



ESUG 2023, Lyon (France)



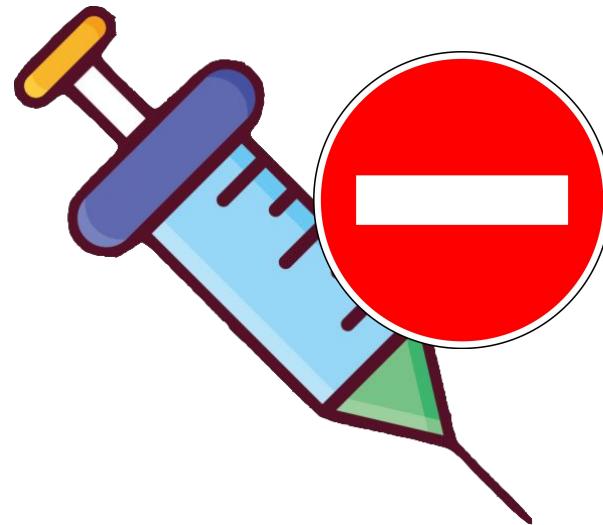
Hijack the JavaScript Ecosystem

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 screenshot shows a GitHub search interface with the query "language:javascript". The results are filtered by "Repositories".

Filter by:

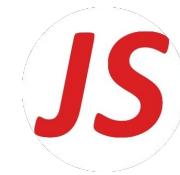
- <> Code: 78.9M
- Repositories: 4.2M** (selected)
- Issues: 15M
- ➡ Pull requests: 72M
- 💬 Discussions: 0
- 👤 Users: 4M

Addict + Lazy = *Phar*

11



+



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 *PharJS*

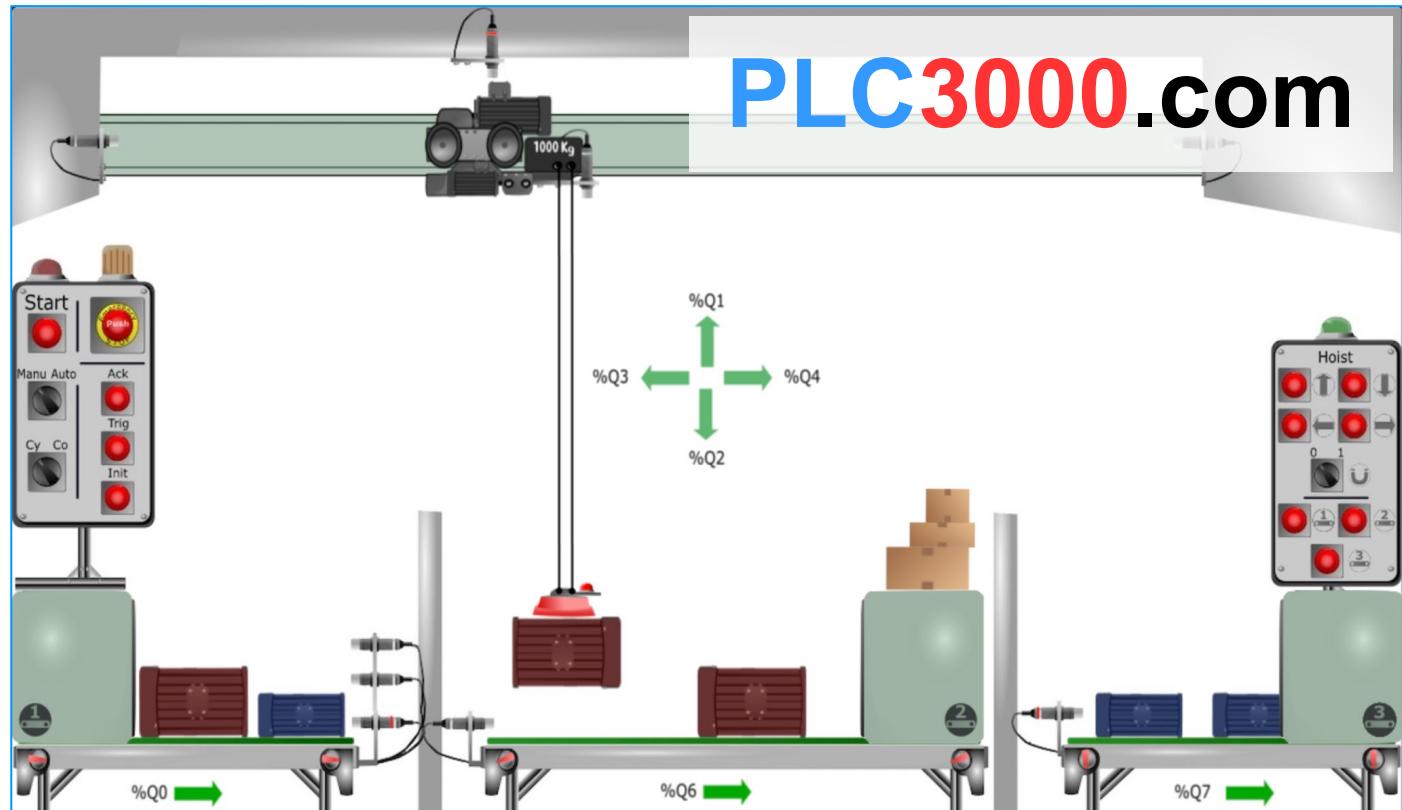
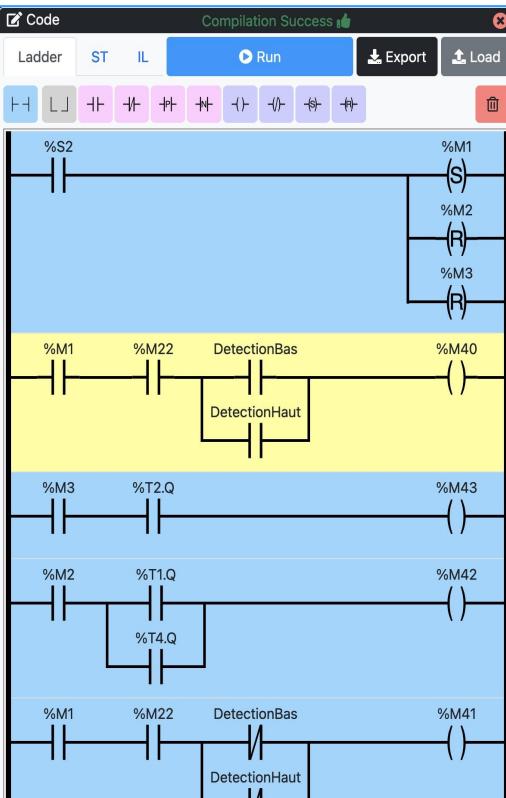


The screenshot shows the Pharo IDE interface. At the top, there's a menu bar with items: Pharo, Browse, Debug, **PharJS**, Sources, System, Library, Windows, and Help. Below the menu, the main workspace displays the 'PharJS-Examples' package. The browser pane lists several classes: ChartJS, Counter, D3JS, DrawRectangle, ExpressJS, HelloWorld, and Minimal. The 'Test Runner' pane shows a list of test cases under the 'pharjs*' category, with many entries having red exclamation marks indicating failures. A status message at the bottom of the runner pane states: '499 ran, 491 passed, 0 skipped, 8 expected failures, 0 failures, 0 errors, 0 as...'. At the very bottom of the interface, there are buttons for 'Run Selected', 'Run Profiled', 'Run Coverage', 'Run Failures', 'Run Errors', and 'File out results'.

