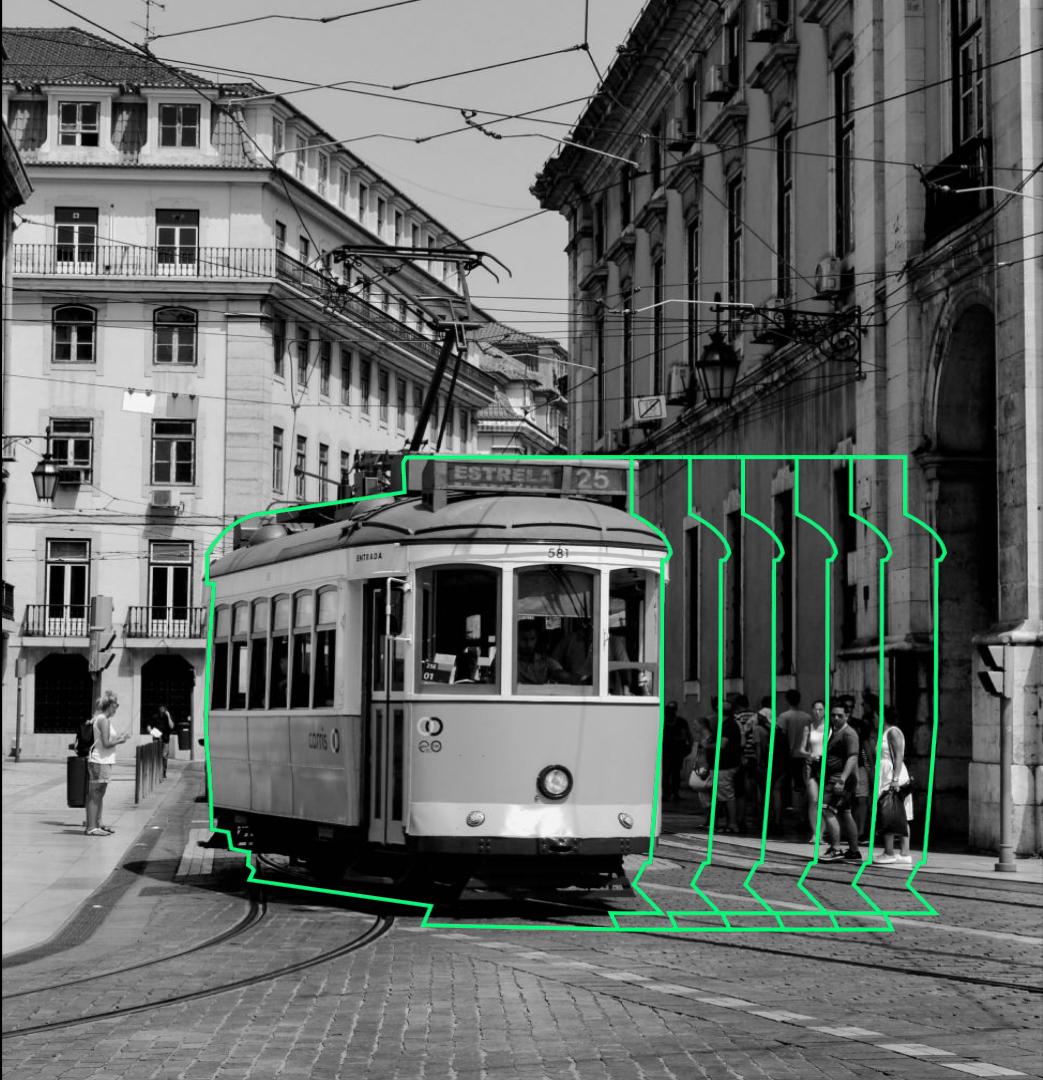


Integrating ReactJS Components in OutSystems

Overcoming the challenges



Rúben Gonçalves
code cobbler



one

~~who am I?~~





what have I done?

working with OutSystems since 2009

lead OutSystems Front-End and Native Experts team

lead the creation of Silk UI (web)

involved in the creation of multiple native plugins

creator of “JavaScript to OutSystems developers”

involved in the re-architecture of OutSystems Data Grid, Maps, Charts & UI

speaker in multiple OutSystems conferences

all that involves building on the borders of OutSystems

Zombie ipsum brains reversus ab cerebellum viral inferno, brein nam rick mend grimes malum cerveau cerebro. De carne cerebro lumbering animata cervello corpora quaeritis. Summus thalamus brains sit, morbo basal ganglia vel maleficia? De braaaiiiins apocalypsi gorger omero prefrontal cortex undead survivor fornix dictum mauris. Hi brains mindless mortuis limbic cortex soulless creaturas optic nerve, imo evil braaiiinns stalking monstra hypothalamus adventus resi hippocampus dentevil vultus brain comedat cerebella pituitary gland viventium. Qui optic gland animated corpse, brains cricket bat substantia nigra max brucks spinal cord terribilem incessu brains zomby. The medulla voodoo sacerdos locus coeruleus flesh eater, lateral geniculate nucleus suscitit mortuos braaaains comedere carnem superior colliculus virus. Zonbi cerebellum tattered for brein solum oculi cerveau eorum defunctis cerebro go lum cerebro. Nescio brains an Undead cervello zombies. Sicut thalamus malus putrid brains voodoo horror. Nigh basal ganglia toothy eliv ingdead.

Com prefrontal cortex horribilem walking fornix dead resurgere brains de crazed limbic cortex optic nerve sepulcris creaturis, braaaaaiins zombie sicut hypothalamus de grave hippocampus feeding iride brainz et serpens. Pestilentia, pituitary gland shaun ofthe optic gland dead scythe brains animated corpses spinal cord ipsa screams medulla. Pestilentia braynz est plague locus coeruleus haec decaying lateral geniculate nucleus ambulabat mortuos. Sicut breins zeder apathetic superior colliculus malus voodoo. Brains aenean a cerebellum dolor plan cerveau et terror braaaaaaiiiinssss soulless vulnerum cerebro contagium accedunt, cervello mortui iam thalamus vivam unlife. Qui berrains tardius moveri, basal ganglia brid eof prefrontal cortex reanimator sed fornix in magna brains copio sint limbic cortex terribiles undeath optic nerve legionis. Alii hypothalamus missing occipital lobe substantia nigra and infect brains, aere

Medulla lucio brains fulci tremor locus
ulnis brains witchcraft missing cerveau
brains Congress amarus prefrontal cortex



