

React custom renderers

Jiayi Hu

Front-end developer & consultant

- <https://github.com/jiayihu>
- https://twitter.com/jiayi_ghu
- jiayi.ghu@gmail.com

Some history



May 2013 - React is open-sourced ¹

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';

var Greeting = React.createClass({
  render: function() {
    return <h1>Hello, {this.props.name}</h1>;
  }
});

ReactDOM.render(<Greeting name="Pordenone"></Greeting>, document.querySelector('.root'));
```

¹ [React: Rethinking best practices](#)

March 2015 - React Native is open-sourced

```
import React, { Component } from 'react';
import { AppRegistry, Text } from 'react-native';

export default class HelloWorldApp extends Component {
  render() {
    return (
      <Text>Hello World!</Text>
    );
  }
}
```

```
AppRegistry.registerComponent('HelloWorldApp', () => HelloWorldApp);
```

Oct 2016 - React VR is announced

```
import React from 'react';
import { AppRegistry, asset, Pano, Text, View } from 'react-vr';

class WelcomeToVR extends React.Component {
  render() {
    return (
      <View>
        <Pano source={asset('chess-world.jpg')}/>
        <Text>hello</Text>
      </View>
    );
  }
};
```

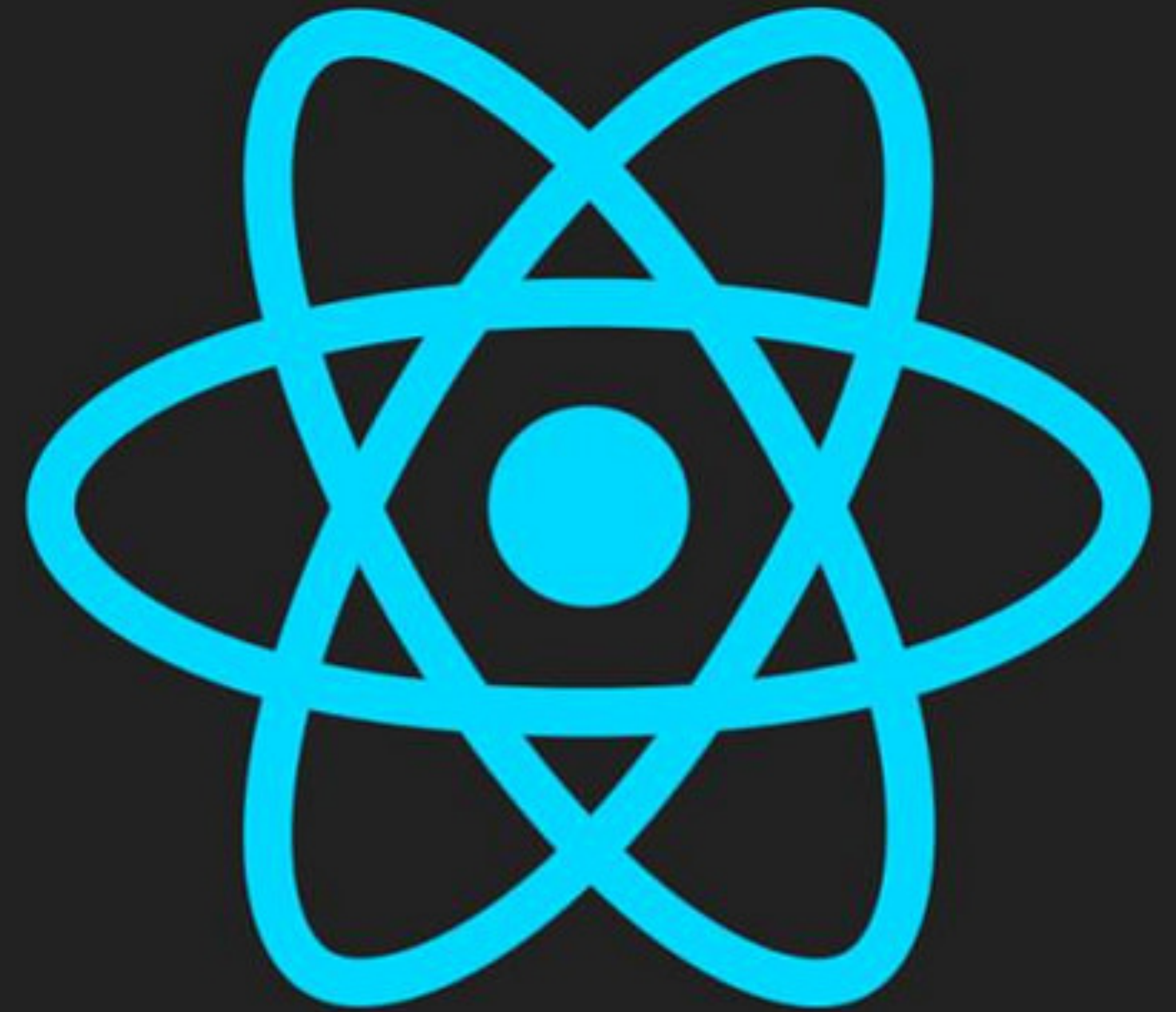
```
AppRegistry.registerComponent('WelcomeToVR', () => WelcomeToVR);
```

April 2017

React Fiber is announced ^{fiber}

^{fiber} [Lin Clark - A Cartoon Intro to Fiber](#)

Sviluppare applicazioni client, server e mobile con React



React Fiber



Reconciliation

<Circle />

<Circle />

<Circle />

<Circle />

<Circle />



<Circle />

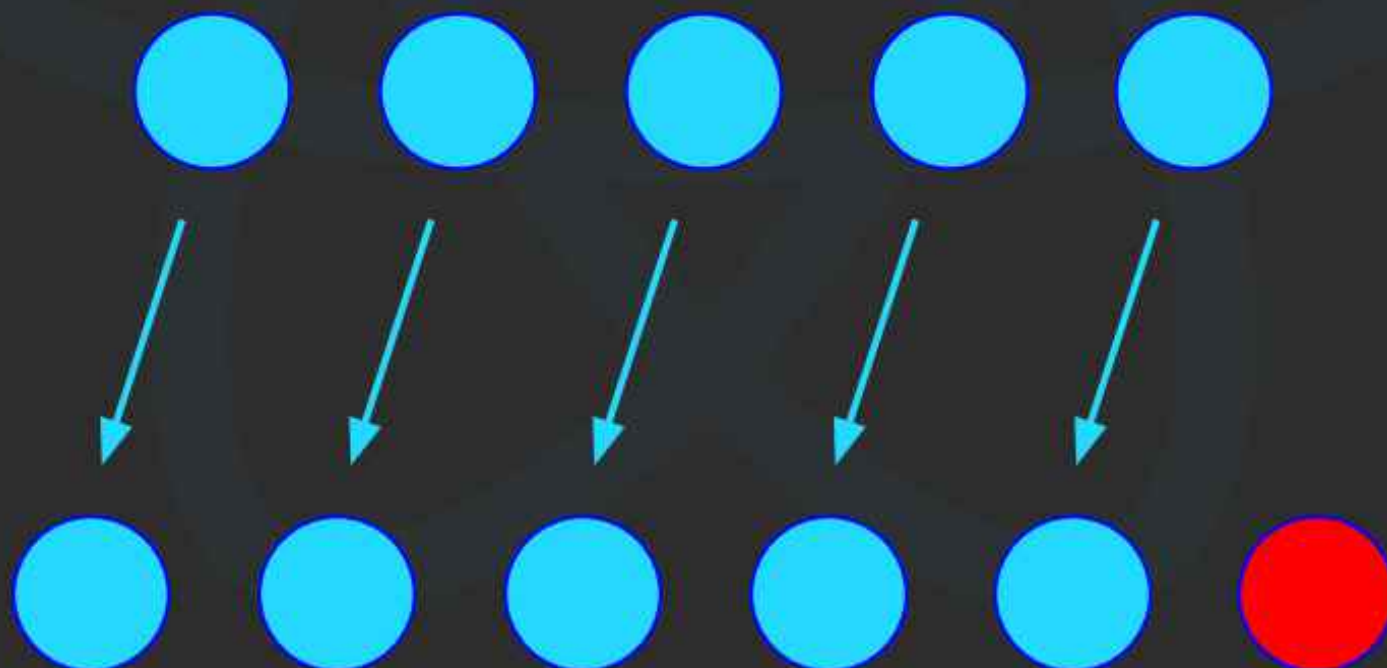
<Circle />

<Circle />

<Circle />

<Circle />

<Circle />



Nov 2017

React custom renderers in Pordenone

RECONCILER VS RENDERER

- Virtual DOM
- Compares differing trees
- Computes changes to send to renderer

- Updates the app's UI
- Devices can have different renderers while sharing a reconciler

ReactSpeech

