

REACT CODE GENERATOR FROM TREE

Open Source SW Project Team 1
Hercules Health Gym

CONTENTS

Overview

Necessity

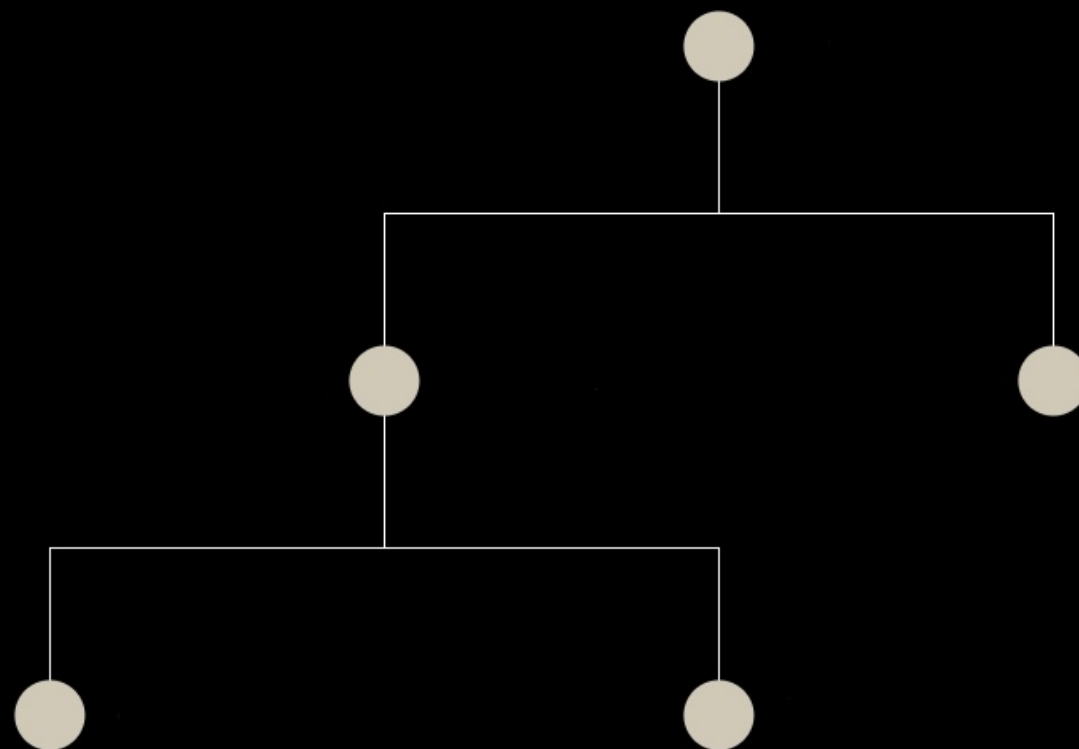
Differentiation

Development Plan

Open Source SW for Reference or Use

OVERVIEW

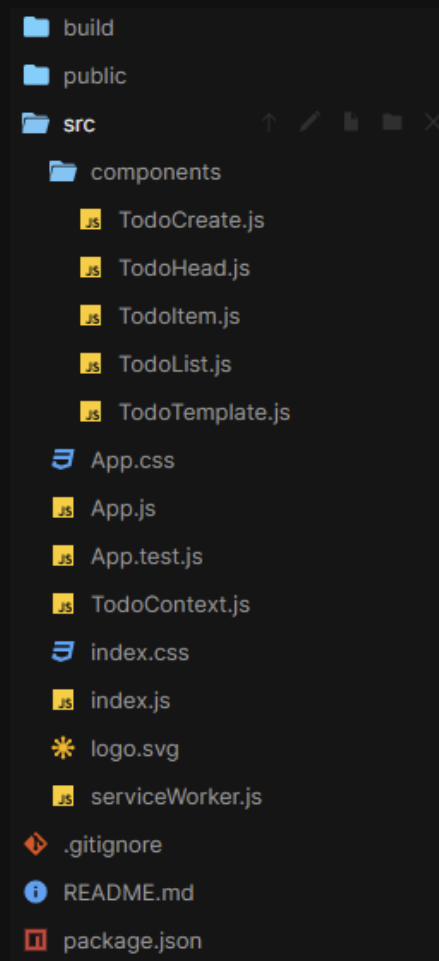
FROM TREE



TO CODE

```
1
2 import React from 'react';
3 import TodoTemplate from './components/TodoTemplate';
4 import TodoHead from './components/TodoHead';
5 import TodoList from './components/TodoList';
6 import TodoCreate from './components/TodoCreate';
7 import { TodoProvider } from './TodoContext';
8
9
10 function App() {
11   return (
12     <TodoProvider>
13       <TodoTemplate>
14         <TodoHead />
15         <TodoList />
16         <TodoCreate />
17       </TodoTemplate>
18     </TodoProvider>
19   );
20 }
21
22 export default App;
23
```

TO PROJECT DIRECTORY STRUCTURE



TO WITH EMPTY COMPONENT

```
1  import React from "react";
2
3  function TodoItem() {
4      return <></>;
5  }
6
7  export default TodoItem;
8
```

NECESSITY

<Import Statements>

...

