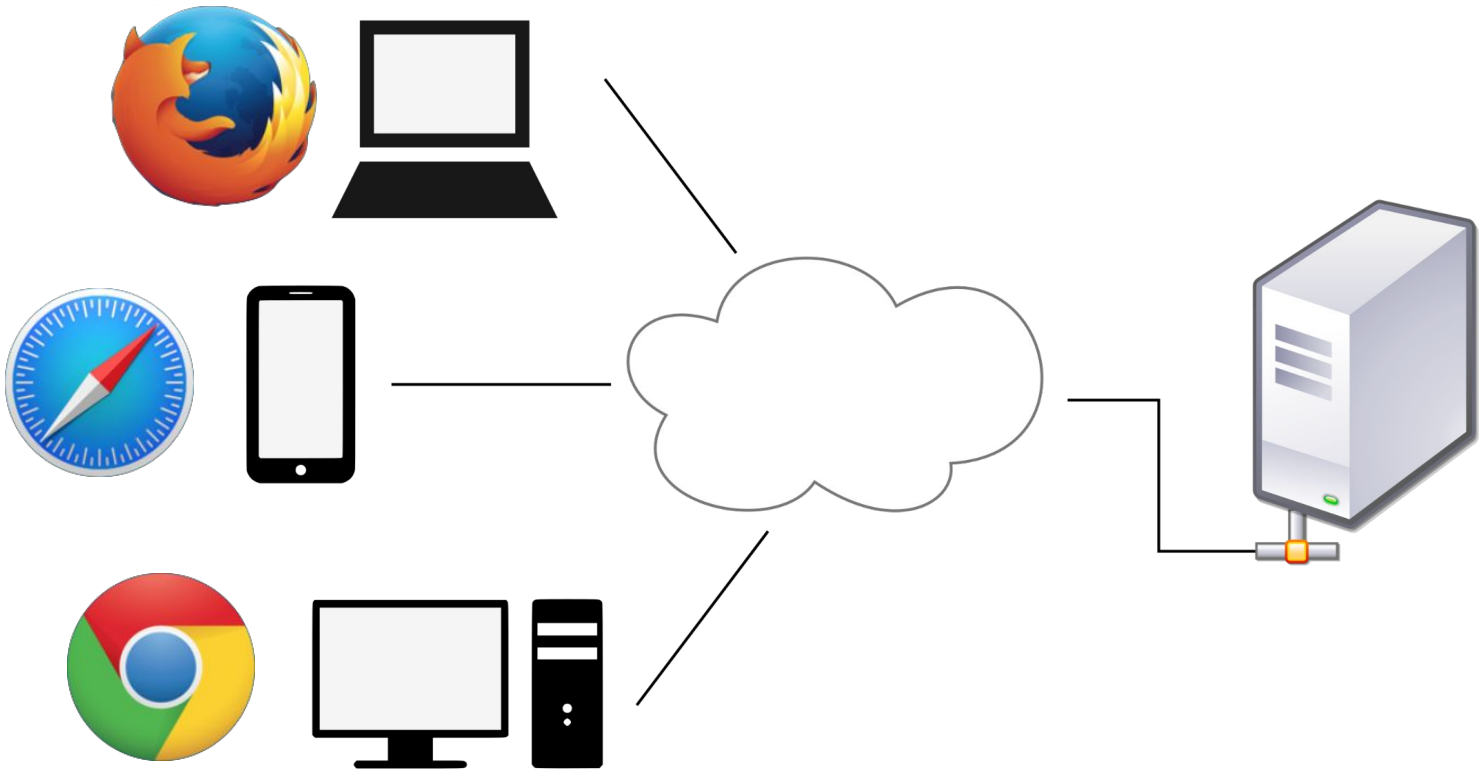
Mobile Apps with *PharJS*



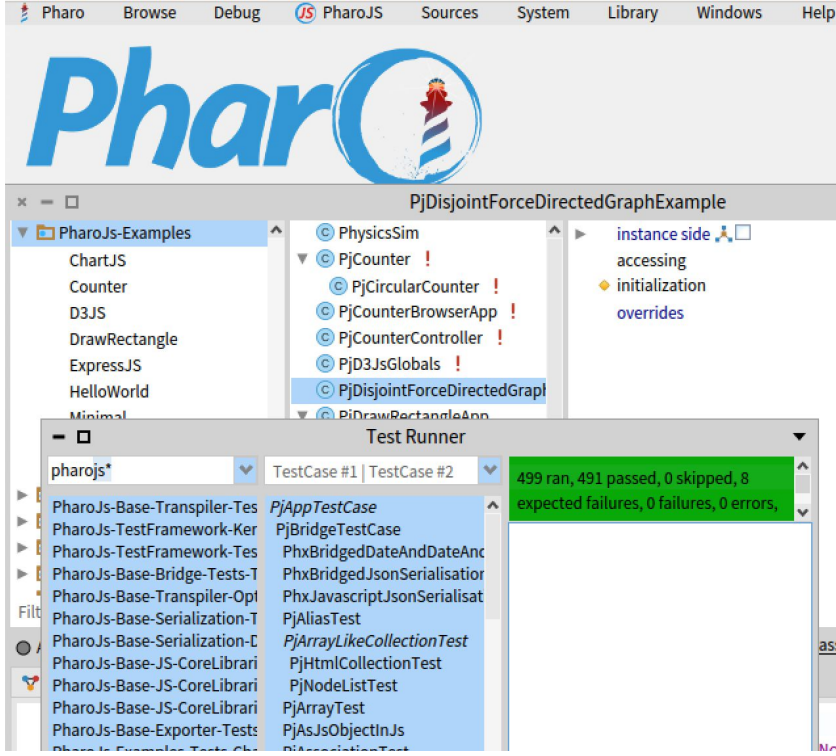
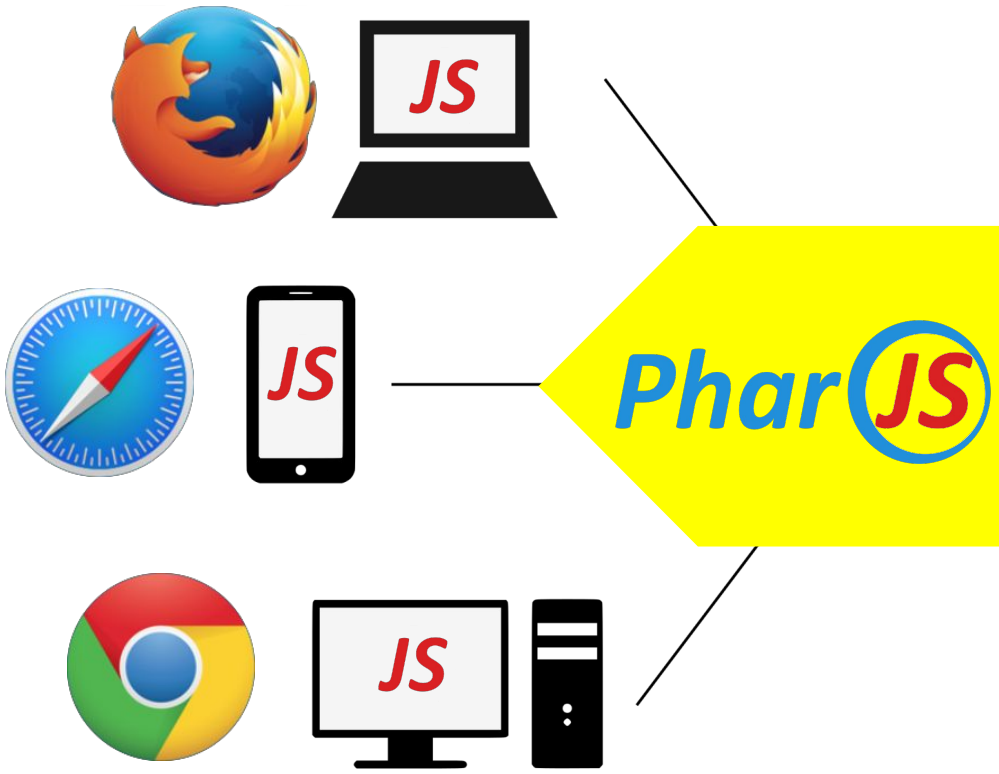
Web Apps with *PharJS*