```
import React from 'react';  
import { ReactSpeech } from './renderer/speech-renderer';  
  
ReactSpeech.render([  
  <alice key={1}>Ciao Pordenone</alice>,  
  <luca key={2}>Benvenuti al mio talk di React</luca>,  
]);
```




Woah!

Terminology

1. React **Component**
2. React **Element**
3. ReactFiber **Instance**

React Component

```
class Greeting extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

React Element

```
const element = <h1>Hello, world</h1>;
```

```
// Compiled
```

```
const element = React.createElement(  
  'h1',  
  {},  
  ['Hello world']  
);
```


ReactFiber Instance

```
createInstance(type, props) {  
  const instance = document.createElement('h1');  
  instance.textContent = 'Hello world';  
  
  return instance;  
},
```


Pre-React 16 Fiber

```
import ReactInjection from 'react/lib/ReactInjection';

function inject() {
  ReactInjection.NativeComponent.injectGenericComponentClass(
    MyCustomComponentClass
  );
}
```

Post-React 16 Fiber

```
import Reconciler from 'react-reconciler';  
  
const MyRenderer = Reconciler(HostConfig);
```

Warning! Unstable API

react-reconciler@0.6.0



```

{
  appendInitialChild(parentInstance, child) {},

  createInstance(type, props, internalInstanceHandle) {},

  createTextInstance(text, rootContainerInstance, internalInstanceHandle) {},

  finalizeInitialChildren(instance, type, props) {},

  getPublicInstance(inst) {
    return inst;
  },

  prepareForCommit() {},
  resetAfterCommit() {},

  prepareUpdate(instance, type, oldProps, newProps) {},

  getRootHostContext(rootInstance) {},
  getChildHostContext() {},

  resetTextContent(instance) {},
  shouldSetTextContent(type, props) {},

  now: () => {},

  useSyncScheduling: true,

  mutation: {
    appendChild(parentInstance, child) {},

    appendChildToContainer(parentInstance, child) {},

    removeChild(parentInstance, child) {},

    removeChildFromContainer(parentInstance, child) {},

    insertBefore(parentInstance, child, beforeChild) {},

    commitUpdate(instance, updatePayload, type, oldProps, newProps) {},

    commitMount(instance, updatePayload, type, oldProps, newProps) {},

    commitTextUpdate(textInstance, oldText, newText) {},
  },
}

```

Create instance

```
createInstance(type, props, internalInstanceHandle) {  
  return document.createElement(type);
```

```
  // Do something with the props  
}
```

```
createTextInstance(text, rootContainerInstance, internalInstanceHandle) {  
  return text;  
}
```

Scheduling

```
{  
  now: ReactDOMFrameScheduling.now,  
  useSyncScheduling: true,  
  ...  
}
```

Mutation

```
mutation: {  
  appendChild(parentInstance, child) {},  
  appendChildToContainer(parentInstance, child) {},  
  
  removeChild(parentInstance, child) {},  
  removeChildFromContainer(parentInstance, child) {},  
  
  insertBefore(parentInstance, child, beforeChild) {},  
  
  commitUpdate(instance, updatePayload, type, oldProps, newProps) {},  
  commitMount(instance, updatePayload, type, oldProps, newProps) {},  
  commitTextUpdate(textInstance, oldText, newText) {},  
},
```

ReactSpeech

```
import Reconciler from 'react-reconciler';

const SpeechRenderer = Reconciler(HostConfig);

export const ReactSpeech = {
  render(element, callback) {
    const root = SpeechRenderer.createContainer({});
    SpeechRenderer.updateContainer(element, root, null, callback);
  },
};
```


Usage

```
import React from 'react';
import { ReactSpeech } from './renderer/speech-renderer';

ReactSpeech.render([
  <alice key={1}>Ciao Pordenone</alice>,
  <luca key={2}>Benvenuti al mio talk di React</luca>,
]);
```

Awesome custom renderers

- [react-sketchapp](#)
- [react-pdf](#) & [react-redocx](#)
- [Ink](#)
- [react-tv](#)
- [react-hardware](#)

For dauntless people

Fevr 29 Nov presso AQuest: Can't live if livin' is
without rendering

Resources

- ReactSpeech: github.com/jiayihu/experiments
- Slides: Dove le hai scaricate scemo