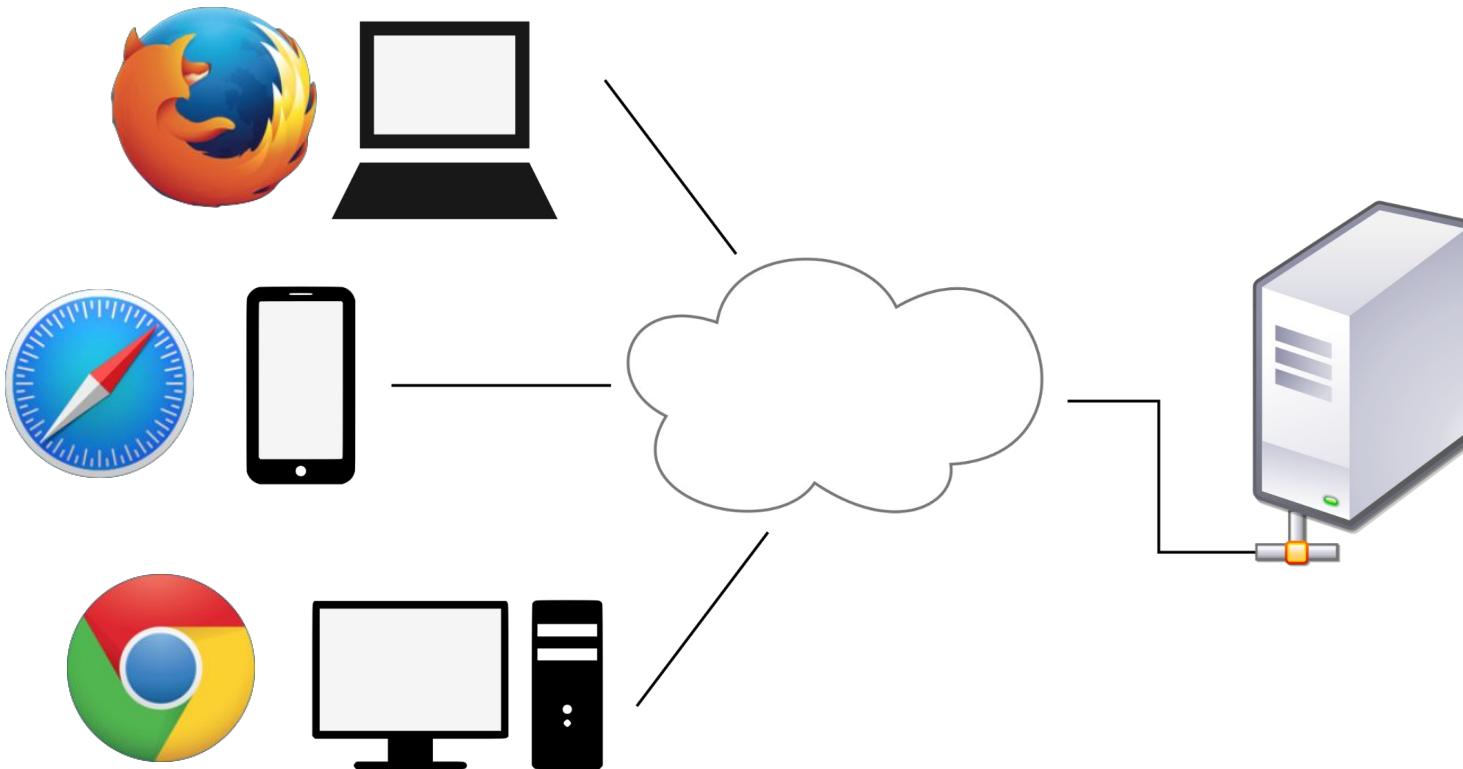
Mobile Apps with *PharJS*



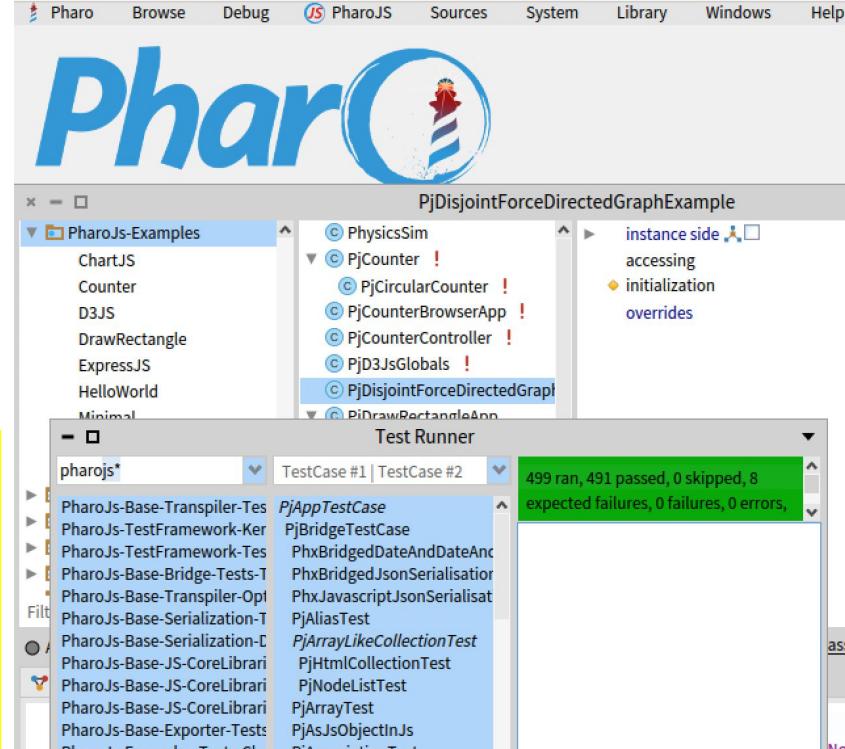
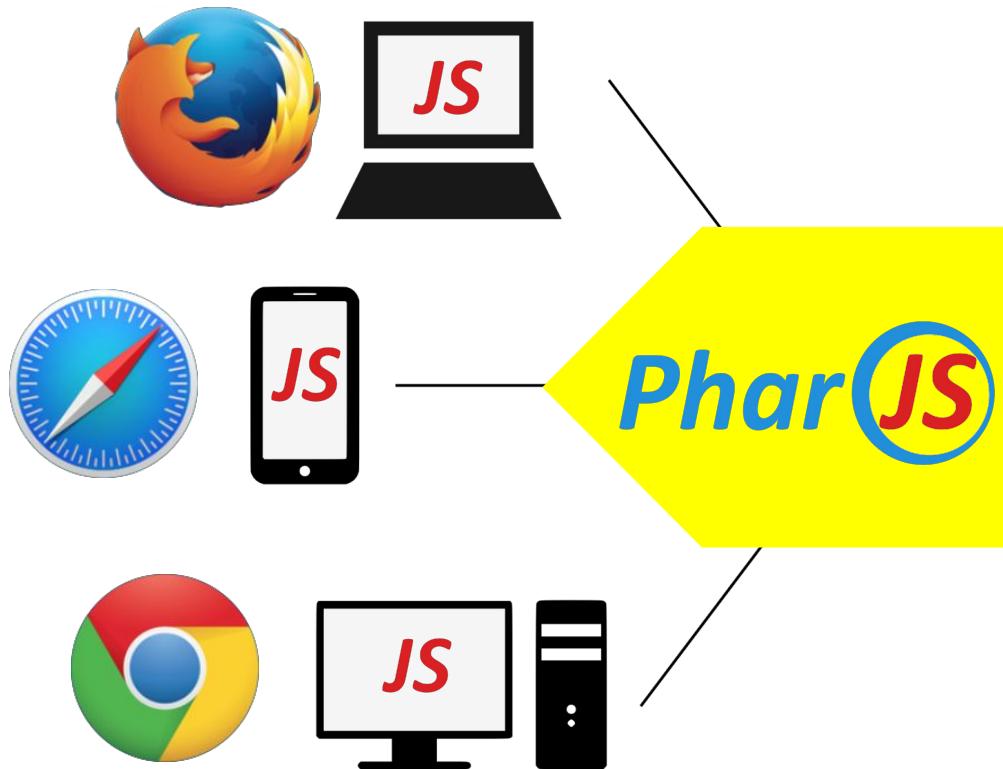
Web Apps with *PharJS*