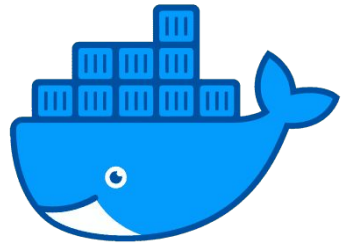
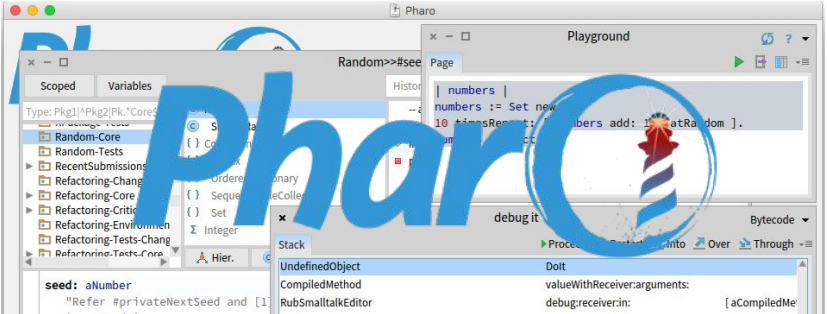
PLC3000.com Architecture



PLC3000.com Clients



PLC3000.com Server

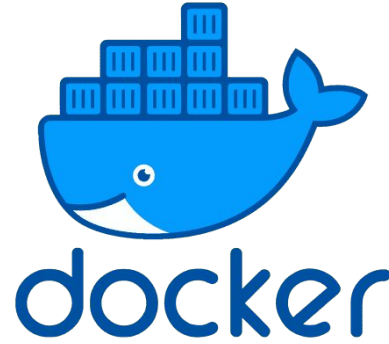
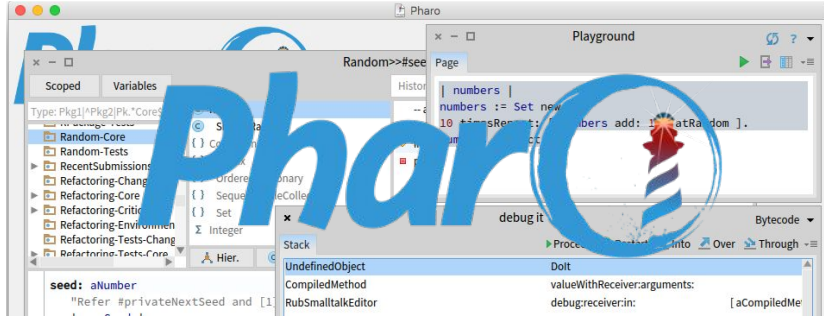


