

734 lines (554 loc) · 17.4 KB



Snippets info

Fvery space inside { } and () means that this is pushed into next line ·) \$ represent vscode-react-javascript-snippets / docs / Snippets.md



I.E. tsrcc

• Hooks from official docs are added with hook name as prefix.

Prefix	Method
imp→	import moduleName from 'module'
imn→	import 'module'
imd→	<pre>import { destructuredModule } from 'module'</pre>

Prefix	Method
ime→	import * as alias from 'module'
ima→	<pre>import { originalName as aliasName} from 'module'</pre>
exp→	export default moduleName
exd→	<pre>export { destructuredModule } from 'module'</pre>
exa→	export { originalName as aliasName} from 'module'
enf→	<pre>export const functionName = (params) => { }</pre>
edf→	export default (params) => { }
ednf→	<pre>export default function functionName(params) { }</pre>
met→	methodName = (params) => { }
fre→	<pre>arrayName.forEach(element => { }</pre>
fof→	for(let itemName of objectName { }
fin→	for(let itemName in objectName { }
anfn→	(params) => { }
nfn→	<pre>const functionName = (params) => { }</pre>
dob→	<pre>const {propName} = objectToDescruct</pre>
dar→	const [propName] = arrayToDescruct
sti→	<pre>setInterval(() => { }, intervalTime</pre>
sto→	<pre>setTimeout(() => { }, delayTime</pre>
prom→	return new Promise((resolve, reject) => { }
cmmb→	comment block
cp→	<pre>const { } = this.props</pre>
CS→	<pre>const { } = this.state</pre>

Prefix	Method
imr→	import React from 'react'
imrd→	import ReactDOM from 'react-dom'
imrc→	<pre>import React, { Component } from 'react'</pre>
imrpc→	<pre>import React, { PureComponent } from 'react'</pre>
imrm→	<pre>import React, { memo } from 'react'</pre>
imrr→	<pre>import { BrowserRouter as Router, Route, NavLink} from 'react- router-dom'</pre>
imbr→	<pre>import { BrowserRouter as Router} from 'react-router-dom'</pre>
imbrc→	<pre>import { Route, Switch, NavLink, Link } from react-router-dom'</pre>
imbrr→	<pre>import { Route } from 'react-router-dom'</pre>
imbrs→	<pre>import { Switch } from 'react-router-dom'</pre>
imbrl→	<pre>import { Link } from 'react-router-dom'</pre>
imbrnl→	<pre>import { NavLink } from 'react-router-dom'</pre>
imrs→	<pre>import React, { useState } from 'react'</pre>
imrse→	<pre>import React, { useState, useEffect } from 'react'</pre>
redux→	<pre>import { connect } from 'react-redux'</pre>
est→	<pre>this.state = { }</pre>
cdm→	<pre>componentDidMount = () => { }</pre>
scu→	<pre>shouldComponentUpdate = (nextProps, nextState) => { }</pre>
cdup→	<pre>componentDidUpdate = (prevProps, prevState) => { }</pre>
cwun→	<pre>componentWillUnmount = () => { }</pre>
gdsfp→	<pre>static getDerivedStateFromProps(nextProps, prevState) { }</pre>
gsbu→	<pre>getSnapshotBeforeUpdate = (prevProps, prevState) => { }</pre>
sst→	this.setState({ })
ssf→	this.setState((state, props) => return { })

Prefix	Method
props→	this.props.propName
state→	this.state.stateName
rcontext→	<pre>const \$1 = React.createContext()</pre>
cref→	<pre>this.\$1Ref = React.createRef()</pre>
fref→	<pre>const ref = React.createRef()</pre>
bnd→	<pre>this.methodName = this.methodName.bind(this)</pre>

Prefix	Method
imrn→	<pre>import { \$1 } from 'react-native'</pre>
rnstyle→	<pre>const styles = StyleSheet.create({})</pre>

Prefix	Method
rxaction→	redux action template
rxconst→	export const \$1 = '\$1'
rxreducer→	redux reducer template
rxselect→	redux selector template
rxslice→	redux slice template

∂ PropTypes

Prefix	Method
pta→	PropTypes.array
ptar→	PropTypes.array.isRequired
ptb→	PropTypes.bool
ptbr→	PropTypes.bool.isRequired

Prefix	Method
ptf→	PropTypes.func
ptfr→	PropTypes.func.isRequired
ptn→	PropTypes.number
ptnr→	PropTypes.number.isRequired
pto→	PropTypes.object
ptor→	PropTypes.object.isRequired
pts→	PropTypes.string
ptsr→	PropTypes.string.isRequired
ptnd→	PropTypes.node
ptndr→	PropTypes.node.isRequired
ptel→	PropTypes.element
ptelr→	PropTypes.element.isRequired
pti→	PropTypes.instanceOf(name)
ptir→	PropTypes.instanceOf(name).isRequired
pte→	PropTypes.oneOf([name])
pter→	PropTypes.oneOf([name]).isRequired
ptet→	PropTypes.oneOfType([name])
ptetr→	PropTypes.oneOfType([name]).isRequired
ptao→	PropTypes.arrayOf(name)
ptaor→	PropTypes.arrayOf(name).isRequired
ptoo→	PropTypes.objectOf(name)
ptoor→	PropTypes.objectOf(name).isRequired
ptsh→	PropTypes.shape({ })
ptshr→	PropTypes.shape({ }).isRequired
ptany→	PropTypes.any

