Cheatography

VS Code ES7 React/Redux/React-Native/JS snippets Keyboard Shortc

by Pedrecal via cheatography.com/61949/cs/15957/

Basic Methods

imp→	import moduleName from 'module'
imn→	import 'module'
imd→	import { destructuredModule } from 'module'
ime→	import * as alias from 'module'
ima→	import { originalName as aliasName} from 'module'
exp→	export default moduleName
exd→	export { destructuredModule } from 'module'
exa→	export { originalName as aliasName} from 'module'
enf→	export const functionName = (params) => { }
edf→	export default (params) => { }
met→	methodName = (params) => { }
fre→	arrayName.forEach(element => { }
fof→	for(let itemName of objectName { }
fin→	for(let itemName in objectName { }
anfn→	(params) => { }
nfn→	const functionName = (params) => { }
dob→	const {propName} = objectToDescruct
dar→	const [propName] = arrayToDescruct
**sti→	setInterval(() => { }, intervalTime
sto→	setTimeout(() => { }, delayTime

Basic Methods (cont)

prom→	return new Promise((resolve, reject) => { }
cmmb→	comment block

Console

clg→	console.log(object)
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)
ctr→	console.warn
cin→	console.info

React

imr→	import React from 'react'
imr→	import React, { Component } from 'react'
imr→	import React, { Component } from 'react' & import PropTypes from 'prop-types'
imrpc→	import React, { PureComponent } from 'react'

React (cont)

imrpcp→	import React, { PureComponent } from 'react' & import PropTypes from 'prop-types'
redux→	import { connect } from 'react-redux'
rconst→	constructor(props) with this.state
rconc→	constructor(props, context) with this.state
est→	this.state = { }
cwm→	componentWillMount = () => { } DEPRECATED!!!
cdm→	componentDidMount = () => { }
cwr→	componentWillReceiveProps = (nextProps) => { } DEPRECATED!!!
scu→	shouldComponentUpdate = (nextProps, nextState) => { }
cwup→	componentWillUpdate = (nextProps, nextState) => { } DEPRECATED!!!
cdup→	componentDidUpdate = (prevProps, prevState) => { }
cwun→	componentWillUnmount = () => { }
cwun→	componentWillUnmount = () => { }
gdsfp→	static getDerivedStateFromProps(nextProps, prevState) { }
gsbu→	getSnapshotBeforeUpdate = (prevProps, prevState) => { }
ren→	render() { return( ) }

React (cont)

sst→	this.setState({ })
ssf→	this.setState((state, props) => return { })
props→	propName
state→	this.state.stateName
rcontext→	const ${1:contextName} = React.createContext()
cref→	this.${1:refName}Ref = React.createRef()
fref→	const ref = React.createRef()
bnd→	this.methodName = this.methodName.bind(this)

Redux

rxaction→	redux action template
rxconst→	export const $1 = '$1'
rxreducer→	redux reducer template
rxselect→	redux selector template

React Native

imrn→	import { $1 } from 'react-native'
rnstyle→	const styles = StyleSheet.create({})

GraphQL

grap-hql→	import { compose, graphql } from 'react-apollo'
expgql→	export default compose(graphql($1, { name: $2 }))($3)

PropTypes

pta→	PropTypes.array
ptar→	PropTypes.array.isRequired
ptb→	PropTypes.bool
ptbr→	PropTypes.bool.isRequired
ptf→	PropTypes.func
ptptfr→	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])

PropTypes (cont)

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
ptypes→	static propTypes = {}