PLC3000.com Architecture

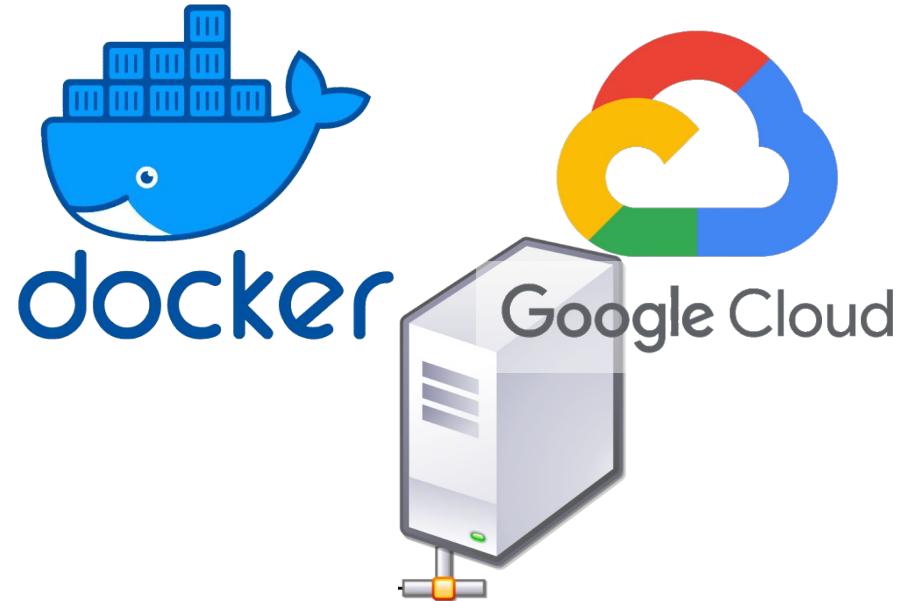
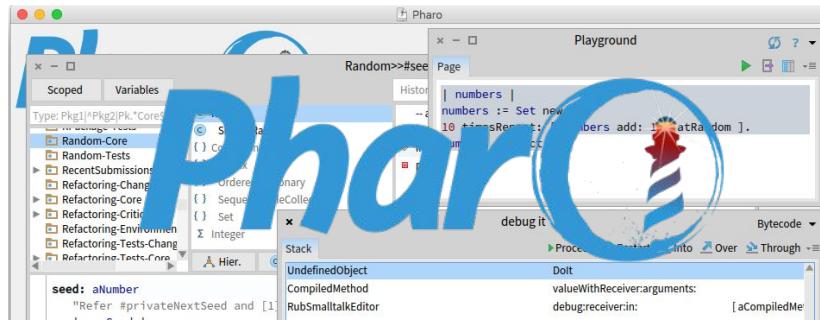


PLC3000.com Clients

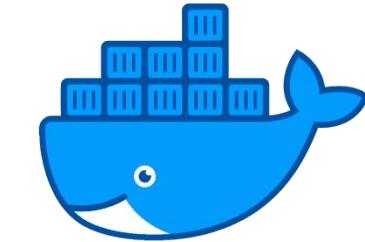
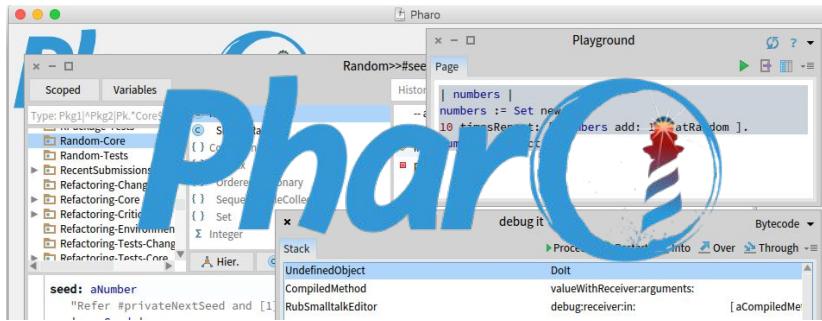


