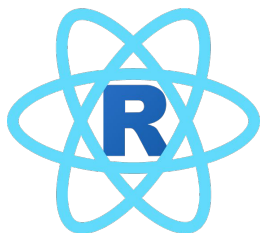# Integrating React.js and Shiny

Or, how to use cool JavaScript libraries from R with minimal effort

Alan Dipert
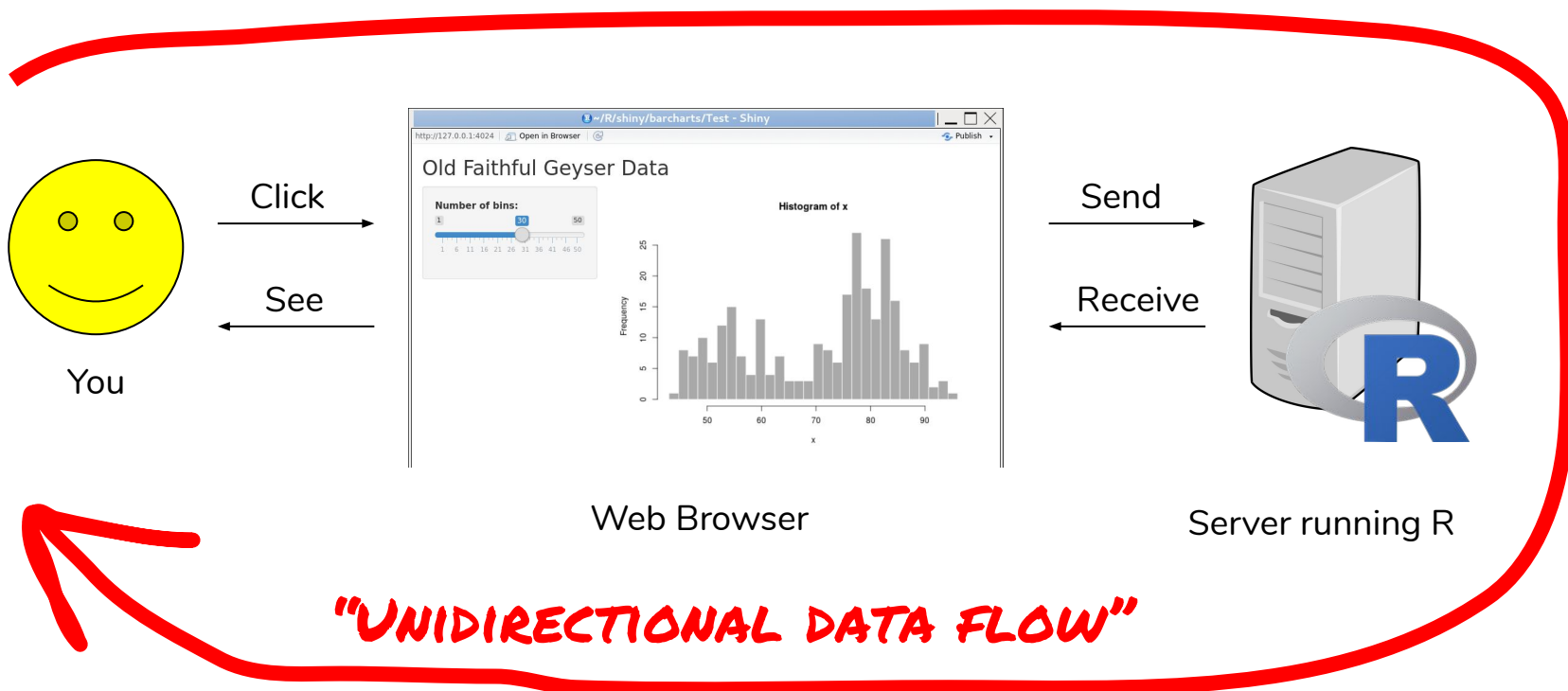Software Engineer, RStudio
@alandipert

Objective

Help you understand enough about Shiny, JavaScript, and React.js to see the utility of the reactR package and motivate you to learn more about it.

Outline

1. How Shiny works: unidirectional data flow
2. What happens in the web browser
3. What is React.js?
4. What is reactR?
5. Demonstration: Adapting a React library

How Shiny works

You — Click → See ←

Web Browser

Old Faithful Geyser Data

Number of bins: 1 — 30 — 50

Histogram of x

Send → Receive ←
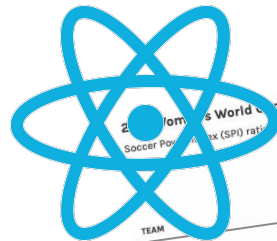
Server running R

"Unidirectional data flow"