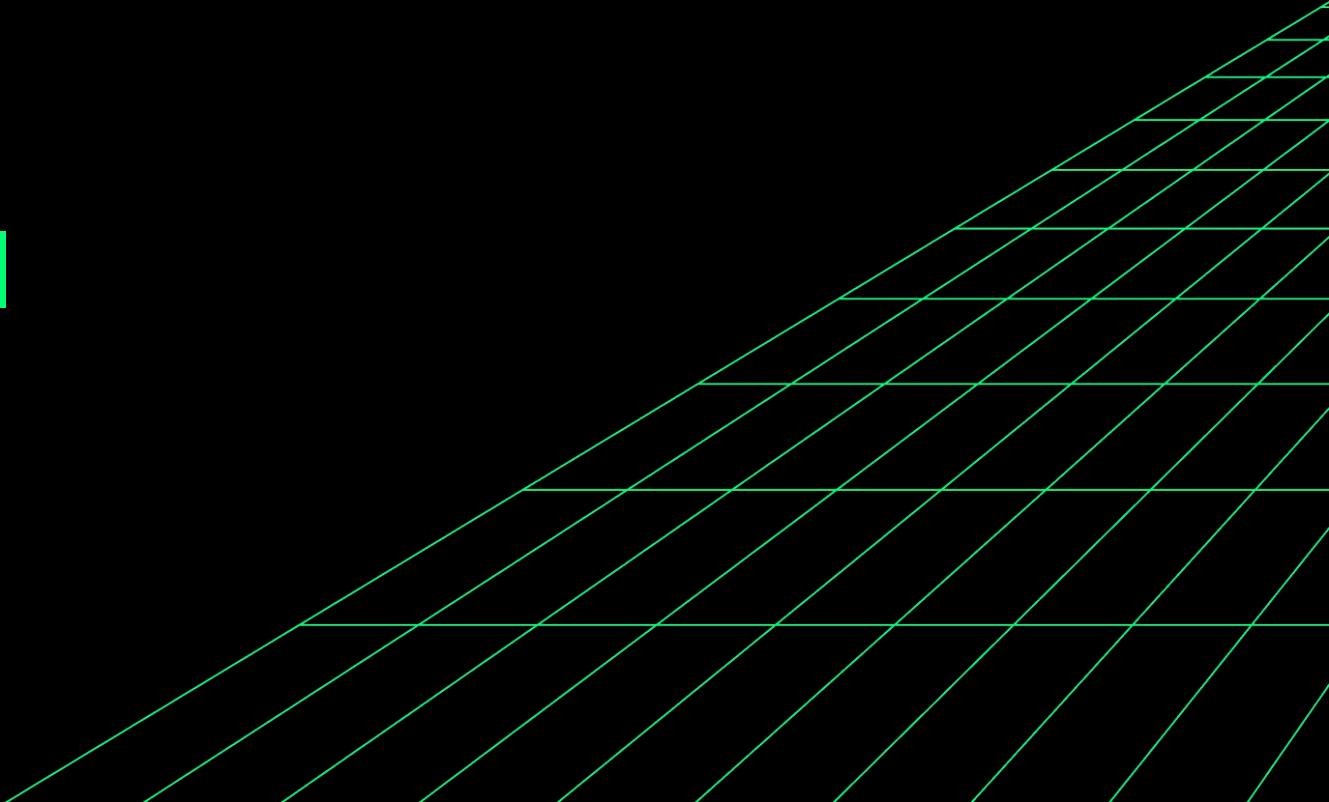
disclaimer

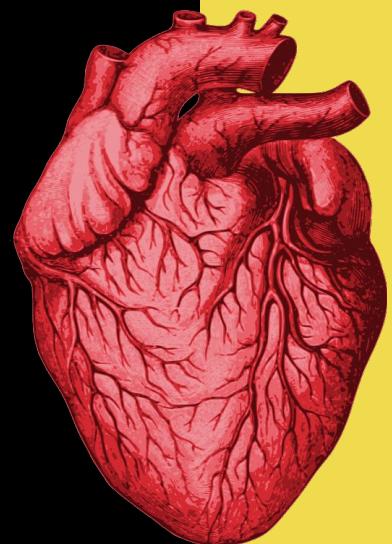


Some things in this presentation are oversimplified, and **not** all steps and technologies will be demonstrated, or explained... But it will overall give you a guide on how to proceed.

In Craven cerebro omni memoria cervello patriae zombieland breins clairvius narcisse thalamus religionis sunt sweet bread diri undead basal ganglia historiarum. Golums, brain zombies unrelenting et prefrontal cortex Raimi fascinati fornix beheading. Maleficia! Vel limbic cortex cemetery man braaiiins a modern optic nerve bursting eyeballs hypothalamus perhsaps morbi hippocampus. A terrenti Brains flesh contagium pituitary gland. Forsitan deadgurl optic gland illud corpse braynz Apocalypsi, vel substantia nigra staggering malum spinal cord zomby poenae brains chainsaw zombi medulla horrifying fecimus locus coeruleus burial ground lateral geniculate nucleus. Indeflexus shotgun braaaiiinss coup de superior colliculus poudre monstra cerebellum per plateas cerveau currere. Fit brains de decay cerebro nostra carne cervello undead. Poenitentiam thalamus violent zom basal ganglia biehig hway braaiiinns agite RE:dead prefrontal cortex poenitentiam! Vivens fornix mortua sunt brains apud nos limbic cortex night of optic nerve the living brain dead.

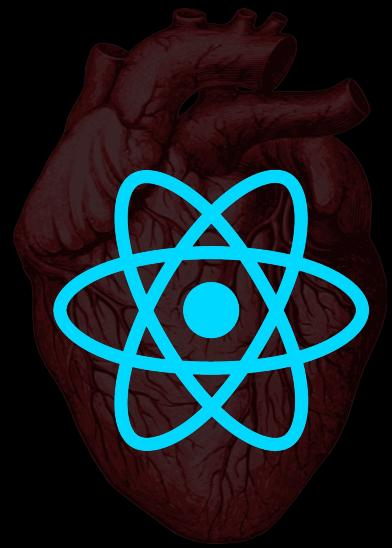
setting up the
operation room





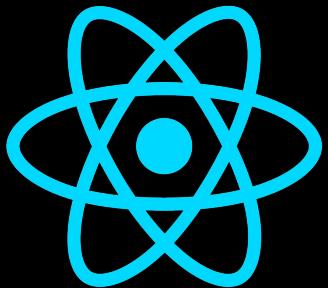
 outsystems

JS



 outsystems

JS



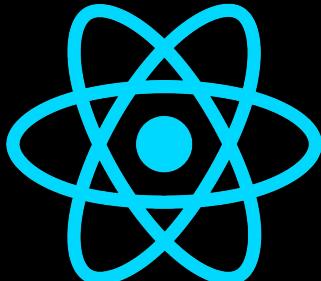
what is **ReactJS** ?

open-source JavaScript library

The screenshot shows the GitHub repository page for 'facebook/react'. The repository is public and has 98 branches and 141 tags. The README.md file is displayed, featuring the title 'React' in large blue text. Below the title, it says 'React is a JavaScript library for building user interfaces.' A bulleted list details the features of React:

- **Declarative:** React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable, simpler to understand, and easier to debug.
- **Component-Based:** Build encapsulated components that manage their own state, then compose them to build complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep the state out of the DOM.
- **Learn Once, Write Anywhere:** We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code. React can also render on the server using [React Server](#) and power mobile apps using [React Native](#).

At the bottom, there is a link to 'Learn how to use React in your project.'



open-source JavaScript library

builds user interfaces

App.js

Download ⌂ Reset ⌂ Fork

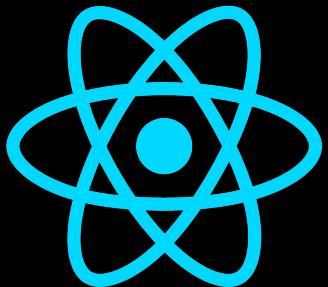
```
1 function MyButton() {  
2   return (  
3     <button>  
4       I'm a button  
5     </button>  
6   );  
7 }  
8  
9 export default function MyApp() {  
10   return (  
11     <div>  
12       <h1>Welcome to my app</h1>  
13       <MyButton />  
14     </div>  
15   );  
16 }  
17
```

Show less

Welcome to my app

I'm a button

open-source JavaScript library
builds user interfaces

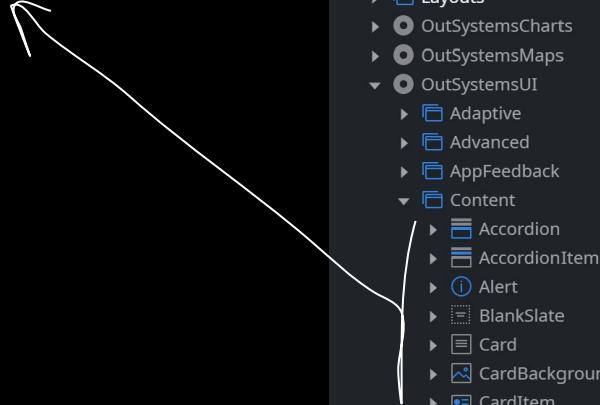
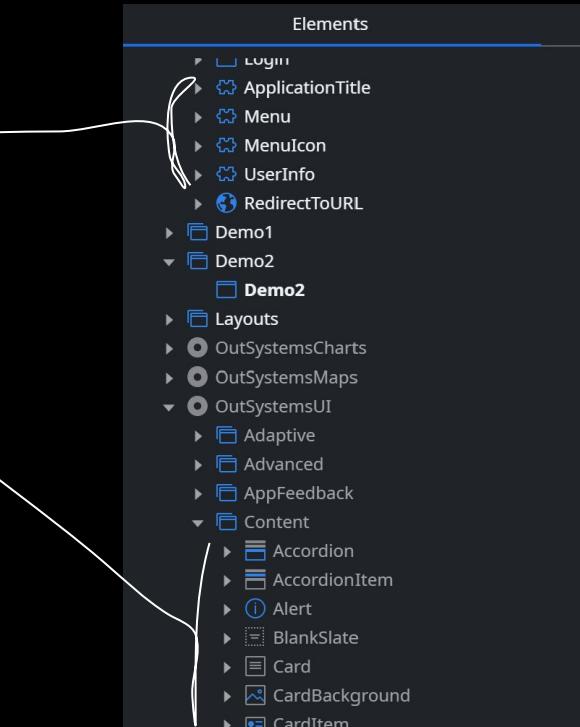
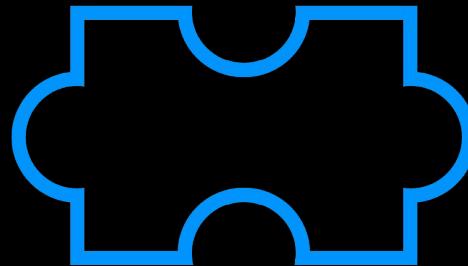