docker



Google Cloud



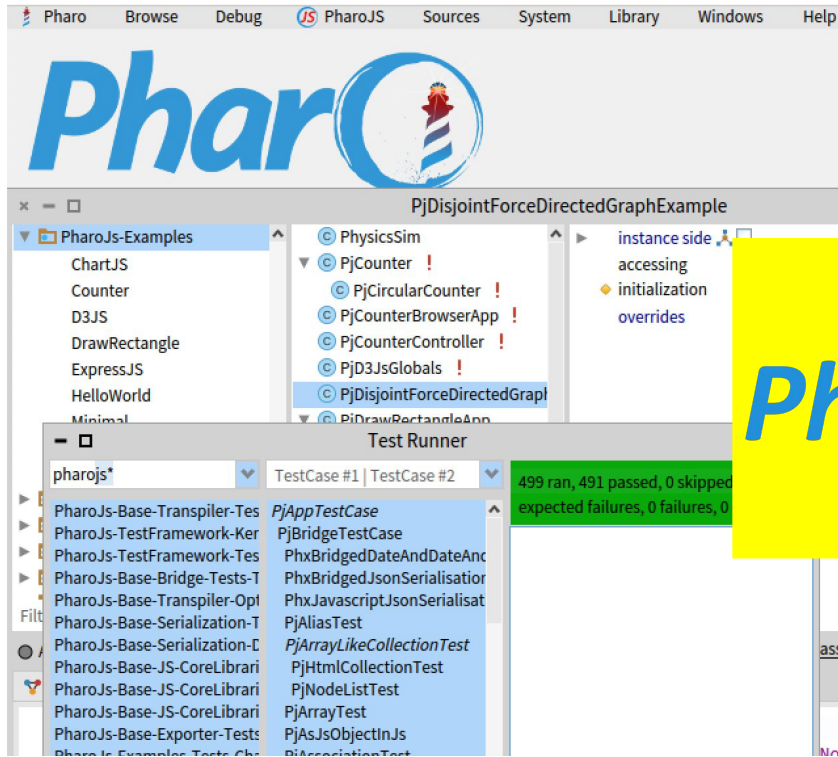


Thank you
Buenos Aires
Smalltalk

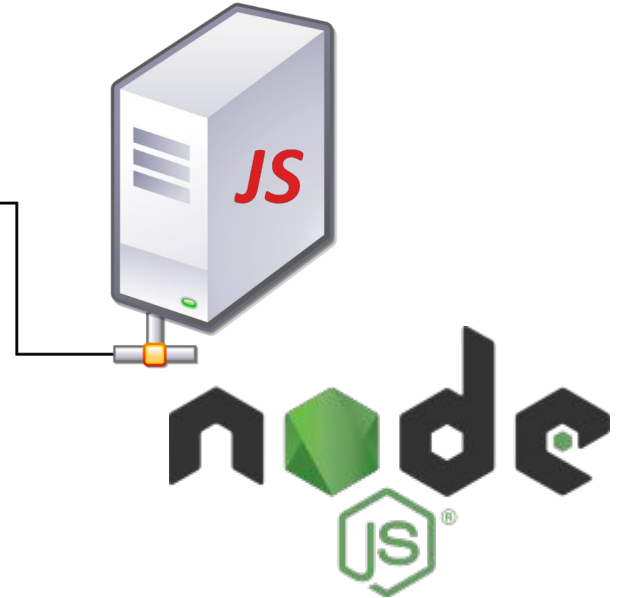


Thank you
Gabriel Cotteli

JS Server Generation?



ExpressJS



Example: Minimal Web Server

- GET /data = read a string from memory
- POST /data = save a string to memory



ExpressJS



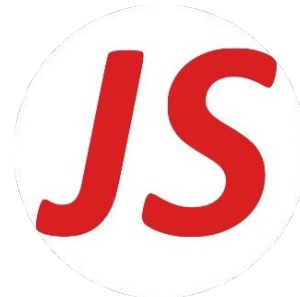
Pharo
100%



1. Write Tests

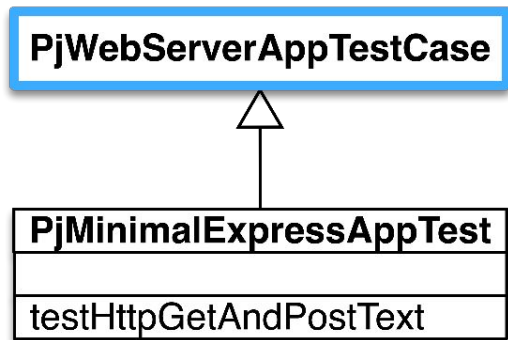
2. Pass the tests

3. Export to JS



100%
JavaScript

Testing Minimal Web Server with *PharJS*



Testing Minimal Web Server with *PharoJS*

PjWebServerAppTestCase



PjMinimalExpressAppTest

testHttpGetAndPostText

testHttpGetAndPostText

| response |

self assert: (self get: '/data') contents isNil.

response := self post: '/data' text: 'PharoJS'.

self assert: response code equals: 200.

self assert: (self get: '/data') contents equals: 'PharoJS'

Testing Minimal Web Server with *PharoJS*

PjWebServerAppTestCase



PjMinimalExpressAppTest

testHttpGetAndPostText

testHttpGetAndPostText

| response |

self assert: (self get: '/data') contents isNil.

response := self post: '/data' text: 'PharoJS'.

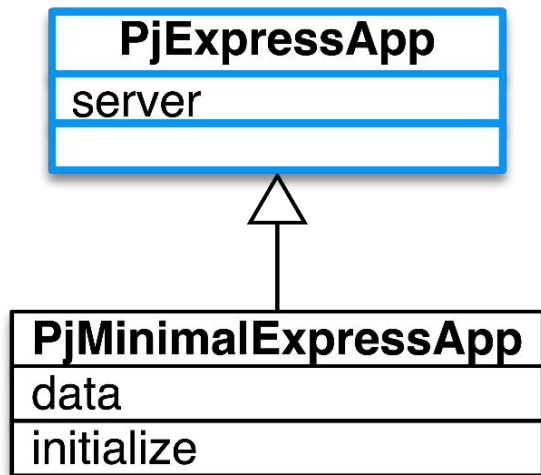
self assert: response code equals: 200.

self assert: (self get: '/data') contents equals: 'PharoJS'