Prefix	Method
ptypes→	<pre>static propTypes = {}</pre>

∂ Console

Prefix	Method
clg→	console.log(object)
clo→	<pre>console.log(`object`, object)</pre>
clj→	<pre>console.log(`object`, JSON.stringify(object, null, 2))</pre>
ctm→	<pre>console.time(`timeId`)</pre>
cte→	<pre>console.timeEnd(`timeId`)</pre>
cas→	console.assert(expression,object)
ccl→	console.clear()
cco→	console.count(label)
cdi→	console.dir
cer→	console.error(object)
cgr→	console.group(label)
cge→	console.groupEnd()
ctr→	console.trace(object)
cwa→	console.warn
cin→	console.info

⊘ React Components

∂ rcc

```
import React, { Component } from 'react'
export default class FileName extends Component {
  render() {
    return <div>$2</div>
```

Q

```
}
     }
♂ rce
                                                                                   Q
     import React, { Component } from 'react'
     export class FileName extends Component {
       render() {
         return <div>$2</div>
       }
     }
     export default $1
♂ rcep
                                                                                   Q
     import React, { Component } from 'react'
     import PropTypes from 'prop-types'
     export class FileName extends Component {
       static propTypes = {}
       render() {
         return <div>$2</div>
       }
     }
     export default $1
∂ rpc
                                                                                   Q
     import React, { PureComponent } from 'react'
     export default class FileName extends PureComponent {
       render() {
         return <div>$2</div>
       }
     }
```

```
ſĠ
     import React, { PureComponent } from 'react'
     import PropTypes from 'prop-types'
     export default class FileName extends PureComponent {
       static propTypes = {}
       render() {
         return <div>$2</div>
       }
     }
ſĊ
     import React, { PureComponent } from 'react'
     import PropTypes from 'prop-types'
     export class FileName extends PureComponent {
       static propTypes = {}
       render() {
         return <div>$2</div>
       }
     }
     export default FileName

  rccp

                                                                                     ſĊ
     import React, { Component } from 'react'
     import PropTypes from 'prop-types'
     export default class FileName extends Component {
       static propTypes = {
         $2: $3,
       }
       render() {
         return <div>$4</div>
       }
     }
♂ rfcp
```

```
Q
     import React from 'react'
     import PropTypes from 'prop-types'
     function $1(props) {
       return <div>$0</div>
     }
     $1.propTypes = {}
     export default $1
∂ rfc
                                                                                    Q
     import React from 'react'
     export default function $1() {
       return <div>$0</div>
     }
♂ rfce
                                                                                    ſĊ
     import React from 'react'
     function $1() {
       return <div>$0</div>
     export default $1
♂ rafcp
                                                                                    Q
     import React from 'react'
     import PropTypes from 'prop-types'
     const $1 = (props) => {
       return <div>$0</div>
     $1.propTypes = {}
     export default $1
```

```
♂ rafc
                                                                                     Q
     import React from 'react'
     export const $1 = () => {
       return <div>$0</div>
     }
♂ rafce
                                                                                     ſŌ
     import React from 'react'
     const $1 = () \Rightarrow {
       return <div>$0</div>
     export default $1
∂ rmc
                                                                                     ſĠ
     import React, { memo } from 'react'
     export default memo(function $1() {
       return <div>$0</div>
     })
♂ rmcp
                                                                                    Q
     import React, { memo } from 'react'
     import PropTypes from 'prop-types'
     const $1 = memo(function $1(props) {
       return <div>$0</div>
     })
     $1.propTypes = {}
     export default $1
♂ rcredux
```

```
ſŪ
     import React, { Component } from 'react'
     import { connect } from 'react-redux'
     export class FileName extends Component {
       render() {
         return <div>$4</div>
       }
     }
     const mapStateToProps = (state) => ({})
     const mapDispatchToProps = {}
     export default connect(mapStateToProps, mapDispatchToProps)(FileName)
♂ rcreduxp
                                                                                     Q
     import React, { Component } from 'react'
     import PropTypes from 'prop-types'
     import { connect } from 'react-redux'
     export class FileName extends Component {
       static propTypes = {
         $2: $3,
       }
       render() {
         return <div>$4</div>
       }
     }
     const mapStateToProps = (state) => ({})
     const mapDispatchToProps = {}
     export default connect(mapStateToProps, mapDispatchToProps)(FileName)
♂ rfcredux
                                                                                     ص
     import React, { Component } from 'react'
     import { connect } from 'react-redux'
     export const FileName = () => {
       return <div>$4</div>
```

```
}
     const mapStateToProps = (state) => ({})
     const mapDispatchToProps = {}
     export default connect(mapStateToProps, mapDispatchToProps)(FileName)
♂ rfreduxp
                                                                                  þ
     import React, { Component } from 'react'
     import PropTypes from 'prop-types'
     import { connect } from 'react-redux'
     export const FileName = () => {
       return <div>$4</div>
     }
     FileName.propTypes = {
       $2: $3,
     }
     const mapStateToProps = (state) => ({})
     const mapDispatchToProps = {}
     export default connect(mapStateToProps, mapDispatchToProps)(FileName)
ſĊ
     const mapStateToProps = (state) => ({})
     const mapDispatchToProps = {}
ℰ React Native Components

∂ rnc

                                                                                  ص
     import React, { Component } from 'react'
     import { Text, View } from 'react-native'
     export default class FileName extends Component {
```

```
render() {
         return (
           <View>
             <Text> $2 </Text>
           </View>
         )
       }
     }
♂ rnf
                                                                                     СŌ
     import React from 'react'
     import { View, Text } from 'react-native'
     export default function $1() {
       return (
         <View>
           <Text> $2 </Text>
         </View>
       )
     }
♂ rnfs
                                                                                     Q
     import React from 'react'
     import { StyleSheet, View, Text } from 'react-native'
     export default function $1() {
       return (
         <View>
           <Text> $2 </Text>
         </View>
       )
     }
     const styles = StyleSheet.create({})
♂ rnfe
                                                                                     ſĠ
     import React from 'react'
     import { View, Text } from 'react-native'
     const $1 = () => {
```

```
return (
         <View>
           <Text> $2 </Text>
         </View>
       )
     }
     export default $1
♂ rnfes
                                                                                      ٦
     import React from 'react'
     import { StyleSheet, View, Text } from 'react-native'
     const $1 = () => {
       return (
         <View>
           <Text> $2 </Text>
         </View>
       )
     }
     export default $1
     const styles = StyleSheet.create({})
  rncs
                                                                                     Q
     import React, { Component } from 'react'
     import { Text, StyleSheet, View } from 'react-native'
     export default class FileName extends Component {
       render() {
         return (
           <View>
             <Text> $2 </Text>
           </View>
         )
       }
     }
     const styles = StyleSheet.create({})
    rnce
```

```
رنی
                                                                               Q
     import React, { Component } from 'react'
     import { Text, View } from 'react-native'
     export class FileName extends Component {
       render() {
        return (
          <View>
            <Text> $2 </Text>
          </View>
        )
      }
     }
     export default $1
⊘ Others
Q
♂ desc
                                                                               Q
     describe('$1', () => {
      $2
     })
♂ test
                                                                               þ
    test('should $1', () => {
      $2
     })
∂ tit
```

```
ſĠ
     it('should $1', () => {
       $2
     })
Q
     import React from 'react'
     import renderer from 'react-test-renderer'
     import { $1 } from '../$1'
     describe('<$1 />', () => {
       const defaultProps = {}
       const wrapper = renderer.create(<$1 {...defaultProps} />)
       test('render', () => {
         expect(wrapper).toMatchSnapshot()
       })
     })
♂ srtest
                                                                                     ſŪ
     import React from 'react'
     import renderer from 'react-test-renderer'
     import { Provider } from 'react-redux'
     import store from 'src/store'
     import { $1 } from '../$1'
     describe('<$1 />', () => {
       const defaultProps = {}
       const wrapper = renderer.create(
         <Provider store={store}>
           <$1 {...defaultProps} />)
         </Provider>,
       )
       test('render', () => {
         expect(wrapper).toMatchSnapshot()
       })
     })
```