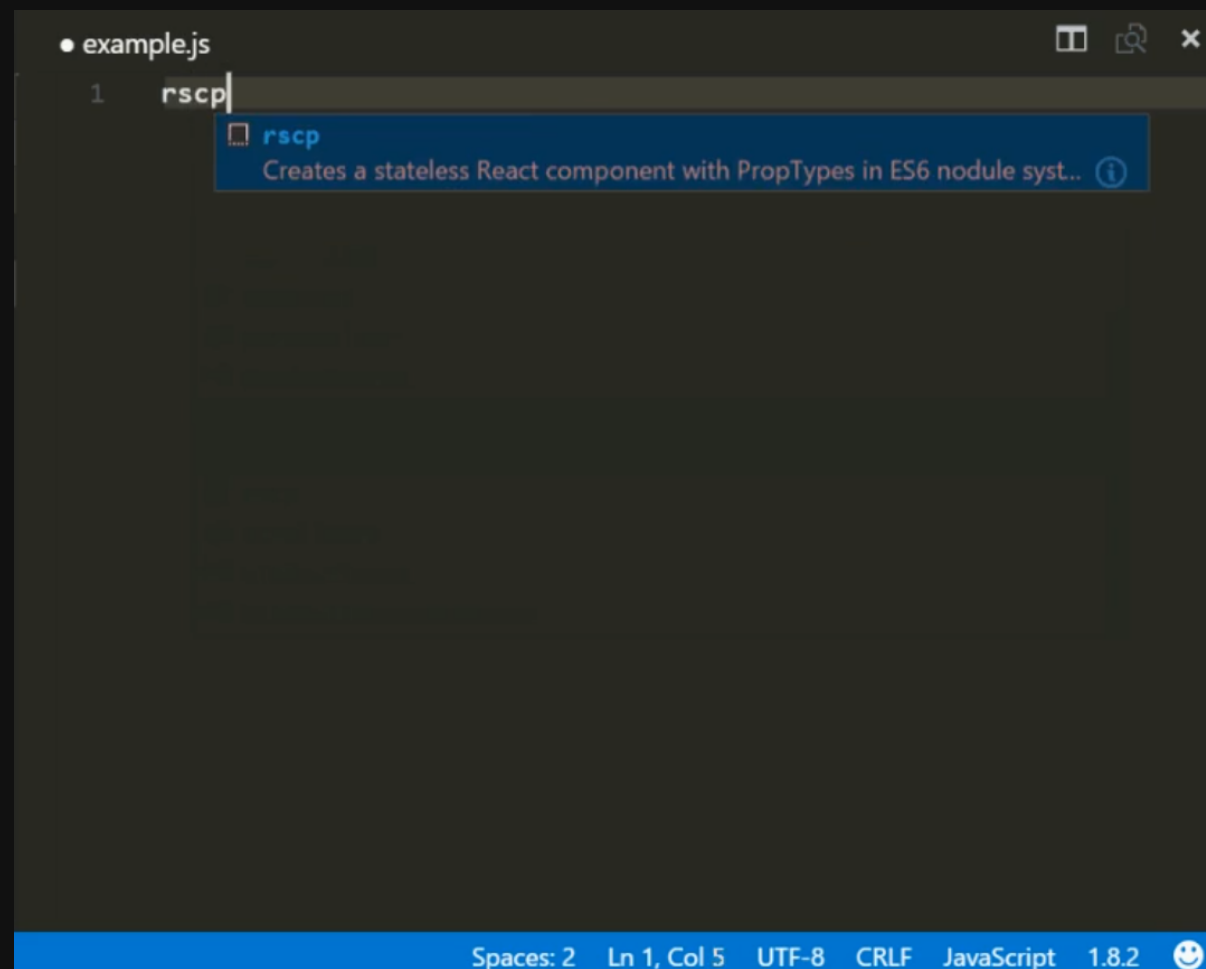
<Components which returns jsx>

...