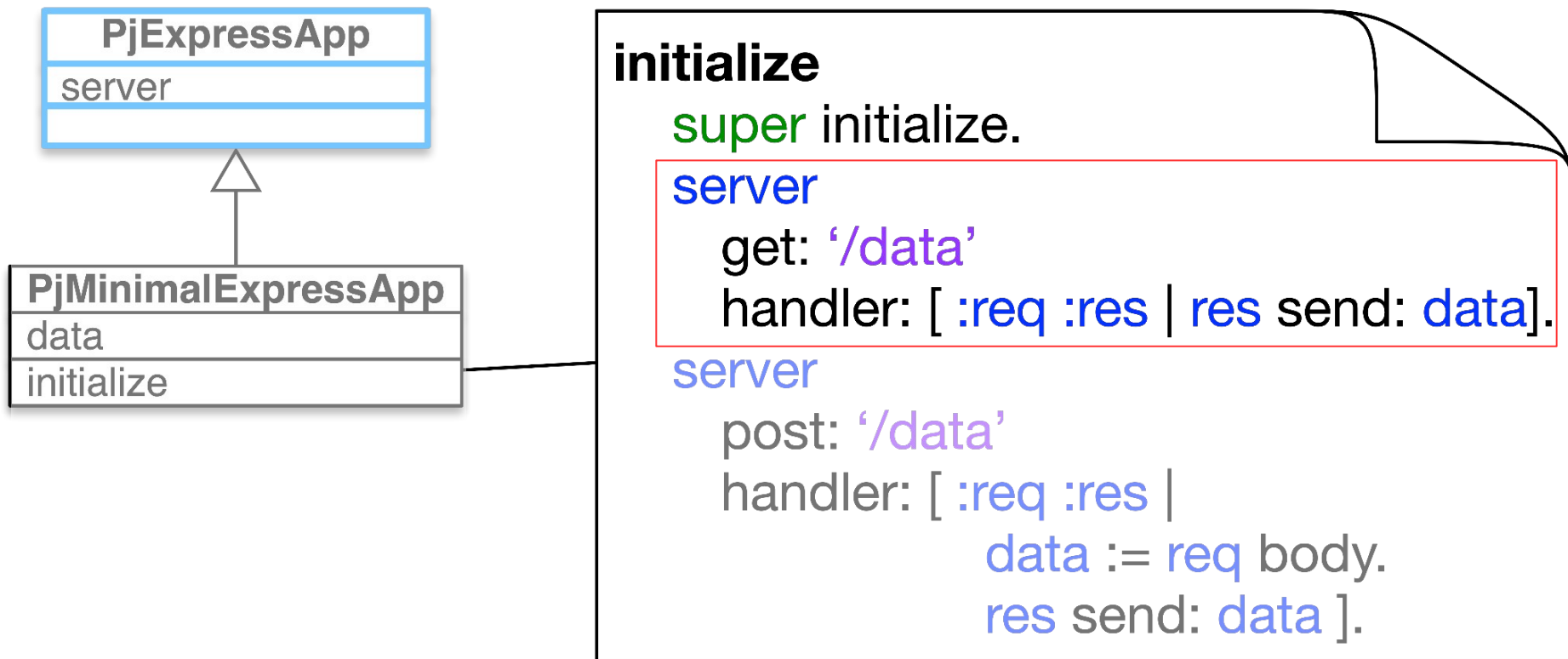
Relies on
Zinc HTTP

Thank you
Sven

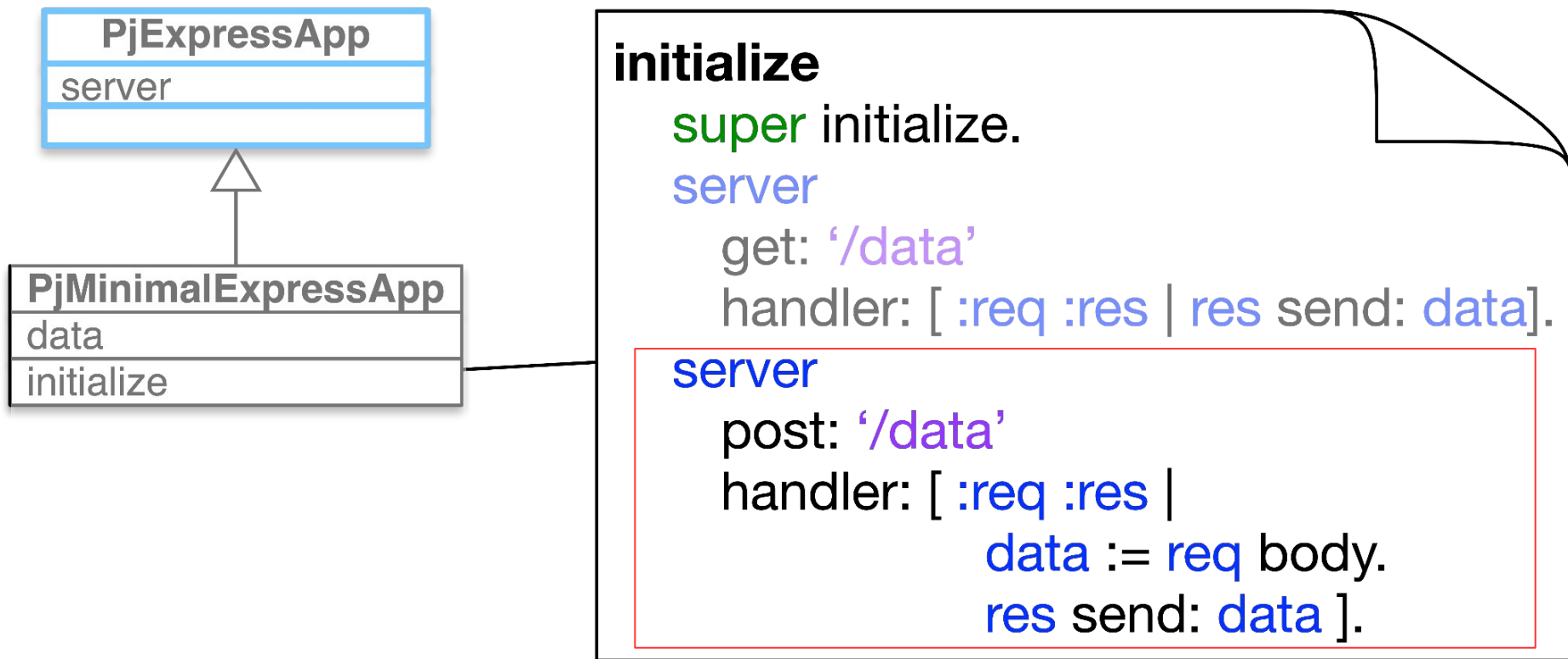
Minimal Web Server with *PharJS*



Minimal Web Server with *PharJS*



Minimal Web Server with *PharJS*



All JS Code is Managed by *Phar* JS

< > PjMinimalExpressApp

Name

>  node_modules

 index.js

 package-lock.json

 package.json

JS packages code from NPM

JS from Pharo code

List of required JS packages

NPM package properties

All JS Code is Managed by *PharJS*

< > PjMinimalExpressApp

| Name |
|-------------------|
| > node_modules |
| index.js |
| package-lock.json |
| package.json |

| |
|---------------------------|
| JS packages from NPM |
| JS from Pharos |
| List of required packages |
| NPM packages |