+
matter.js

PLC3000.com Server



PLC3000.com Server



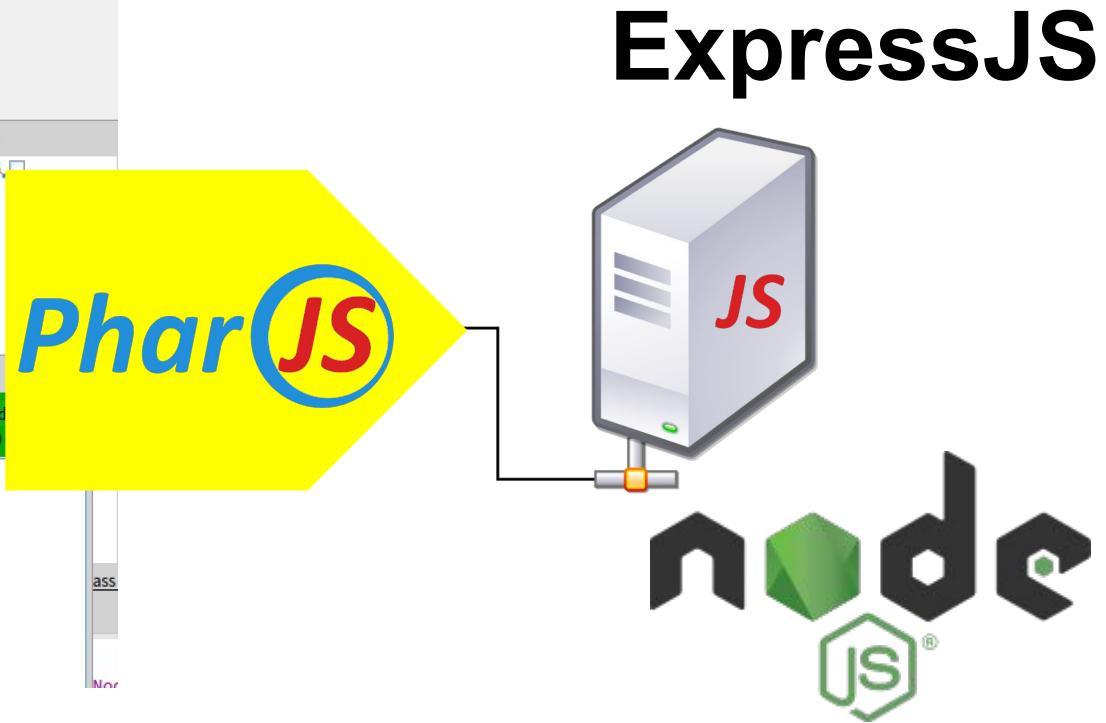
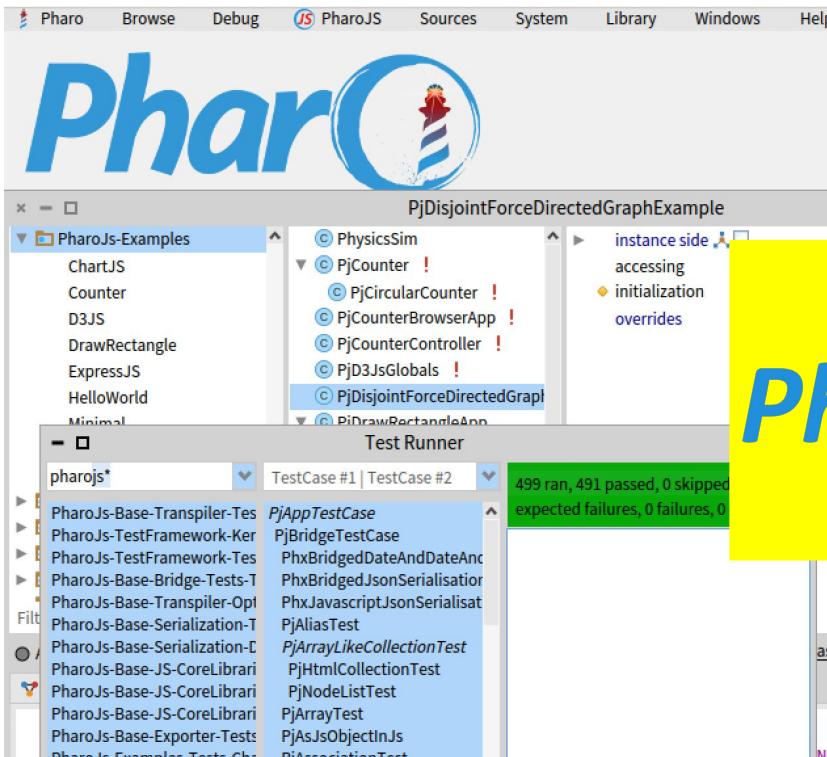
Google Cloud

Thank you
Buenos Aires
Smalltalk



Thank you
Gabriel Cotteli

JS Server Generation?



Example: Minimal Web Server

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



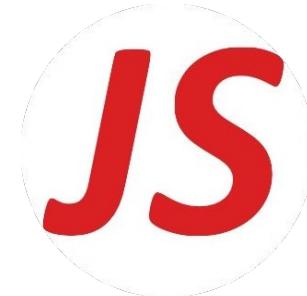
ExpressJS



Pharo
100%



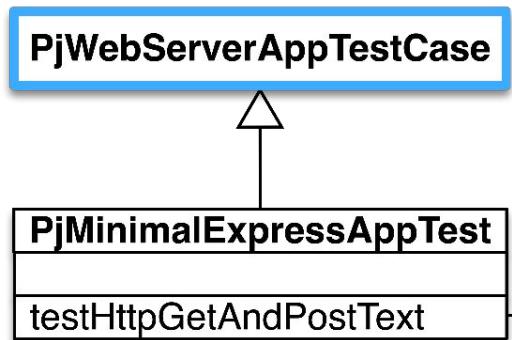
1. Write Tests
2. Pass the tests
3. Export to JS



100%
Javascript

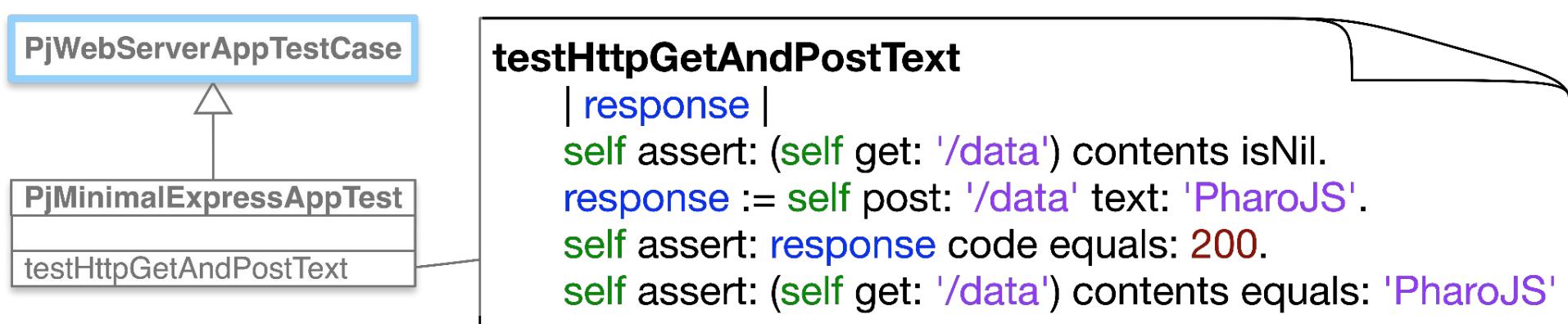
Testing Minimal Web Server with *PharJS*

23



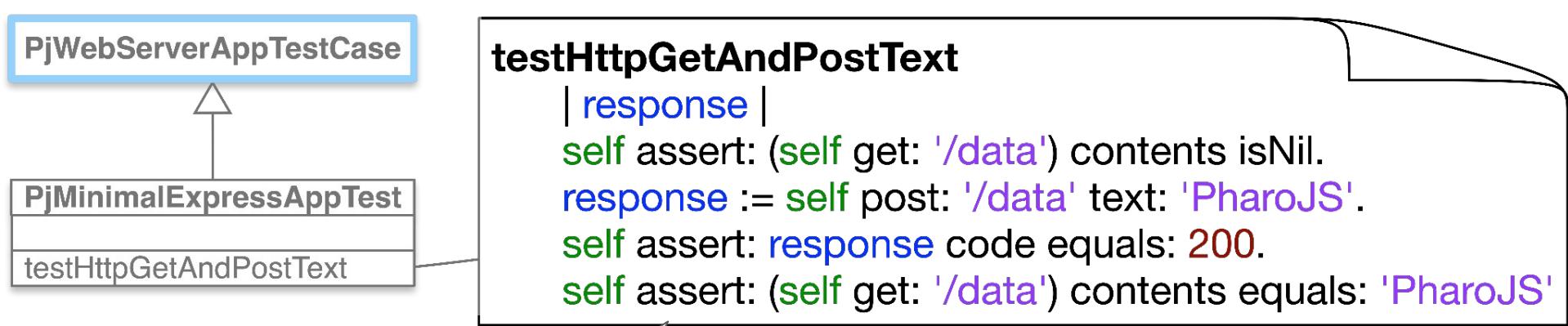
Testing Minimal Web Server with *PharJS*

