

Trigger	Content
<code>rcc</code> →	class component skeleton
<code>rrc</code> →	class component skeleton with react-redux connect
<code>rrdc</code> →	class component skeleton with react-redux connect and dispatch
<code>rccp</code> →	class component skeleton with prop types after the class
<code>rcjc</code> →	class component skeleton without import and default export lines
<code>rcfc</code> →	class component skeleton that contains all the lifecycle methods
<code>rwwd</code> →	class component without import statements
<code>rpc</code> →	class pure component skeleton with prop types after the class
<code>rsc</code> →	stateless component skeleton
<code>rscp</code> →	stateless component with prop types skeleton
<code>rscm</code> →	memoize stateless component skeleton
<code>rscpm</code> →	memoize stateless component with prop types skeleton
<code>rsf</code> →	stateless named function skeleton
<code>rsfp</code> →	stateless named function with prop types skeleton
<code>rsi</code> →	stateless component with prop types and implicit return
<code>fcc</code> →	class component with flow types skeleton
<code>fsf</code> →	stateless named function skeleton with flow types skeleton
<code>fsc</code> →	stateless component with flow types skeleton
<code>rpt</code> →	empty propTypes declaration
<code>rdp</code> →	empty defaultProps declaration
<code>con</code> →	class default constructor with props
<code>conc</code> →	class default constructor with props and context
<code>est</code> →	empty state object
<code>cwm</code> →	<code>componentWillMount</code> method
<code>cdm</code> →	<code>componentDidMount</code> method
<code>cwr</code> →	<code>componentWillReceiveProps</code> method
<code>scu</code> →	<code>shouldComponentUpdate</code> method

<code>cwup →</code>	<code>componentWillUpdate</code> method
<code>cdup →</code>	<code>componentDidUpdate</code> method
<code>cwun →</code>	<code>componentWillUnmount</code> method
<code>gsbu →</code>	<code>getSnapshotBeforeUpdate</code> method
<code>gdsfp →</code>	<code>static getDerivedStateFromProps</code> method
<code>cdc →</code>	<code>componentDidCatch</code> method
<code>ren →</code>	<code>render</code> method
<code>sst →</code>	<code>this.setState</code> with object as parameter
<code>ssf →</code>	<code>this.setState</code> with function as parameter
<code>props →</code>	<code>this.props</code>
<code>state →</code>	<code>this.state</code>
<code>bnd →</code>	binds the <code>this</code> of method inside the constructor
<code>disp →</code>	<code>MapDispatchToProps</code> redux function

Trigger	Content
<code>pta→</code>	<code>PropTypes.array,</code>
<code>ptar→</code>	<code>PropTypes.array.isRequired,</code>
<code>ptb→</code>	<code>PropTypes.bool,</code>
<code>ptbr→</code>	<code>PropTypes.bool.isRequired,</code>
<code>ptf→</code>	<code>PropTypes.func,</code>
<code>ptfr→</code>	<code>PropTypes.func.isRequired,</code>
<code>ptn→</code>	<code>PropTypes.number,</code>
<code>ptnr→</code>	<code>PropTypes.number.isRequired,</code>
<code>pto→</code>	<code>PropTypes.object,</code>
<code>ptor→</code>	<code>PropTypes.object.isRequired,</code>
<code>pts→</code>	<code>PropTypes.string,</code>
<code>ptsr→</code>	<code>PropTypes.string.isRequired,</code>
<code>ptsm→</code>	<code>PropTypes.symbol,</code>
<code>ptsmr→</code>	<code>PropTypes.symbol.isRequired,</code>
<code>ptan→</code>	<code>PropTypes.any,</code>
<code>ptanr→</code>	<code>PropTypes.any.isRequired,</code>
<code>ptnd→</code>	<code>PropTypes.node,</code>
<code>ptndr→</code>	<code>PropTypes.node.isRequired,</code>
<code>ptel→</code>	<code>PropTypes.element,</code>
<code>ptelr→</code>	<code>PropTypes.element.isRequired,</code>
<code>pti→</code>	<code>PropTypes.instanceOf(ClassName),</code>

ptir→	PropTypes.instanceOf(ClassName).isRequired,
pte→	PropTypes.oneOf(['News', 'Photos']),
pter→	PropTypes.oneOf(['News', 'Photos']).isRequired,
ptet→	PropTypes.oneOfType([PropTypes.string, PropTypes.
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, fontSiz PropTypes.number}),
ptshr→	PropTypes.shape({color: PropTypes.string, fontSiz PropTypes.number}).isRequired,