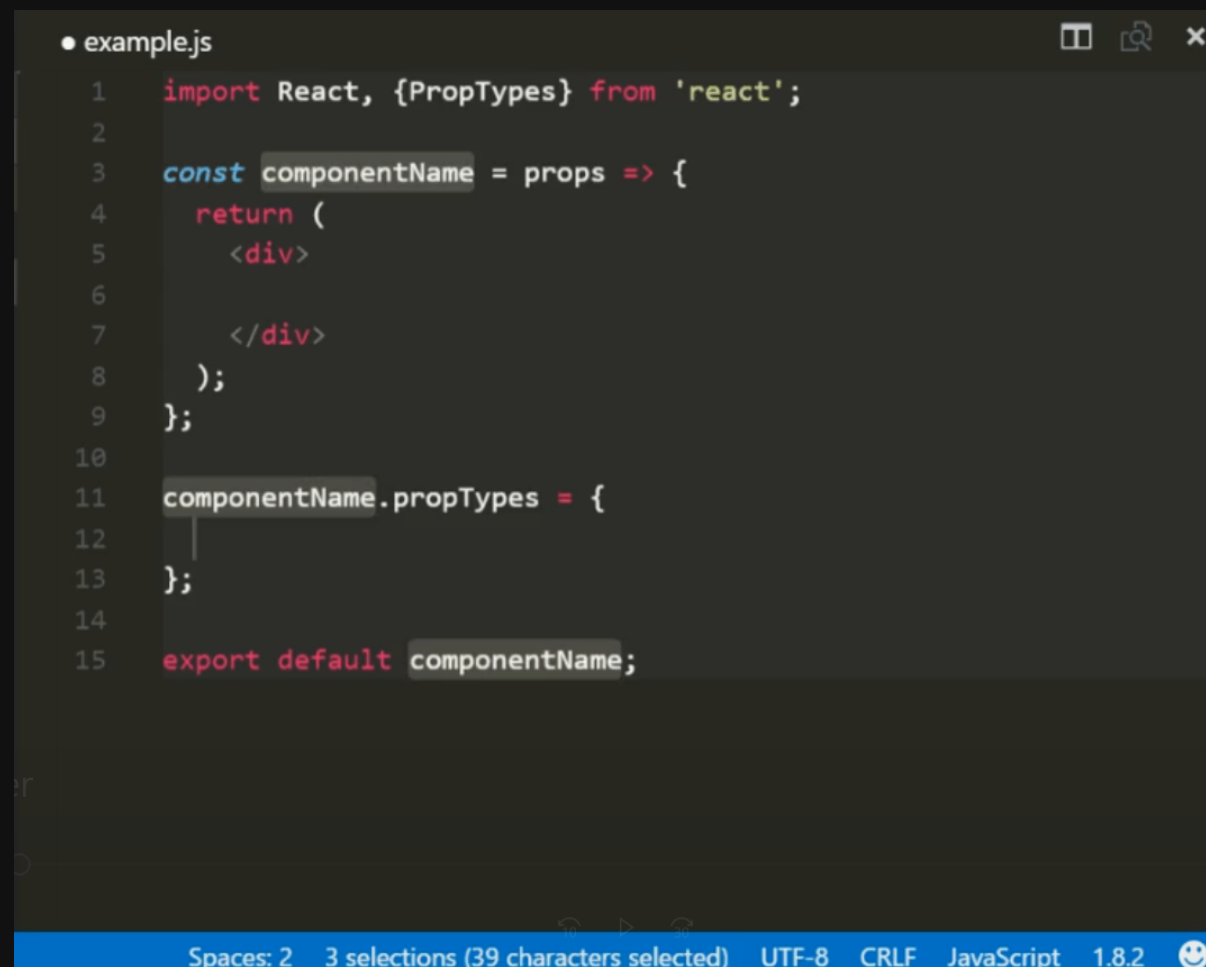
<Export Statements>

LOTS OF REPEATED WORK

LOTS OF CODE SNIPPETS



LOTS OF CODE SNIPPETS



The image shows a code editor window titled "example.js". The code is a JavaScript snippet for a React component. It starts with an import statement for React and PropTypes. Then, it defines a constant component named "componentName" which is a function that takes "props" and returns a JSX element. The JSX element consists of a single "div" tag. After the function definition, it sets the "propTypes" for "componentName" and finally exports "componentName" as the default export. The code is syntax-highlighted, with keywords in blue, strings in red, and identifiers in white. The editor has a dark theme and a status bar at the bottom showing "Spaces: 2", "3 selections (39 characters selected)", "UTF-8", "CRLF", "JavaScript", and "1.8.2".

```
1  import React, {PropTypes} from 'react';
2
3  const componentName = props => {
4    return (
5      <div>
6
7      </div>
8    );
9  };
10
11  componentName.propTypes = {
12
13  };
14
15  export default componentName;
```

WHAT IF?