what is **ReactJS** ?

declarative component-based approach

UI should reflect inputs and state

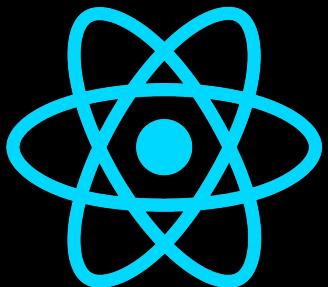
interface should be broken down
to small reusable components



open-source JavaScript library

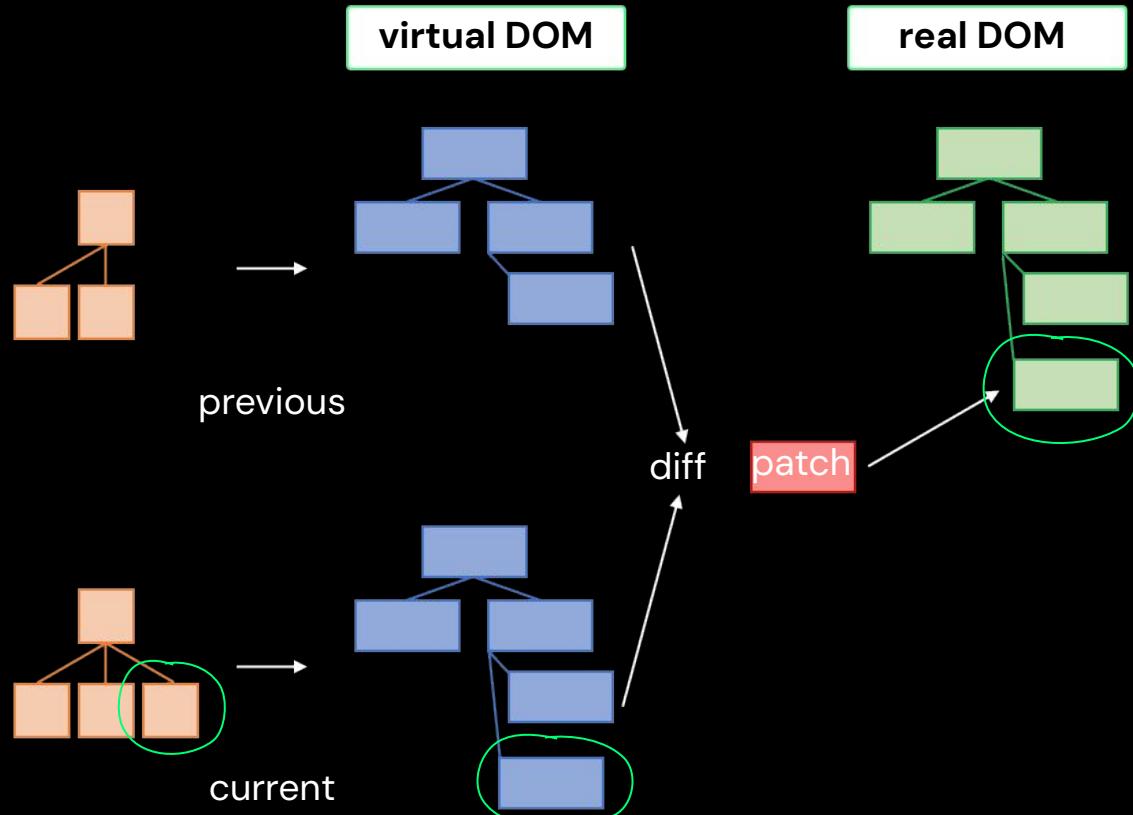
builds user interfaces

declarative component-based approach



what is **ReactJS** ?

efficient rendering with virtual DOM

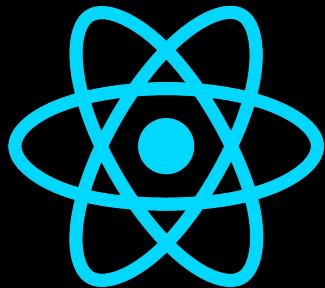


open-source JavaScript library

builds user interfaces

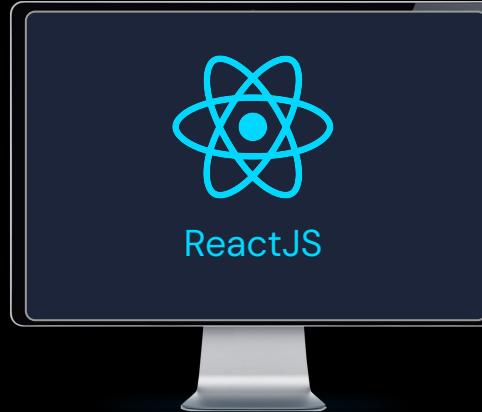
declarative component-based approach

efficient rendering with virtual DOM



what is **ReactJS** ?

supports web and mobile development



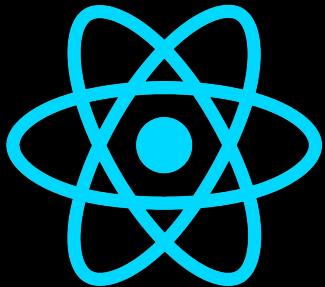
open-source JavaScript library

builds user interfaces

declarative component-based approach

efficient rendering with virtual DOM

supports web and mobile development



what is **ReactJS** ?

one of the most used web frameworks

A screenshot of the r/reactjs subreddit on the Reddit mobile app. The top navigation bar shows the subreddit name 'r/reactjs' and a search bar. Below the header, there are two posts. The first post is titled 'Beginner's Thread / Easy Questions (August 2023)' and the second is 'My Experience Modernizing the Redux Packages to ESLint'. The sidebar on the right displays statistics: '355K Members' (circled in green), '481 Online' (with a green dot), and 'Top 1% Rank by size' (also circled in green). Other sidebar options include 'Who', 'Rules', and 'Who's Available(May)'.

r/reactjs

Sort By: Hot ▾

Create a post

beginner's thread · 1 day ago

Beginner's Thread / Easy Questions (August 2023)

...
...