Sntest

```
ſŪ
     import 'react-native'
     import React from 'react'
     import renderer from 'react-test-renderer'
     import $1 from '../$1'
     describe('<$1 />', () => {
       const defaultProps = {}
       const wrapper = renderer.create(<$1 {...defaultProps} />)
       test('render', () => {
         expect(wrapper).toMatchSnapshot()
       })
     })
♂ snrtest
                                                                                      ſĊ
     import 'react-native'
     import React from 'react'
     import renderer from 'react-test-renderer'
     import { Provider } from 'react-redux'
     import store from 'src/store/configureStore'
     import $1 from '../$1'
```

ص

∂ hocredux

```
import React from 'react'
import PropTypes from 'prop-types'
import { connect } from 'react-redux'
```

```
export const mapStateToProps = (state) => ({})
     export const mapDispatchToProps = {}
     export const $1 = (WrappedComponent) => {
       const hocComponent = ({ ...props }) => <WrappedComponent {...props} />
       hocComponent.propTypes = {}
       return hocComponent
     }
     export default (WrapperComponent) =>
       connect(mapStateToProps, mapDispatchToProps)($1(WrapperComponent))

∂ hoc

                                                                                     Q
     import React from 'react'
     import PropTypes from 'prop-types'
     export default (WrappedComponent) => {
       const hocComponent = ({ ...props }) => <WrappedComponent {...props} />
       hocComponent.propTypes = {}
       return hocComponent
     }
```