24



Testing Minimal Web Server with *PharJS*

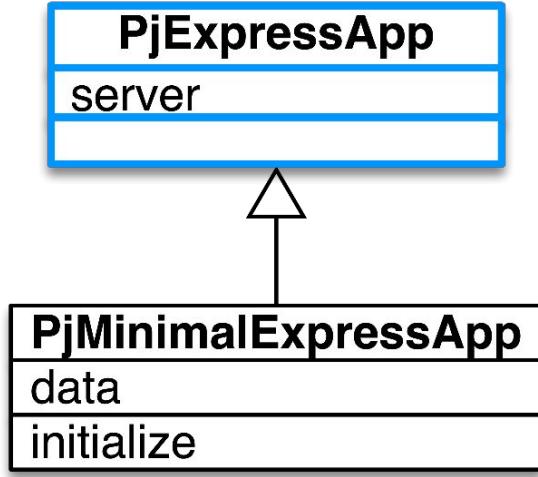
25



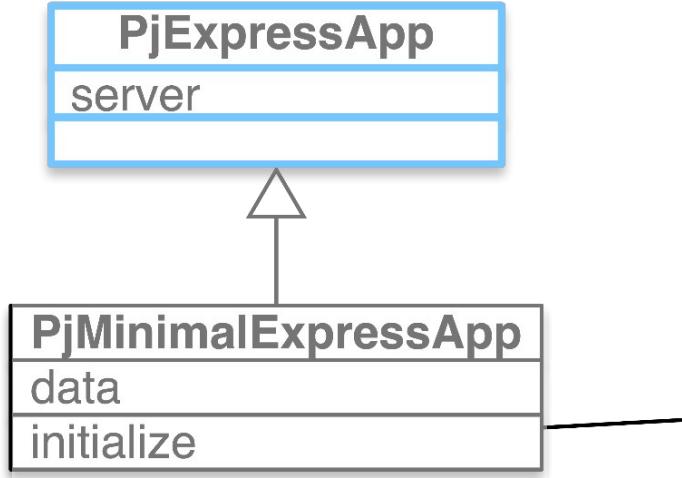
Relies on
Zinc HTTP

Thank you
Sven

Minimal Web Server with *PharJS*



Minimal Web Server with *PharJS*



initialize

`super initialize.`

`server`

`get: '/data'`

`handler: [:req :res | res send: data].`

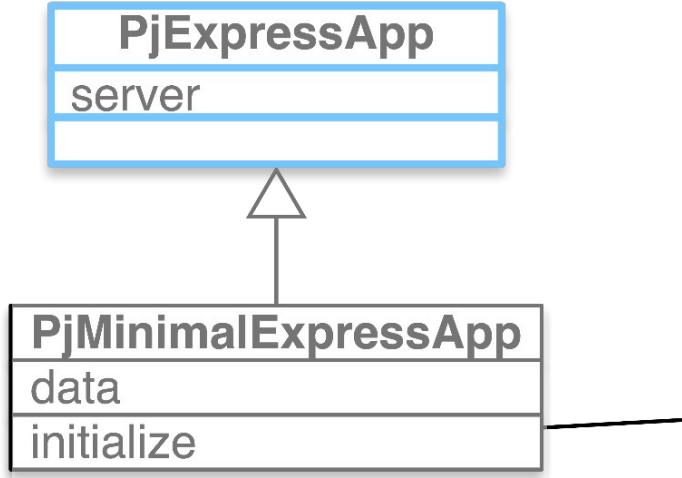
`server`

`post: '/data'`

`handler: [:req :res |`

`data := req body.`
`res send: data].`

Minimal Web Server with *PharJS*



initialize

```
super initialize.
```

```
server
```

```
get: '/data'
```

```
handler: [ :req :res | res send: data ].
```

```
server
```

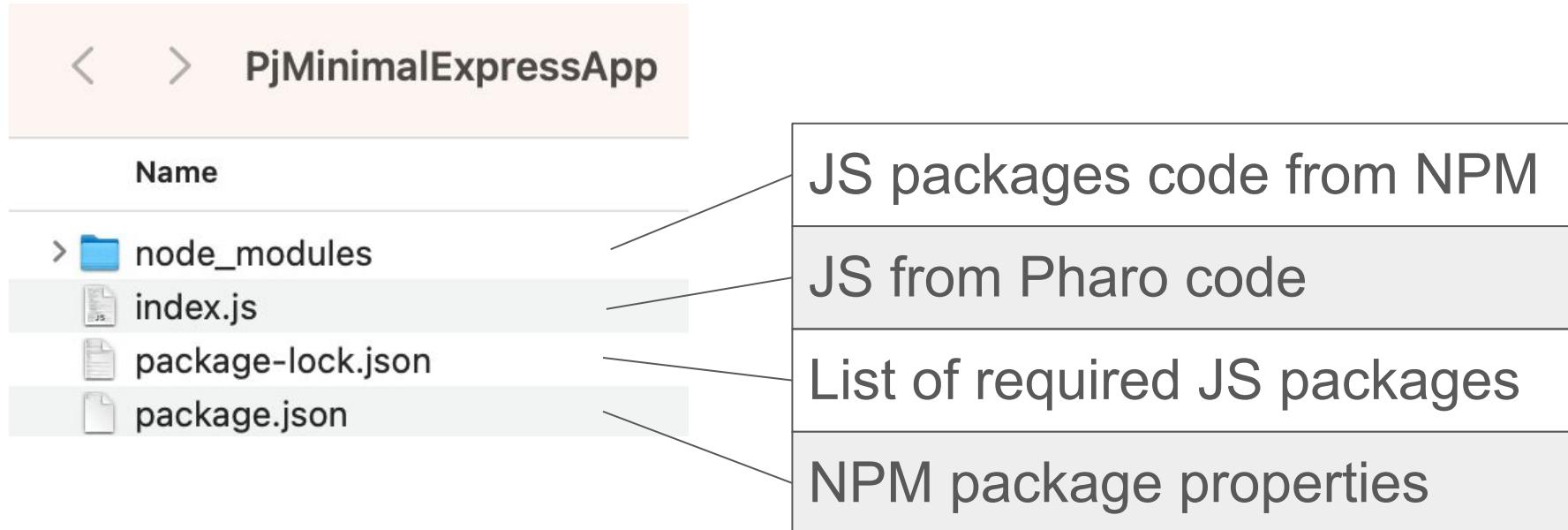
```
post: '/data'
```

```
handler: [ :req :res |
```

```
data := req body.  
res send: data ].
```

