VS Code Reactjs snippets

Below is a list of all available snippets and the triggers of each one. The ⇥ means the TAB key.

Snippets

Trigger	Content
rcc⇥	class component skeleton
rrc⇥	class component skeleton with react-redux connect
rrdc⇥	class component skeleton with react-redux connect and dispatch
rccp⇥	class component skeleton with prop types after the class
rcjc⇥	class component skeleton without import and default export lines
rcfc⇥	class component skeleton that contains all the lifecycle methods
rwwd⇥	class component without import statements
rpc⇥	class pure component skeleton with prop types after the class
rsc⇥	stateless component skeleton
rscp⇥	stateless component with prop types skeleton
rscm⇥	memoize stateless component skeleton
rscpm⇥	memoize stateless component with prop types skeleton
rsf⇥	stateless named function skeleton
rsfp⇥	stateless named function with prop types skeleton
rsi⇥	stateless component with prop types and implicit return
fcc⇥	class component with flow types skeleton
fsf⇥	stateless named function skeleton with flow types skeleton
fsc⇥	stateless component with flow types skeleton
rpt⇥	empty propTypes declaration
rdp⇥	empty defaultProps declaration
con⇥	class default constructor with props
conc⇥	class default constructor with props and context

Trigger	Content
est⇥	empty state object
cwm⇥	componentWillMount method
cdm⇥	componentDidMount method
cwr⇥	componentWillReceiveProps method
scu⇥	shouldComponentUpdate method
cwup⇥	componentWillUpdate method
cdup⇥	componentDidUpdate method
cwun⇥	componentWillUnmount method
gsbu⇥	getSnapshotBeforeUpdate method
gdsfp⇥	static getDerivedStateFromProps method
cdc⇥	componentDidCatch method
ren⇥	render method
sst⇥	this.setState with object as parameter
ssf⇥	this.setState with function as parameter
props⇥	this.props
state⇥	this.state
bnd⇥	binds the this of method inside the constructor
disp⇥	MapDispatchToProps redux function

VS Code Reactjs snippets

Below is a list of all available snippets and the triggers of each one. The ⇥ means the TAB key.

The following table lists all the snippets that can be used for prop types. Every snippet regarding prop types begins with `pt` so it's easy to group it all together and explore all the available options. On top of that each prop type snippets has one equivalent when we need to declare that this property is also required.

For example `pta` creates the `PropTypes.array` and `ptar` creates the `PropTypes.array.isRequired`

Trigger	Content
pta⇥	`PropTypes.array,`
ptar⇥	`PropTypes.array.isRequired,`
ptb⇥	`PropTypes.bool,`
ptbr⇥	`PropTypes.bool.isRequired,`
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,`
ptsm⇥	`PropTypes.symbol,`
ptsmr⇥	`PropTypes.symbol.isRequired,`
ptan⇥	`PropTypes.any,`
ptanr⇥	`PropTypes.any.isRequired,`
ptnd⇥	`PropTypes.node,`
ptndr⇥	`PropTypes.node.isRequired,`
ptel⇥	`PropTypes.element,`
ptelr⇥	`PropTypes.element.isRequired,`
pti⇥	`PropTypes.instanceOf(ClassName),`
ptir⇥	`PropTypes.instanceOf(ClassName).isRequired,`
pte⇥	`PropTypes.oneOf(['News', 'Photos']),`
pter⇥	`PropTypes.oneOf(['News', 'Photos']).isRequired,`
ptet⇥	`PropTypes.oneOfType([PropTypes.string, PropTypes.number]),`
ptetr⇥	`PropTypes.oneOfType([PropTypes.string, PropTypes.number]).isRequired,`
ptao⇥	`PropTypes.arrayOf(PropTypes.number),`
ptaor⇥	`PropTypes.arrayOf(PropTypes.number).isRequired,`
ptoo⇥	`PropTypes.objectOf(PropTypes.number),`
ptoor⇥	`PropTypes.objectOf(PropTypes.number).isRequired,`
ptoos⇥	`PropTypes.objectOf(PropTypes.shape()),`
ptoosr⇥	`PropTypes.objectOf(PropTypes.shape()).isRequired,`
ptsh⇥	`PropTypes.shape({color: PropTypes.string, fontSize: PropTypes.number}),`
ptshr⇥	`PropTypes.shape({color: PropTypes.string, fontSize: PropTypes.number}).isRequired,`