Installed

Generated from Pharos

Server JS Code Generated with *PharoJS*



Development

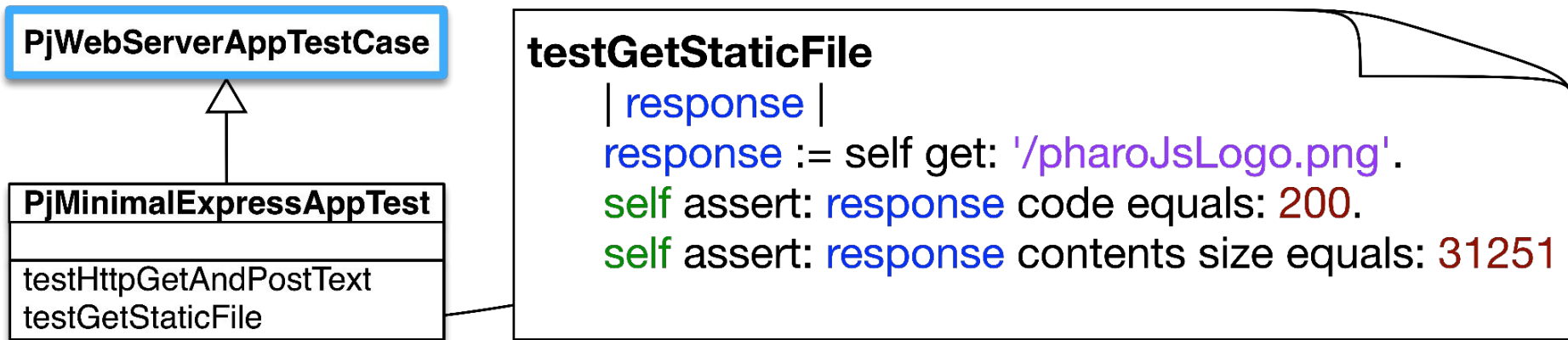
Production

JS *100% Javascript*

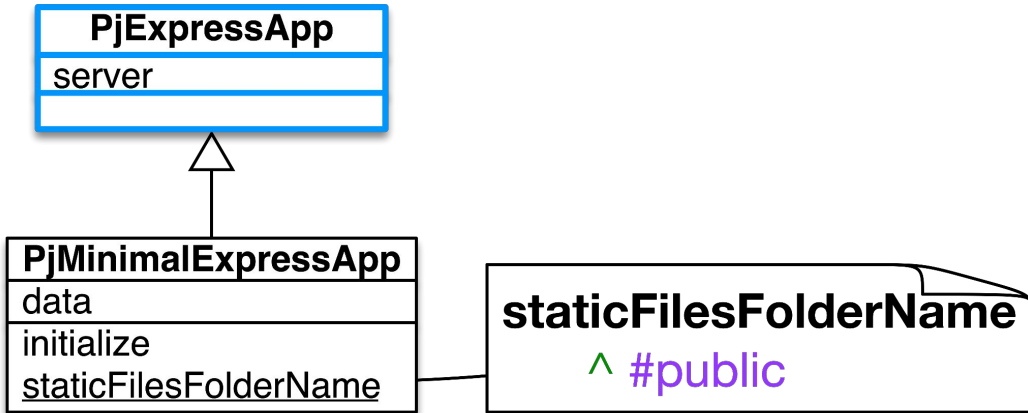
```
1  #!/usr/bin/env node
2  // generated by PharoJS *** DO NOT EDIT *** on 2023-08-05T14:29:56.07175+02:00 by noury
3  'use strict';
4
5  global.Smalltalk=global.Smalltalk??class extends Ob
6  Smalltalk.copyMethods=function(source, target, filter) {
7    let propertyNames = Object.getOwnPropertyNames(source).filter((each)=>
8      /\b[a-z_].test(each)
9    );
10   propertyNames.forEach((name) => {
11     let value=source[name],
12     configurable=Object.getOwnPropertyDescriptor(source, name).configurable;
13   });
14   Smalltalk.fixMethodNames(propertyNames);
15   if (aClass == Object) {
16     if (aClass === Object) {
17       return Object.getPrototypeOf(aClass).prototype;
18     }
19     if (Object.getPrototypeOf(aClass) === Function.prototype){
20       return Object.setPrototypeOf(aClass, Object);
21     }
22   }
```

233 KB

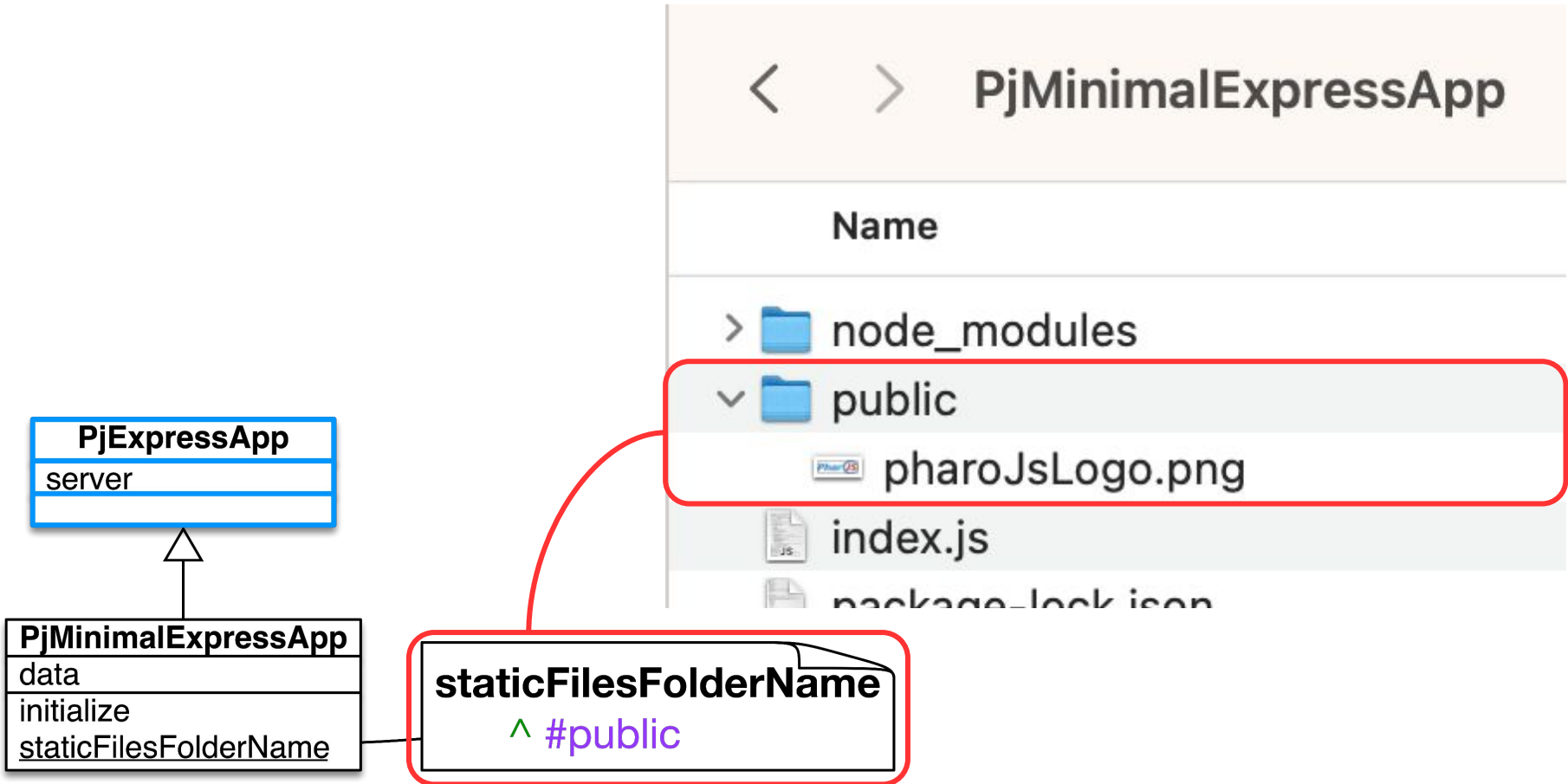
Testing Static File Serving with *PharJS*