All JS Code is Managed by PharJS

29



< > PjMinimalExpressApp

Name

>  node_modules

 index.js

 package-lock.json

 package.json

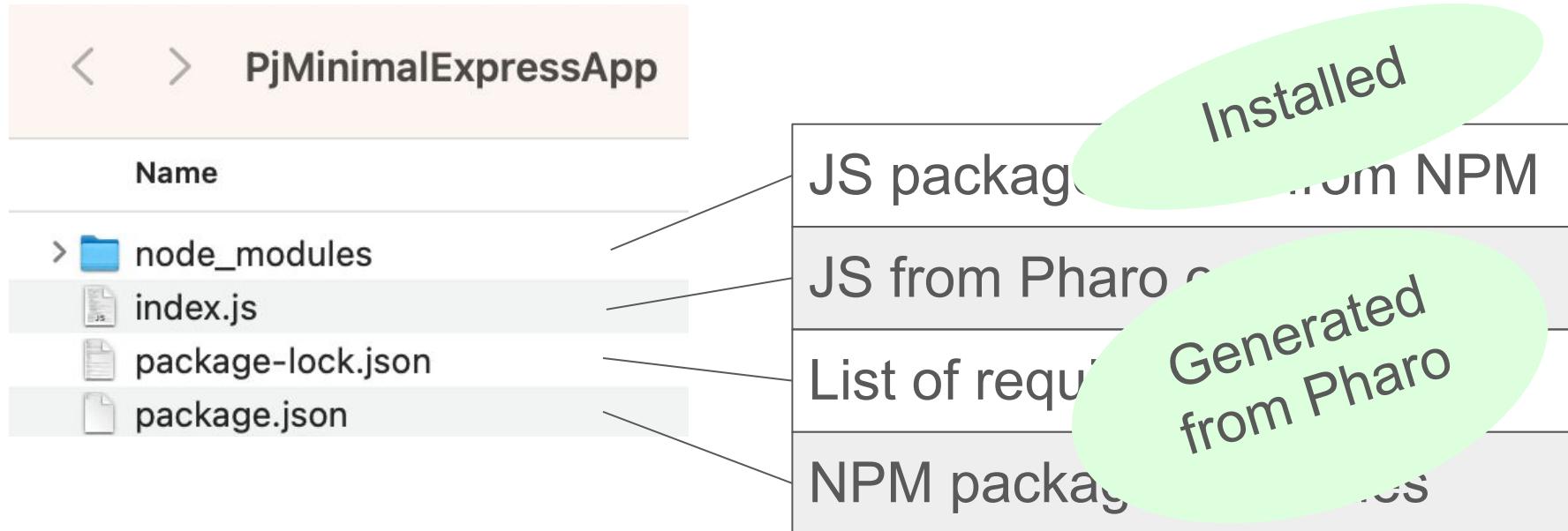
JS packages code from NPM

JS from Pharo code

List of required JS packages

NPM package properties

All JS Code is Managed by PharJS



Server JS Code Generated with *PharJS*



Development

Production

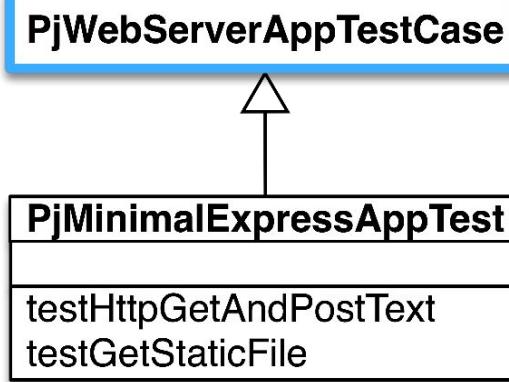
JS *100% Javascript*

The screenshot shows a terminal window titled "PjMinimalExpressApp" displaying a file named "index.js". The code is generated by PharJS and is intended for Node.js execution. It includes a shebang line, a timestamp, and a copyright notice. The code itself is a copy-and-paste operation from Smalltalk, handling methods like "copyMethods" and "fixMethods". A large yellow starburst graphic containing the text "233 KB" is overlaid on the right side of the terminal window.

```
JS index.js x
JS index.js > ...
1  #! /usr/bin/env node
2  // generated by PharJS *** DO NOT EDIT *** on 2023-08-05T14:29:56.07175+02:00 by nouri
3  'use strict';
4
5  global.Smalltalk=global.Smalltalk??class