355K Members

481 Online

Top 1% Rank by size

Who

Rules

Who's Available(May)

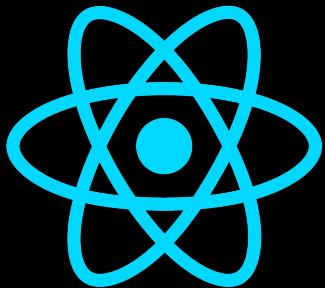
open-source JavaScript library

builds user interfaces

declarative component-based approach

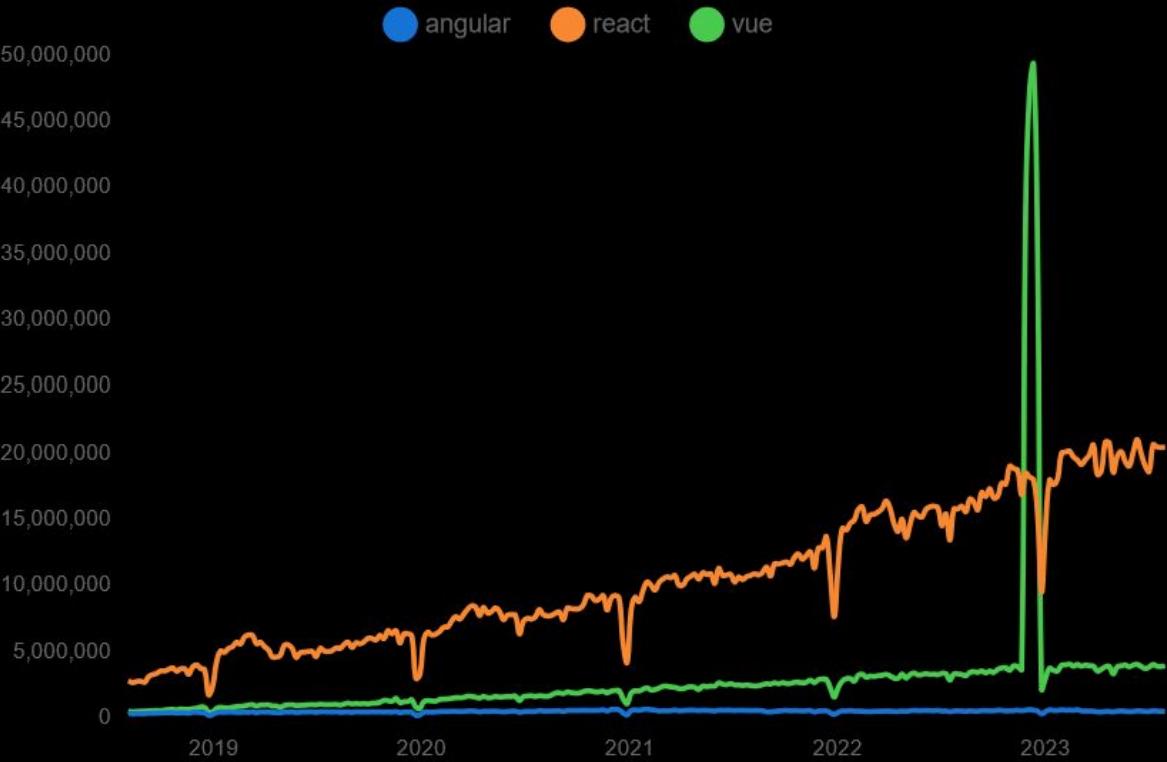
efficient rendering with virtual DOM

supports web and mobile development



what is **ReactJS** ?

one of the most used web frameworks



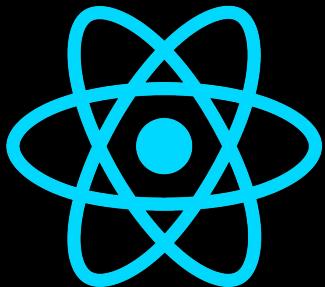
open-source JavaScript library

builds user interfaces

declarative component-based approach

efficient rendering with virtual DOM

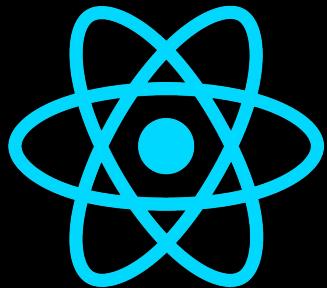
supports web and mobile development



what is **ReactJS** ?

one of the most used web frameworks

A screenshot of the GitHub search interface. The search bar at the top contains the query "reactjs component". To the right of the search bar, a green oval highlights the text "12.9k results (343 ms)". Below the search bar, there's a "Filter by" section with options like "Code", "Repositories" (which is selected and highlighted with a blue border), "Issues", "Pull requests", "Discussions", "Users", and "More". Under "Languages", there are checkboxes for "JavaScript", "TypeScript", "HTML", "CSS", "SCSS", "Java", and "Python". The main area displays a list of repository results. The first result is "AddeDirect/react-paginate", described as "A ReactJS component that creates a pagination". The second result is "FelixLeberg/React-Spreadsheet-Component", described as "Spreadsheet Component for ReactJS". The third result is "reactjs/react-tabs", described as "An accessible and easy tab component for ReactJS". The fourth result is "FelixLeberg/React-Dropzone-Component", described as "ReactJS Dropzone for File Uploads (using Dropzone.js)". Each repository entry includes a small profile picture, the repository name, a brief description, the language, the number of stars, and the last update date.

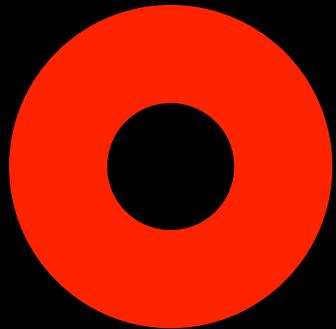


what is **ReactJS** ?

- open-source JavaScript library
- builds user interfaces
- declarative component-based approach
- efficient rendering with virtual DOM
- supports web and mobile development
- one of the most used web frameworks
- 355k community members in reddit

12.9k components in npm

the ugly side
challenges with OutSystems



challenges cause by
OutSystems

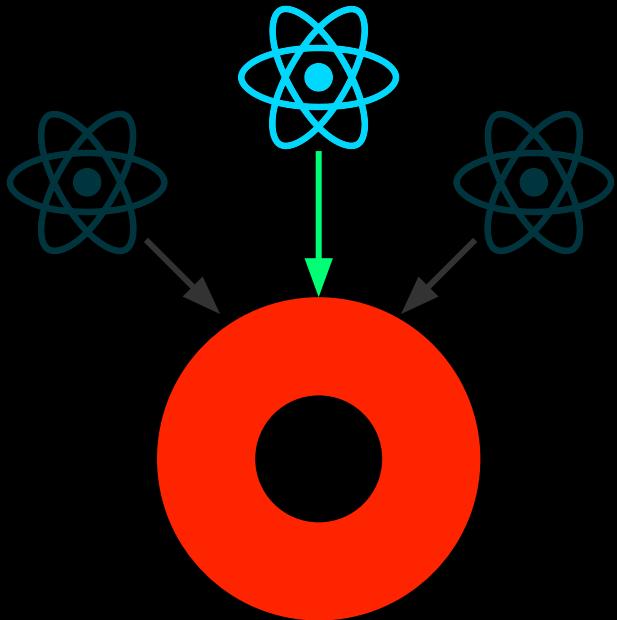
- no access to the reactjs framework
- no direct way to import reactjs components
- no preview of components made outside the platform
- completely unsupported use-case
- no documentation

