**Consume a React component in a non-React website**

Created a really cool React component, [Chat-Template](https://github.com/sevenleaps/chat-template), a configurable component that simulates instant message conversations.

I wanted to make this component easily consumable for non-react websites as well.

**The approach:**

One thing that was clear from the start is that in order to make the component consumable, I was going to need to package it up [webpack](http://webpack.github.io/" \t "_blank).

Webpack takes modules with dependencies and generates static assets representing those modules

I wrote a node script to wrap the component. It included the component, React and ReactDom and exported a method to display it on a given dom element :

**wrapper.js**

As this code was to be consumed by browsers, I use [babel](https://babeljs.io/) to ensure the best compatibly. Babel is basically a javascript code compiler that converts EC6 javascript code to the much more widely compatible EC5

[.**babelrc**](https://gist.github.com/witnessmenow/4ed36c79db717d48d078f6e19b04d064#file-babelrc)

I now needed to configure webpack. One thing that was needed was to make the method a global variable, so the **showChatTemplate** function was available, this was done using the libraryTarget and library configurations. The webpack config looked like this

[**webpack.dev.config.js**](https://gist.github.com/witnessmenow/46c0d7cacd60704269966741d171b8f1#file-webpack-dev-config-js)

I now had a single javascript file that included everything the consumer needed and could be easily added to a web page.

[**example.html**](https://gist.github.com/witnessmenow/f5829ffff375a982a7d3dc18924676c2#file-example-html)