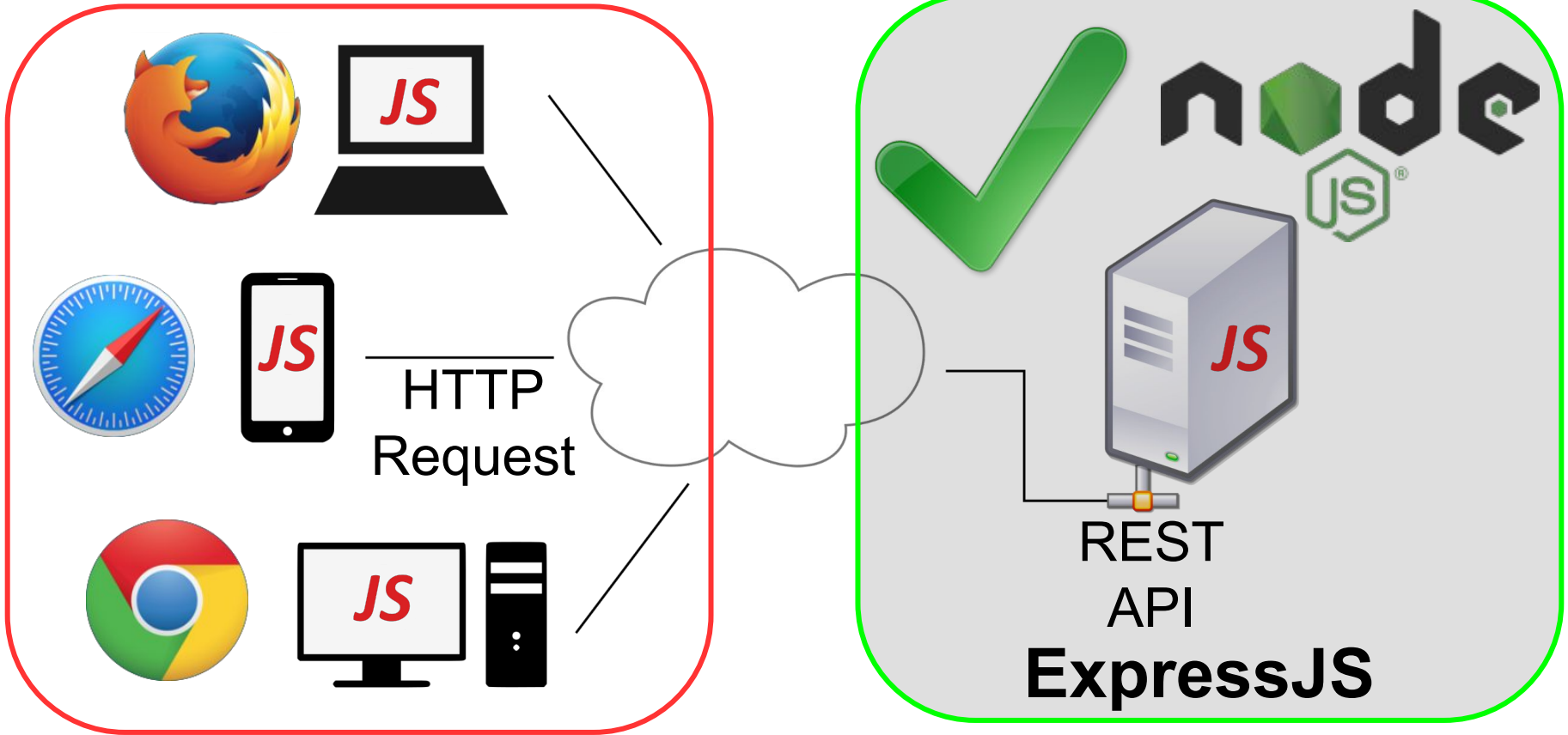
Serving Static Files with *PharJS*



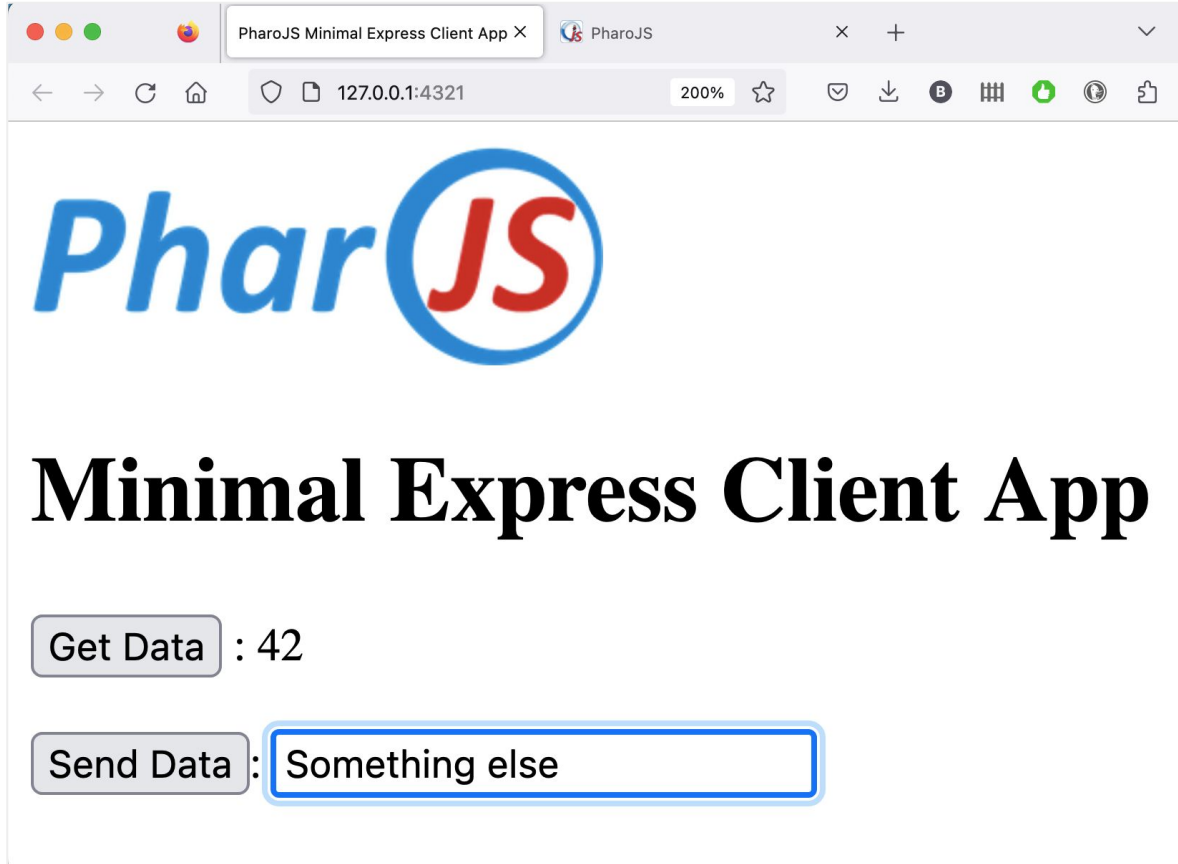
Serving Static Files with *PharJS*



Let's Address the Client



Web Client HTML



The screenshot shows a web browser window with the following elements:

- Browser tabs: "PharoJS Minimal Express Client App X" and "PharoJS".
- Address bar: "127.0.0.1:4321" with a magnification of "200%".
- Logo: "PharJS" where "Phar" is in blue and "JS" is in red inside a blue circle.
- Text: "Minimal Express Client App" in a large, bold, black font.
- Buttons: "Get Data" and "Send Data".
- Input field: A text input field containing "Something else" with a blue border.

index.html

```
<img...>  
<h1>...</h1>  
<button...>...</button>  
<span...>...</span>  