6  Smalltalk.copyMethods=function(source,+
7    let propertyNames =
8      \bphj_/.test(each)
9    );
10   property,
11   configurable
12 } Smalltalk.fixMethods
13 {
14   if (aClass ==
15     if (aClass ===
16       return Obj
17     }
18   if (Object.getPrototypeOf(aClass) ==
19     return Object.setPrototypeOf(aClass, Function.prototype);
20   }
21 };
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript

Testing Static File Serving with

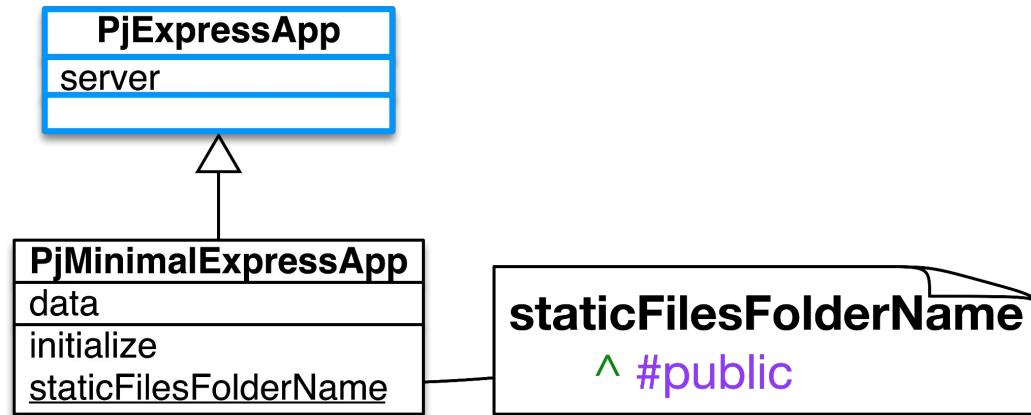


testGetStaticFile

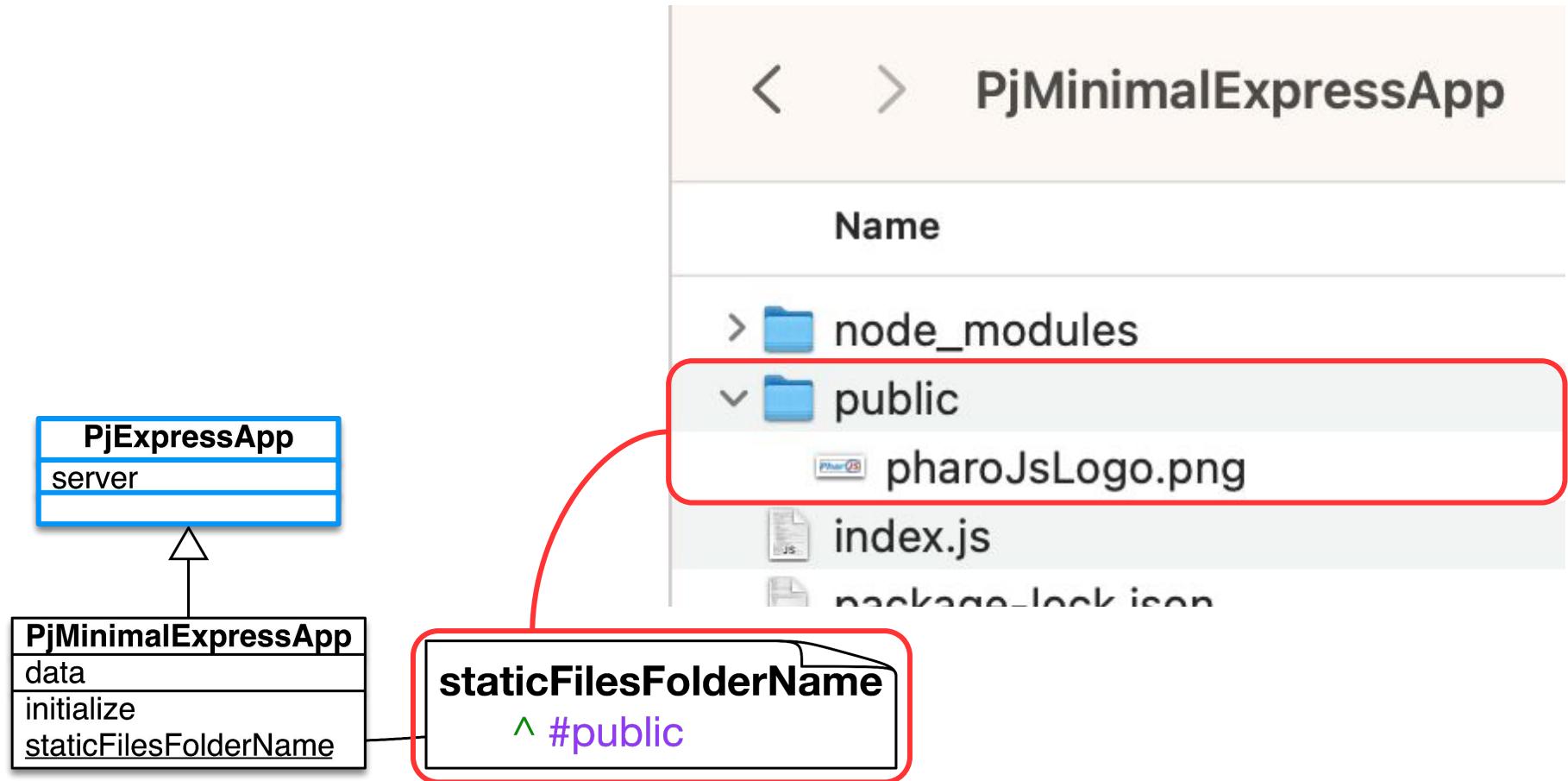
```
| response |
response := self get: '/pharoJsLogo.png'.
self assert: response code equals: 200.
self assert: response contents size equals: 31251
```

Serving Static Files with *PharJS*

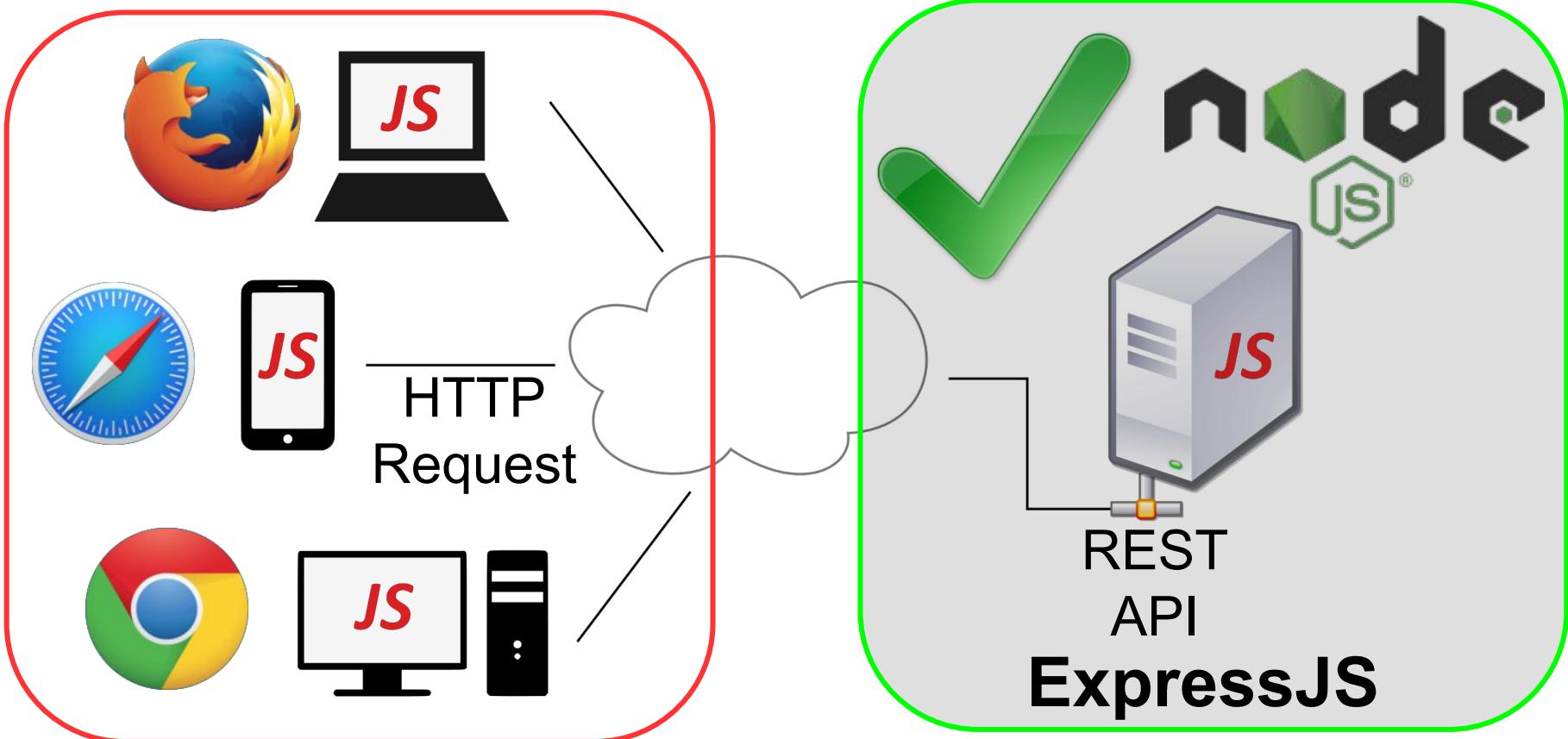
33



Serving Static Files with *PharJS*



Let's Address the Client



Web Client HTML

PharJS Minimal Express Client App X PharoJS 127.0.0.1:4321 200%

PharJS

Minimal Express Client App

Get Data : 42

Send Data : Something else

index.html

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

Html Loads JS Generated by PharoJS