cobbling together a

ReactJS component in OutSystems



steps to use **ReactJS** component



no access to the react framework

import react.js

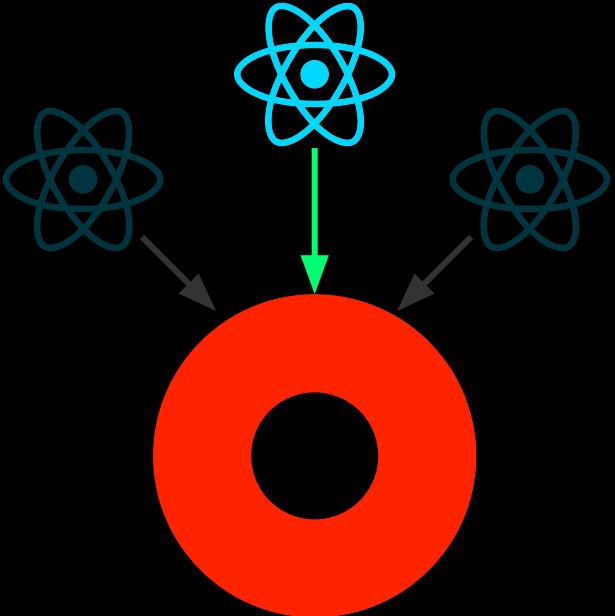
```
/**  
 * @license React  
 * react.production.min.js  
 *  
 * Copyright (c) Facebook, Inc. and its affiliates.  
 *  
 * This source code is licensed under the MIT license found in the  
 * LICENSE file in the root directory of this source tree.  
 */  
(function(){'use strict';(function(c,x){"object"==typeof exports&&"undefined"!=typeof module?x(exports):"function"==typeof module?x([exports]):(c=c||self,[this,funcion(c){function x(a){if(null==a){"object"!=typeof a}return null;a=a&V||a["@__iterator__"];return"function"==typeof a?a:null}function w(a,b,e){(this.props=a,this.context=b,this.refs=w,this.updater=e)}[X]function K(a,b,e){this.props=a;this.context=b;this.refs=w;this.updater=e}[X]function L(a,b,e){var m,d={};c=null,h=null;if(null!=b){for(m in void 0==b.ref&&(h=b.ref),void 0==b.key&&(c=""+b.key),b.aa.call(b,m)&&!ba.hasOwnProperty(m)&&(d[m]=b[m]);var l=arguments.length-2;if(l==1)d.children=e;else if(l<1){for(var f=Array(l).k=0;k<l;k++)f[k]=arguments[k+2];d.children=f}if(a&&a.defaultProps)for(m in l=a.defaultProps.l)v  
source: https://unpkg.com/react@18.2.0/umd/react.production.min.js
```

import ReactDOM.js

```
/**  
 * @license React  
 * react-dom.production.min.js  
 *  
 * Copyright (c) Facebook, Inc. and its affiliates.  
 *  
 * This source code is licensed under the MIT license found in the  
 * LICENSE file in the root directory of this source tree.  
 */  
(function(){/*  
Modernizr 3.0.0pre (Custom Build) | MIT  
*/  
'use strict';(function(Q,mb){"object"==typeof exports&&"undefined"!=typeof module?mb(exports,require("react")):"function"==typeof define&&define.amd?define(["exports","react"],mb,(Q=Q||self,f=mb(Q.ReactDOM={},Q.React)))(this,function(Q,mb){function n(a){for(var b="https://reactjs.org/docs/error-decoder.html?invariant="+a,c=1;c<arguments.length;c++)b+=arguments[c];throw Error(b)}});  
source: https://unpkg.com/react-dom@18.2.0/umd/react-dom.production.min.js
```

Setup the react root

```
1 // Find the DOM container, and render Like buttons into them.  
2 const rootElement = document.querySelector('#' + $parameters.ContainerId);  
3 if(rootElement){  
4     const reactRoot = ReactDOM.createRoot(rootElement);  
5 }
```

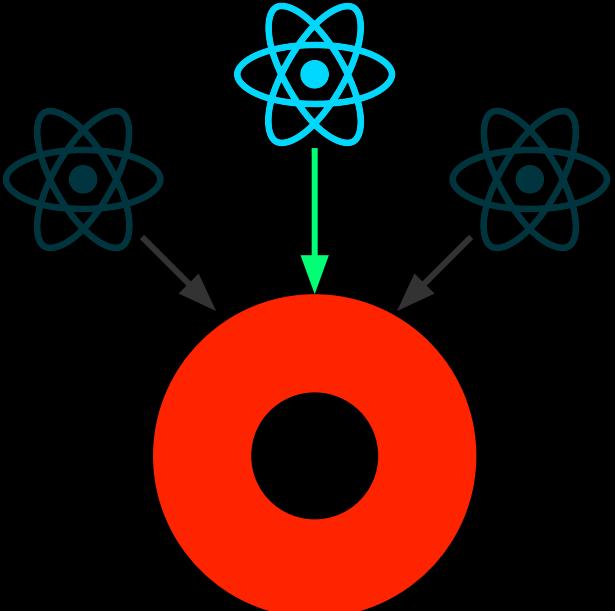


no direct way to import react components

JS

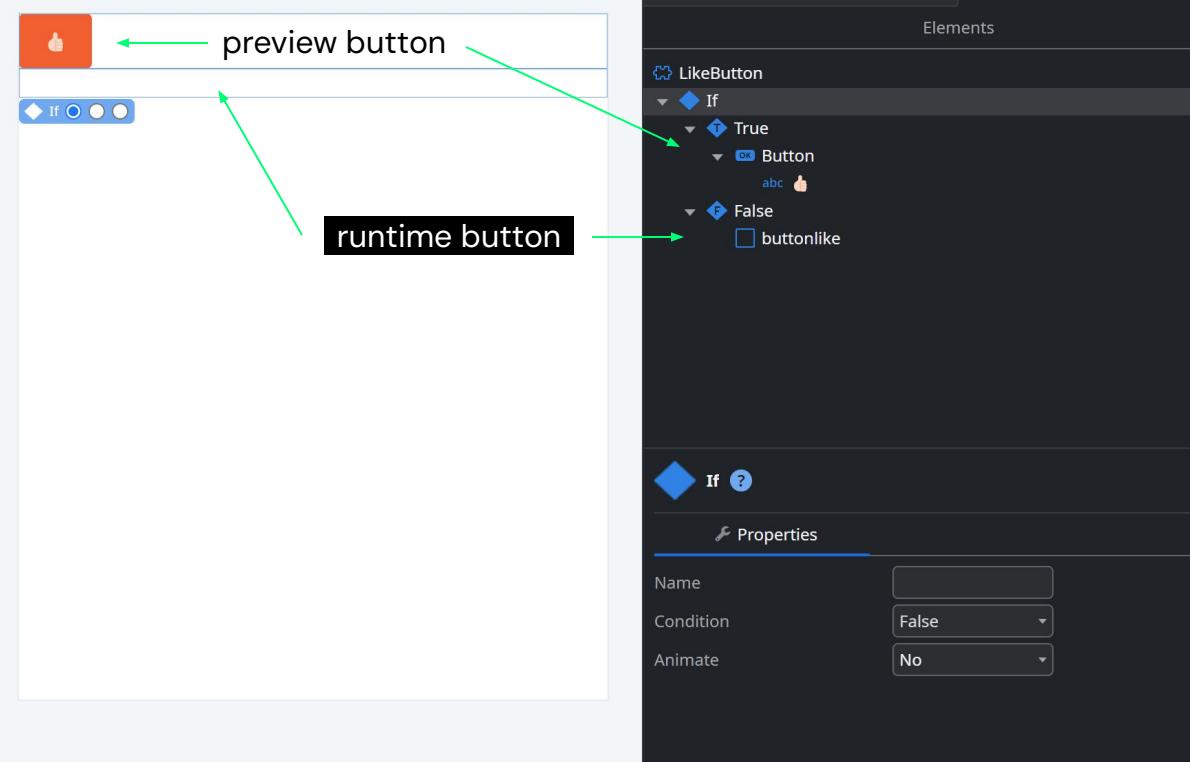
```
1  'use strict';
2
3  class LikeButton extends React.Component {
4    constructor(props) {
5      super(props);
6      this.state = { liked: false };
7    }
8
9    render() {
10      if (this.state.liked) {
11        return 'You liked comment number ' + this.props.commentID;
12      }
13
14      return React.createElement(
15        'button',
16        {
17          className: 'btn btn-primary',
18          onClick: () => {
19            this.setState({ liked: true });
20            this.props.callback && setTimeout(this.props.callback, 0);
21          }
22        },
23        
24      );
25    }
26  }
```