<button...>...</button>  
<input ...>  
<script ...></script>
```

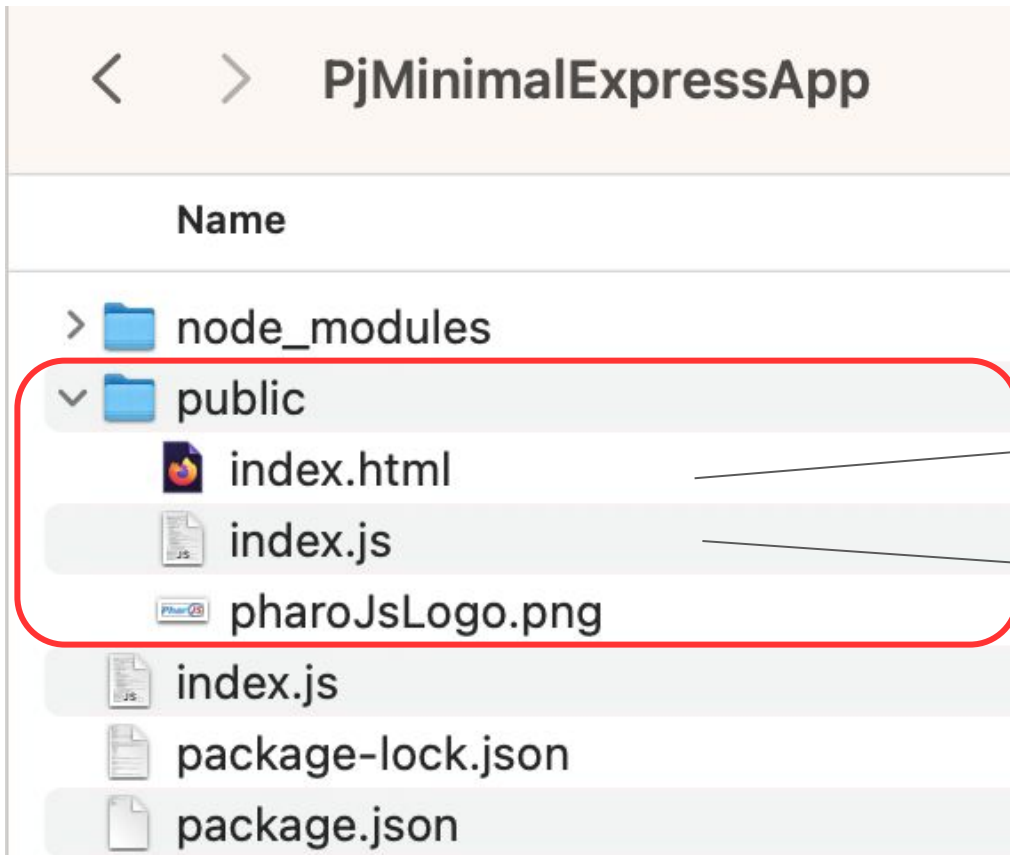
Html Loads JS Generated by *Phar* JS

```
<html>
<body>
  
  <h1>Minimal Express Client App</h1>
  <button id="getDataButton">Get Data</button> :
  <span id="dataDisplaySpan"></span>
  <br><br>
  <button id="postDataButton">Send Data</button> :
  <input type="text" id="dataInput">
  <script src="index.js"></script>
</body>
</html>
```

Html IDs for Integration with *PharJS*

```
<html>
<body>
  
  <h1>Minimal Express Client App</h1>
  <button id="getDataButton">Get Data</button> :
  <span id="dataDisplaySpan"></span>
  <br><br>
  <button id="postDataButton">Send Data</button> :
  <input type="text" id="dataInput">
  <script src="index.js"></script>
</body>
</html>
```