Inside the Web Browser

1. DOM events collected from sliders, buttons, text boxes, etc.
2. Events filtered, aggregated
3. Events combined in JavaScript into a single outgoing WebSocket message of **input values**
4. Incoming WebSocket messages received in JavaScript
5. Messages contain multiple incoming **output values**
6. Output values mapped to plots, tables, etc. in the DOM
7. GOTO 1
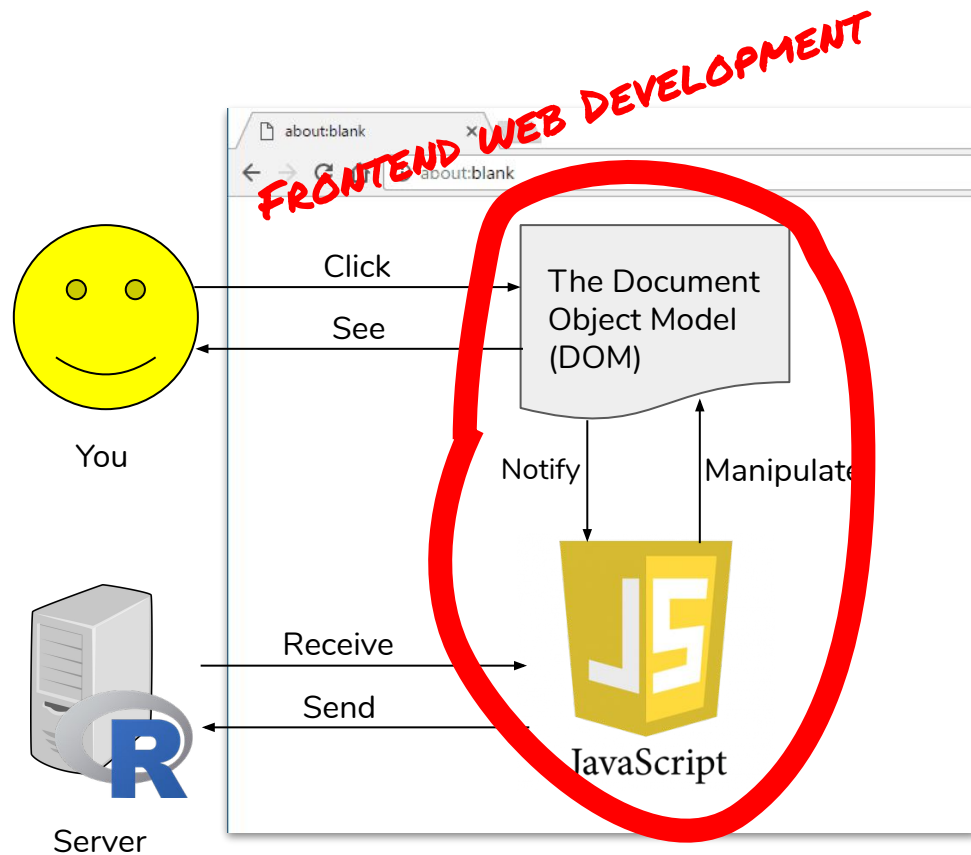
React.js: for Frontend Web Development

1. There are many different ways to approach frontend web development
2. React.js is one of the most popular
3. A React **component** describes reusable appearances and behaviors
4. Components are the building blocks of a React-based interface
5. There is a large ecosystem of stock React components packaged as **npm libraries**

reactR: Leverage React in an R context

- Introduces the world of R to the world of React
- Documents and recommends hybrid R/JavaScript project tools and configuration
- Provides JavaScript for using React components as easily as possible
- Builds on existing Shiny/htmlwidget APIs
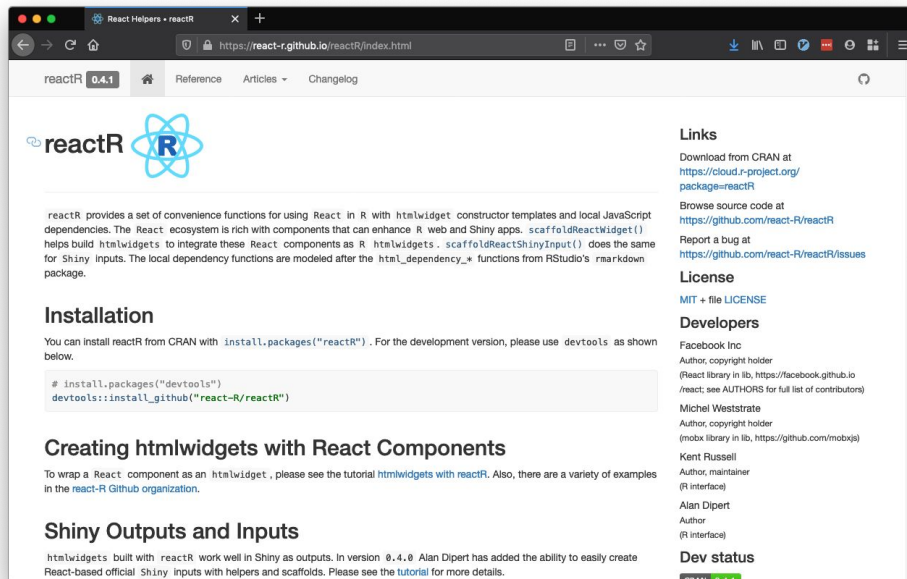
A Kent Russell joint
@timelyportfolio

Demo: Color Picker Input

Learning Resources

- reactR: https://react-r.github.io/reactR/
- Learn JavaScript: https://github.com/getify/You-Dont-Know-JS
- Cool JavaScript ideas: https://wattenberger.com/