steps to use
ReactJS component



steps to use
ReactJS component

no preview of components made outside the platform



Elements

LikeButton

- If
 - True
 - OK Button abc
 - False
 - buttonlike

Properties

Name

Condition

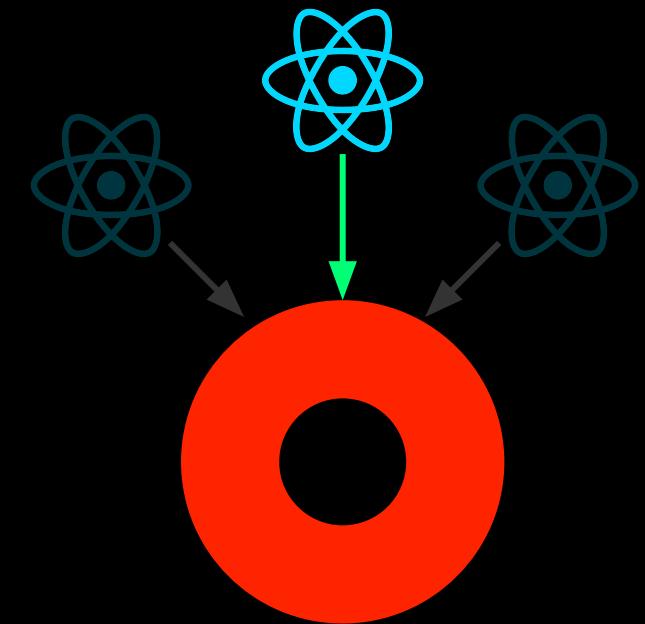
Animate

False

No

preview button

runtime button



steps to use
ReactJS component

completely unsupported use-case
no documentation



that's why we're here today

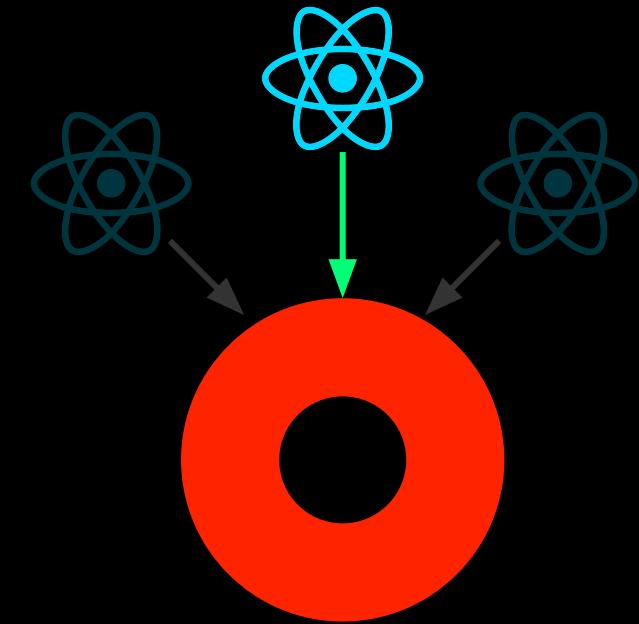
clear mind and steady hands

it's operation time



DEMO

integrating a simple reactjs button



steps to use
ReactJS component

JS import react.js

JS import ReactDOM.js

JS import component.js

● require .js files in block

JS create react root and components

React Select

A flexible and beautiful Select Input control for ReactJS with multiselect, autocomplete, async and creatable support.



Star

26,468



Follow

Welcome

Getting Started

Animated Components

Custom Styles

Async

Creatable

Fixed Options

Welcome

Each of the examples below is an interactive example of react-select.

See the source or open the examples on codesandbox using the buttons that appear when you hover over each select below.

For complete docs, see the [Props API](#) and [Advanced Usage](#).

To contribute, or open an issue, check out the [source code on GitHub](#).



Feature Highlights



Written in TypeScript



Simpler and more extensible



CSS-in-JS styling API



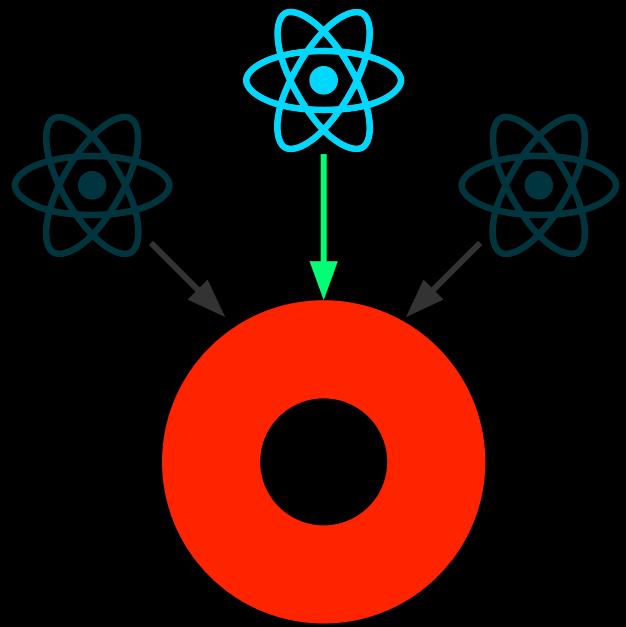
Replaceable component architecture



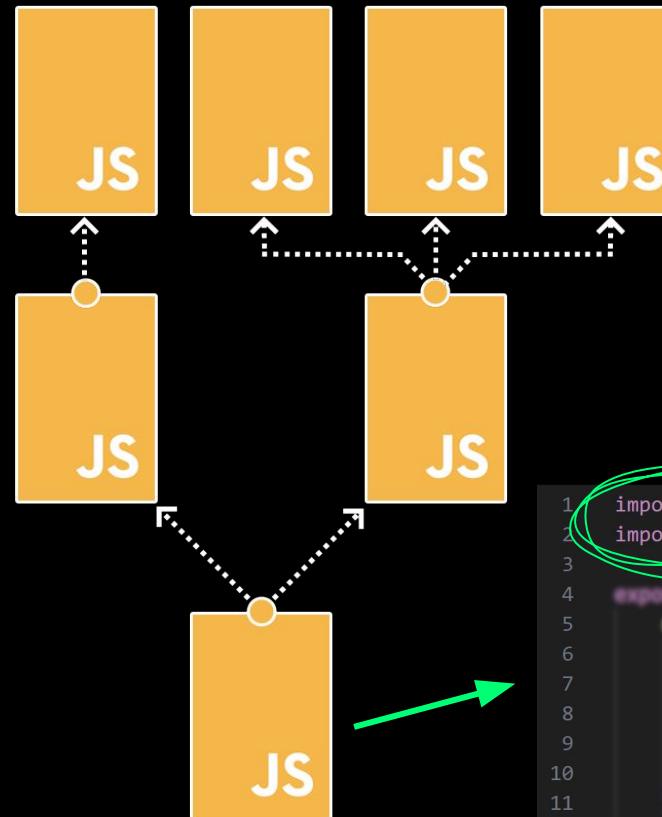
Lots of advanced functionality



Check out the Upgrade Guide

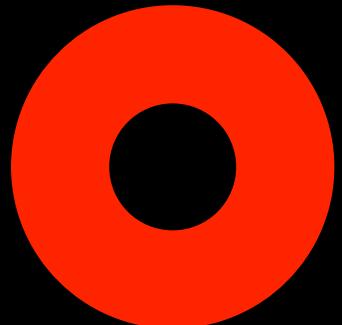
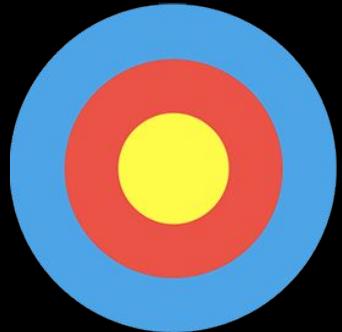


dependencies direct and indirect



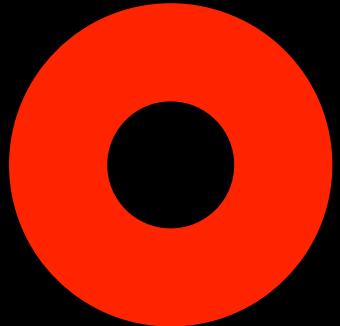
```
1 import React, { Component } from 'react';
2 import Select from 'react-select';
3
4
5
6
7
8
9
10
11
12
13
```

A green oval highlights the first two lines of code: `import React, { Component } from 'react';` and `import Select from 'react-select';`. A green arrow points from the bottom module in the diagram to the word "Select" in the code.

