

## **Look at the file Below...get to know what it is doing**

This is the typical use in React project/ the getWeb3.js file

### **Good overview:**

#### **Principles**

<https://medium.com/@lyricalpolymath/web3-design-principles-f21db2f240c1>

### **Video- old but. Good overview of how it works**

<https://www.youtube.com/watch?v=Jw6VI7NYN3o>

### **State of WEb3- watch this: updated**

<https://www.youtube.com/watch?v=JiDoVOPFx0A>

### **Good Web3:**

<https://www.youtube.com/watch?v=cDmwXtU1fr8>

### **Using Event listener:**

This one gives you an idea of real usage— use the code

<https://www.youtube.com/watch?v=OGeSkM70gwY>

### **Signature storage:**

<https://www.youtube.com/watch?v=w5YqGVfUGcU>

### **Suggestions:**

```
JS truffle.js  JS getWeb3.js x  {} Settings
1  import Web3 from "web3";
2
3  const getWeb3 = () =>
4  new Promise((resolve, reject) => {
5    // Wait for loading completion to avoid race conditions with web3 injection timing.
6    window.addEventListener("load", () => {
7      let web3 = window.web3;
8
9      // Checking if Web3 has been injected by the browser (Mist/MetaMask).
10     const alreadyInjected = typeof web3 !== "undefined";
11
12     if (alreadyInjected) {
13       // Use Mist/MetaMask's provider.
14       web3 = new Web3(web3.currentProvider);
15       console.log("Injected web3 detected.");
16       resolve(web3);
17     } else {
18       // Fallback to localhost if no web3 injection. We've configured this to
19       // use the development console's port by default.
20       const provider = new Web3.providers.HttpProvider(
21         "http://127.0.0.1:7545"
22       );
23       web3 = new Web3(provider);
24       console.log("No web3 instance injected, using Local web3.");
25       resolve(web3);
26     }
27   });
28 });
29
30 export default getWeb3;
```

**<https://truffleframework.com/boxes>**

Unbox everybox and look at the web3 file—check usage

Search “web3” in your ide

-Go to GitHub and look at open source project....dig through the code..

If you get the boxes then it will be easier to hack away

- Also find old react projects and add web3 to them. That will get you going fast..
- Compare the normal react apps vs the Dapp apps with Web3