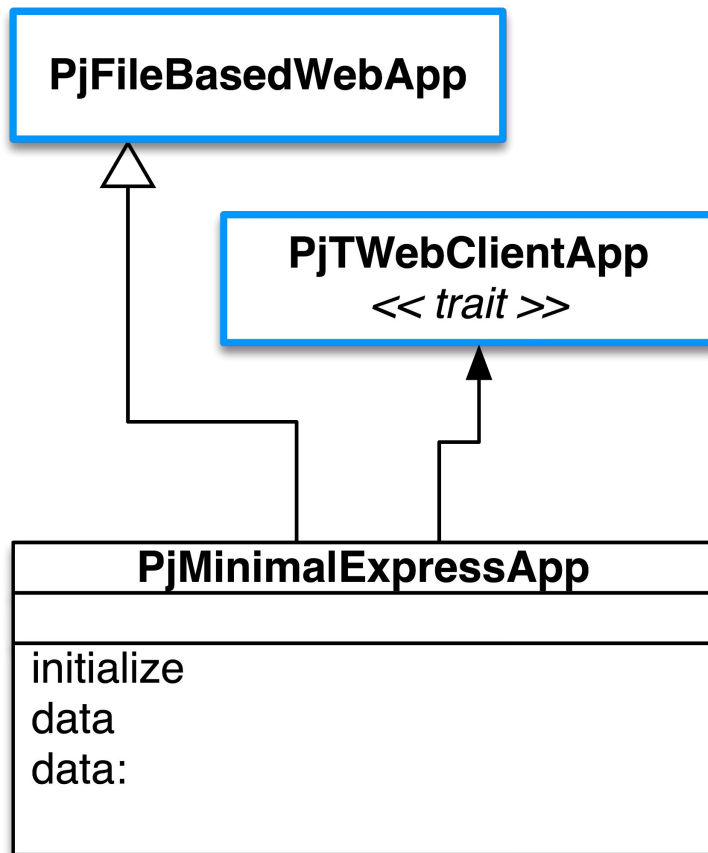
Web Client HTML & JS Served Statically



Client HTML file

Client JS from Pharo code

Web Client Pharo Code with *PharoJS*



```
initialize
  super initialize.
  self
  onClick: #getDataButton
  do: [ self get: '/data' withTextDo: [ :text | self data: text ] ].
  self
  onClick: #postDataButton
  do: [ self post: '/data' text: self data ]
```

```
data
  ^ (self elementAt: #dataInput) value
```

```
data: aString
  (self elementAt: #dataDisplaySpan) textContent: aString
```


PharJS Talks to Server

initialize

`super` initialize.

`self`

onClick: `#getDataButton`

do: [`self` get: `'/data'` withTextDo: [`:text` | `self` data: `text`]].

`self`

onClick: `#postDataButton`

do: [`self` post: `'/data'` text: `self` data]

data

^ (`self` elementAt: `#dataInput`) value

data: `aString`

(`self` elementAt: `#dataDisplaySpan`) textContent: `aString`

PharJS Links DOM Objects using HTML IDs

initialize

`super` initialize.

`self`

`onClick: #getDataButton`

`do: [self get: '/data' withTextDo: [:text | self data: text]].`

`self`

`onClick: #postDataButton`

`do: [self post: '/data' text: self data]`

data

`^ (self elementAt: #dataInput) value`

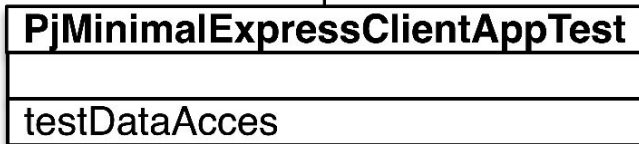
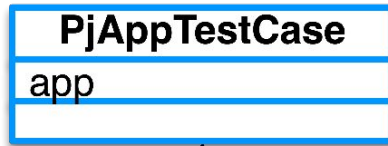
data: aString

`(self elementAt: #dataDisplaySpan) textContent: aString`

Web Client Test with *PharJS*

Instance of
PjMinimalExpressClientApp

JS Object
in the Web Browser



testDataAcces

```
| display |  
display := app elementAt: #dataDisplaySpan.  
self assert: display textContent isEmpty.  
(app elementAt: #dataInput) value: '42'.  
self clickElementById: #postDataButton.  
self clickElementById: #getDataButton.  
self waitUntil: [ display textContent = '42' ]
```

Client JS Code Generated with *PharoJS*



Development

Production

JS *100% Javascript*

A screenshot of a code editor window titled "index.js" showing JavaScript code generated by PharoJS. The code includes a shebang line, a comment indicating it was generated on 2023-08-05T14:29:56.07175+02:00 by noury, and a strict mode directive. It defines a function to copy methods from a source object to a target object, filtering by a regular expression. A large yellow starburst callout with a blue outline is overlaid on the code, containing the text "267 KB". The editor's status bar at the bottom shows "pharo11" and "Ln 1, Col 1 Spaces: 2 UTF-8 LF JavaScript".

Summary

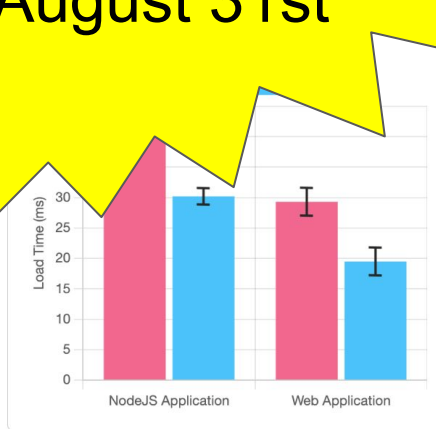
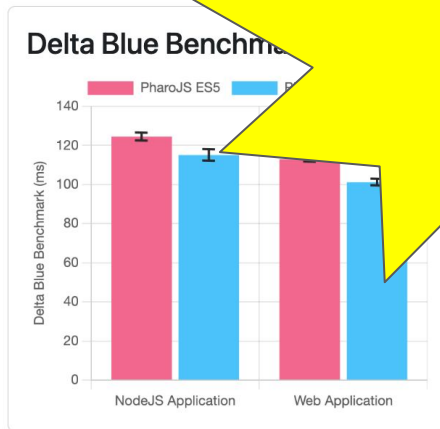
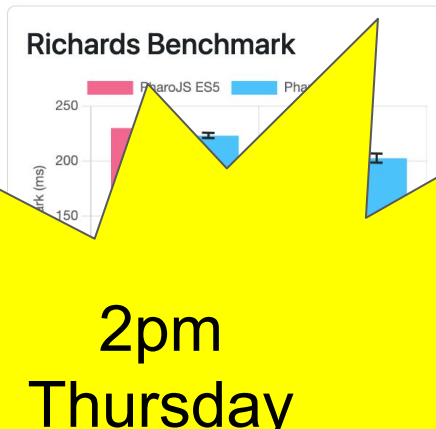
Phar for both Server & Client Sides

- Reuse JS Libraries & Run-Time
- Write 100% Pharo Code
- Test + Debug in Pharo with JS Objects
- Support Different Architectures & Workflows

Other News of *Phar*JS

- Current Stable = Pharo 11
- Continuous Integration
 - SmalltalkCI
 - GitHub Actions
- Improved Performance

IWST Talk on Performance of *PharoJS*



2pm
Thursday
August 31st

Papa Parse
CSV Parser JS Lib



Chart.js
Charting JS Lib

MIT License

Kindly Supported by



PharoJS.org

Develop in Pharo, Run on JavaScript



Thanks to all the contributors!