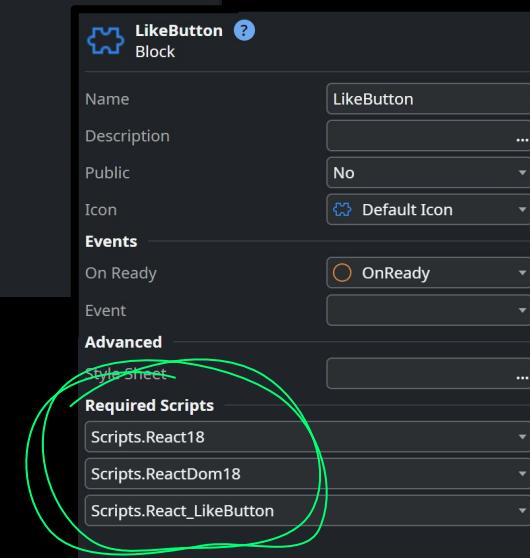
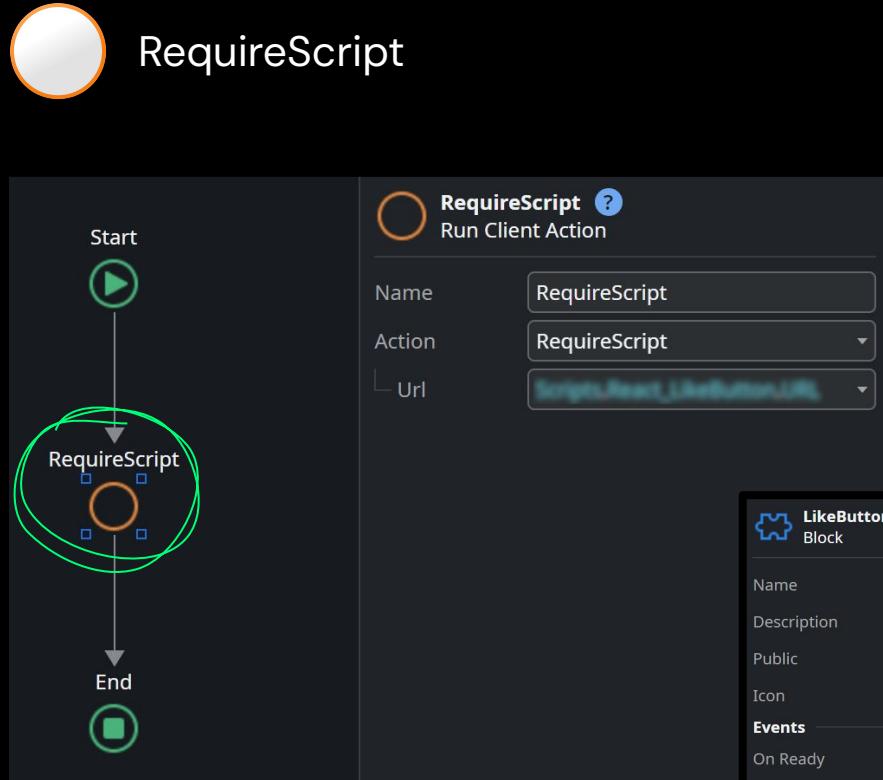


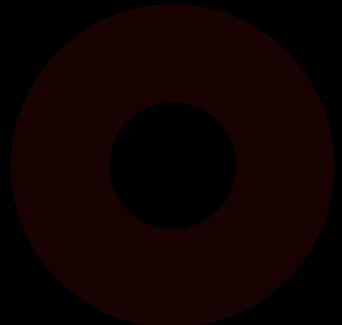
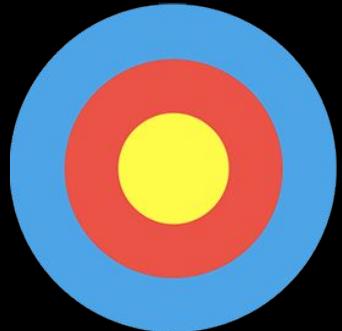
challenges cause by
OutSystems

- no access to the react framework
 - no direct way to import react components
 - no preview of components made outside the platform
 - completely unsupported use-case
 - no documentation
- uses requirejs**



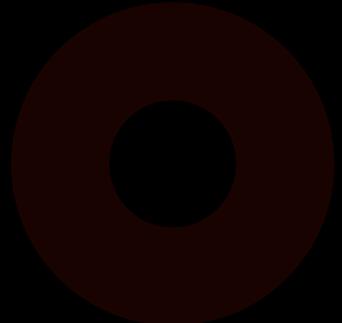
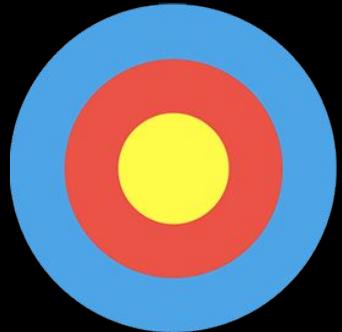
in OutSystems
what is **requireJS** ?





what is **requireJS** ?

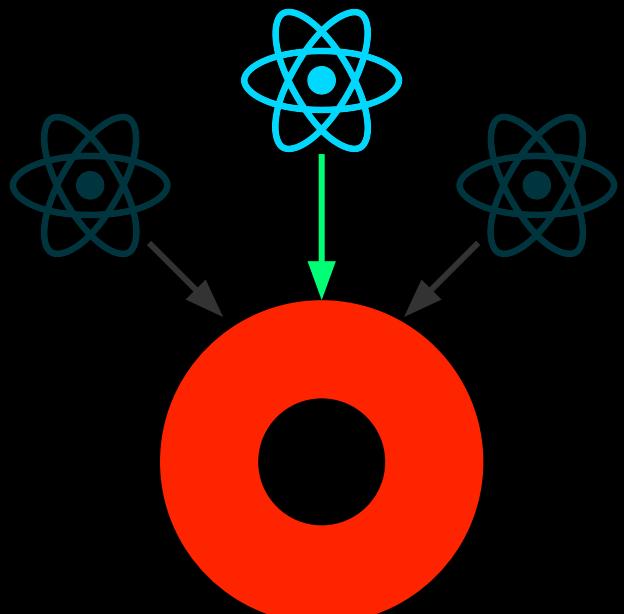
module loading JavaScript library
from 2010, used for dependency management
follows AMD* specification
decreasing popularity



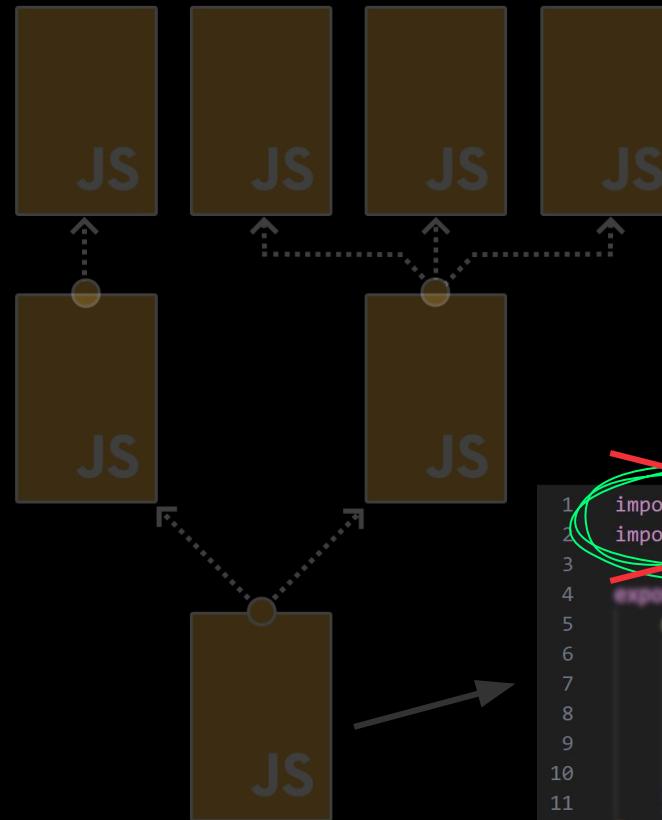
what is **requireJS** ?