WHAT IF?

THERE'S LOTS OF COMPONENTS

WHAT IF?

THERE'S LOTS OF COMPONENTS

THAT DEPENDS ON EACH OTHER?

**SIMPLIFY
THOSE PROCESSES
WITH OUR SW**

SIMPLIFY THOSE PROCESSES

Start to code important logic

SIMPLIFY THOSE PROCESSES

Start to code important logic
from your structure right away!

DIFFERENTIATION ?

DIFFERENTIATION ?

NONE OF OPEN SOURCE TOOL FOR THIS

DIFFERENTIATION ?

ONLY FOR OPPOSING ACTION

DIFFERENTIATION ?

ONLY FOR OPPOSING ACTION

The screenshot displays the ReactION.js development environment. On the left, the Explorer panel shows the project structure with files like `App.js`, `EachComponent.js`, and `TotalSum.js`. The central editor shows the `App.js` code, which defines a class `App` extending `Component`. The code includes state management for `componentsArray` and `totalSum`, and methods for adding and removing components. The right panel shows the HTML Preview, which displays the ReactION logo, the text "Total Sum: 9", a button "Click to add a component!", and three component slots labeled A, B, and C. The bottom panel shows the Virtual DOM Tree, which illustrates the component hierarchy. The tree shows the `App` component as the root, with children `TotalSum` and `EachComponent`. The `EachComponent` node is further expanded, showing its children `AddComponent` and `EachComponent`. The status bar at the bottom indicates the current file is `App.js`, line 13, column 1, with 6 selected characters.

ReactION-js/ReactION

DIFFERENTIATION ?

VS NO-CODE TOOLS

DIFFERENTIATION ?

NO-CODE TOOLS

For **Non**-developers

DIFFERENTIATION ?

OUR SW

For Developers

DIFFERENTIATION ?

OUR SW

For Developers
by building scaffolds
(e.g. Project structure, basic code snippets)

DEVELOPMENT PLAN

DEVELOPMENT PLAN

LANGUAGE

TypeScript / React

DEVELOPMENT PLAN

COLLABORATION

Git

DEVELOPMENT PLAN

COLLABORATION

GitHub

DEVELOPMENT PLAN

COLLABORATION

Git Flow

DEVELOPMENT PLAN

COLLABORATION

Agile

DEVELOPMENT PLAN

COLLABORATION

Agile Khanban

DEVELOPMENT PLAN

MINIMUM VIABLE PRODUCT

DEVELOPMENT SCOPE

VSCode Extension

DEVELOPMENT PLAN

MINIMUM VIABLE PRODUCT

DEVELOPMENT SCOPE

Focus on Basic Feature

DEVELOPMENT PLAN

MINIMUM VIABLE PRODUCT

DEVELOPMENT SCOPE

Just From Tree To Code

DEVELOPMENT PLAN

MINIMUM VIABLE PRODUCT

DEVELOPMENT SCOPE

Just From Tree To Code
Only support building basic component structure

DEVELOPMENT PLAN

DISTRIBUTING ROLES

DEVELOPMENT PLAN

DISTRIBUTING ROLES

PROJECT MANAGEMENT

임동영

DEVELOPMENT PLAN

DISTRIBUTING ROLES

UI DEVELOPMENT

정용준 / 조원희

DEVELOPMENT PLAN

DISTRIBUTING ROLES

LOGIC DEVELOPMENT

문법식 / 박준수

OPEN SOURCE SW
FOR REFERENCE OR USE

OPEN SOURCE SW FOR REFERENCE OR USE

< REACT >

UI IMPLEMENTATION

<https://github.com/facebook/react>

**OPEN SOURCE SW
FOR REFERENCE OR USE**

< REACT D3 TREE >

IMPLEMENT TREE USING REACT

<https://github.com/bkrem/react-d3-tree>

OPEN SOURCE SW FOR REFERENCE OR USE

< GENERATE REACT CODE >

REFERENCE CODE GENERATING MODULE

<https://github.com/JPStrydom/Generate-React-Code>

OPEN SOURCE SW FOR REFERENCE OR USE

< REACTION >

REFERENCE THE IDEA OF PROJECT

which analyze react project and generate tree

<https://github.com/ReactIOn-js/ReactIOn>

LICENSE

MIT LICENSE

FIN

Q & A