```
<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 PharoJS

```
<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

< > PjMinimalExpressApp

Name

>  node_modules

✓  public

 index.html

 index.js

 pharoJsLogo.png

 index.js

 package-lock.json

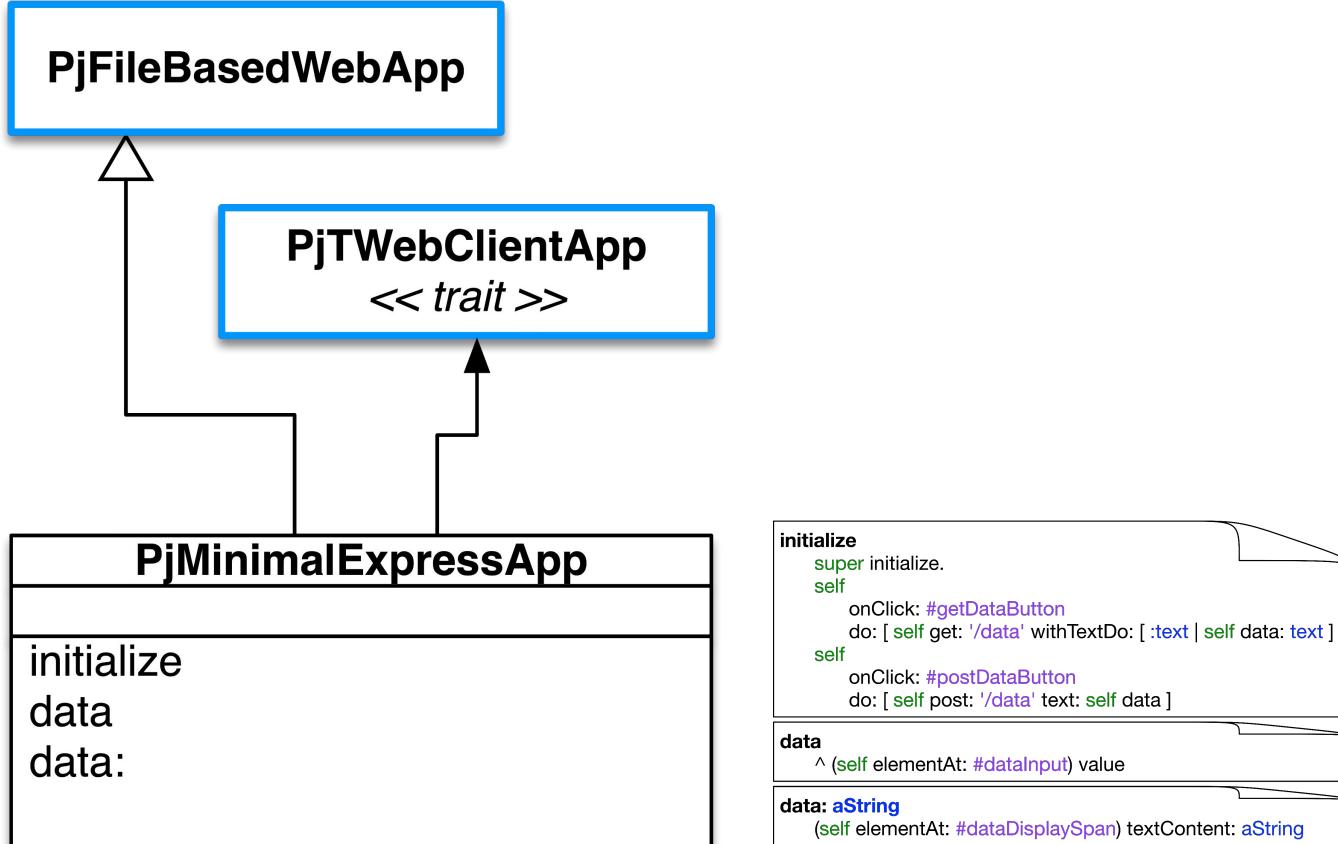
 package.json

Client HTML file

Client JS from Pharo code

Web Client Pharo Code with *PharJS*

40



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

42

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 ]
```

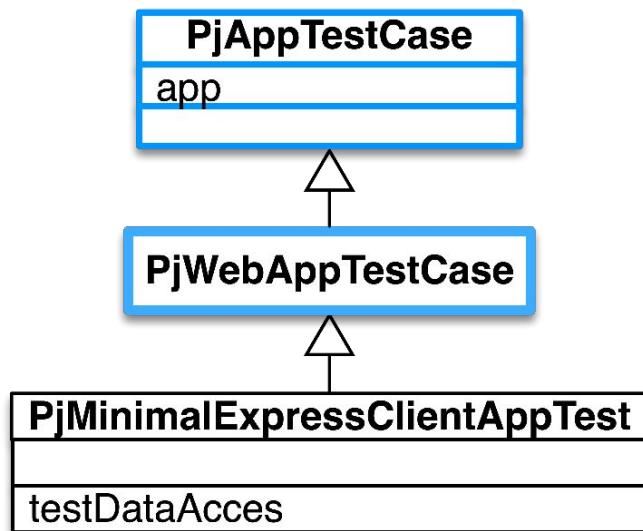
data

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

data: aString

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

Web Client Test with 43



testDataAcces

```
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' ]
```

A red box highlights the variable `app` in the first line of the code, which corresponds to the `app` attribute in the **PjAppTestCase** class.

Instance of
PjMinimalExpressClientApp

JS Object
in the Web Browser

Client JS Code Generated with *PharoJS*



Development Production

JS **100%**
Javascript

The screenshot shows a browser window titled "PjMinimalExpressApp". The tab bar shows "JS index.js x" and "JS index.js > ...". The main content area displays a block of JavaScript code. A large yellow starburst graphic is overlaid on the right side of the code, containing the text "267 KB". The status bar at the bottom of the browser window shows "Ln 1, Col 1" and "Spaces: 2".

```
JS index.js x
JS index.js > ...
1  #! /usr/bin/env node
2  // generated by PharoJS *** DO NOT EDIT *** on 2023-08-05T14:29:56.07175+02:00 by nouri
3  'use strict';
4
5  global.Smalltalk=global.Smalltalk??class
6  Smalltalk.copyMethods=function(source,tends 0b
7    let propertyNames =
8      /\bjpj_/.test(each)
9    );
10   propertyNames
11  }
12  Smalltalk.fixMethods
13  {
14    if (aClass ==
15    if (aClass ===
16    return Obj
17    }
18    if (Object.getPrototypeOf(aClass) ==
19    return Object.setPrototypeOf(aClass, Function.prototype);
20  };
21  
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript

Summary

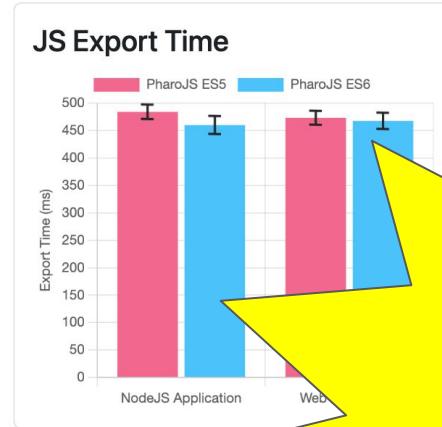
PharJS for both Server & Client Sides

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

Other News of PharJS

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

IWST Talk on Performance of *PharJS*



2pm
Thursday
August 31st

Papa Parse
CSV Parser JS Lib



MIT License

Kindly Supported by



PharoJS.org

Develop in Pharo, Run on JavaScript



GitHub



Thanks to all the contributors!