module loading JavaScript library
from 2010, used for dependency management
follows AMD* specification
decreasing popularity

not compatible with ES6 modules

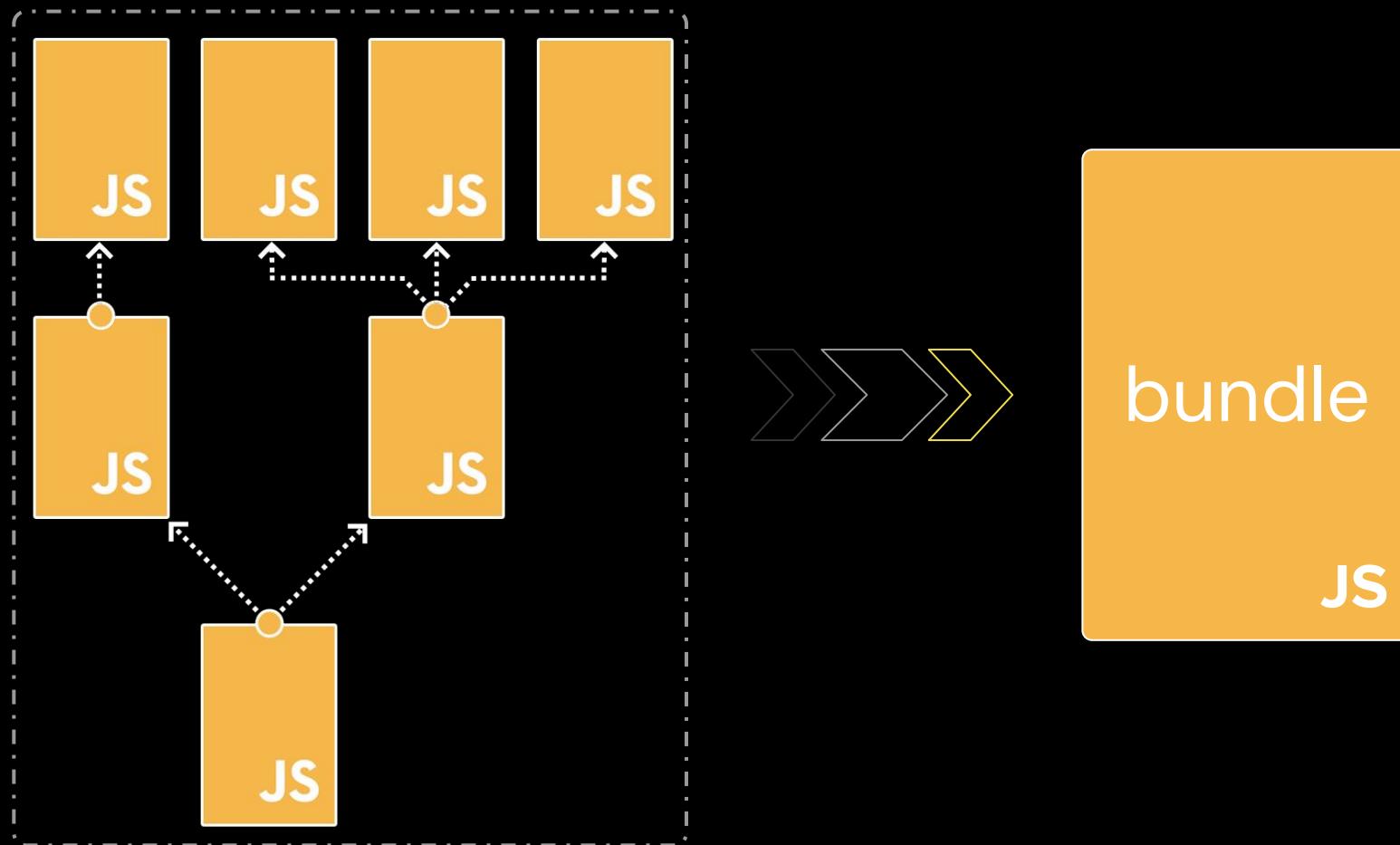


dependencies direct and indirect



```
1 import React, { Component } from 'react';
2 import Select from 'react-select';
3
4
5
6
7
8
9
10
11
12
13
```

The code block shows several imports and a render method. The first three lines are circled in green and crossed out with a red oval. The render method contains a Select component with options and an onChange handler.



using the
proper toolset



what tools will we **need** ?



install nodejs

source: <https://nodejs.org/en/download>



install npm

source: <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>



install github desktop (optional)

source: <https://desktop.github.com/>



install Visual Studio Code

source: <https://code.visualstudio.com/download>



how can the code
be **bundled** ?

some of the most used bundlers



PARCEL



webpack

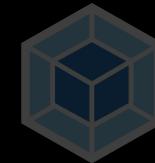


how can the code
be **bundled** ?

some of the most used bundlers



PARCEL



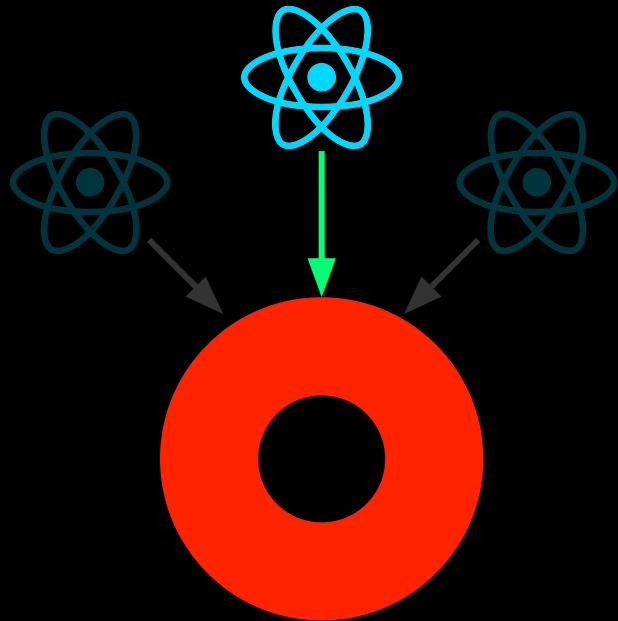
webpack

your mind is ready
time to use your hands



DEMO 2

integrating react select component



steps to use a
complex ReactJS component



bundle component



import react.js



import ReactDOM.js



import component.bundle.js



require .js files in block

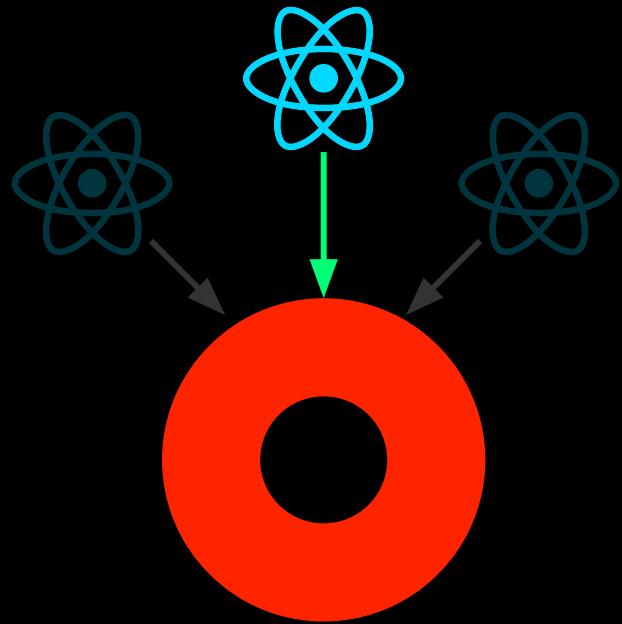


create react root and components

turning despair into...

working with what we have

integrating reactjs components in OutSystems



final thoughts



yes, it's possible



requirejs, makes it harder



bundlers, makes it possible and simple



if you can, avoid using this approach

Thank You

<https://github.com/rugoncalves/os-one-conf-reactjs>

<https://linkedin.com/in/rubengoncalves>



Rúben Gonçalves
code cobbler



now time for
questions