

Truffle Boxes - Making Life Sweeter

When users begin developing on Ethereum, the first question they ask is, "Where do I start?". In the past, there hasn't been an easy answer: the answer generally consisted of many flavors of "it depends", and usually led to an investigation of what the user was trying to build, a general overview of their development skills, their knowledge of Ethereum, and the libraries and tools they're already most comfortable with. When determining how the Truffle team can help in this process, it quickly became apparent that what our users needed were examples. Many of them. So, being the upstanding community members we are, we set out to create those examples, and today, we're happy to show you what we've accomplished.

Introducing Truffle Boxes

A Truffle box is an example Ethereum application and/or boilerplate that puts complimentary tools and libraries into a single, sweet, easily-downloadable package. Every Truffle box comes with libraries and tools already preinstalled, code that uses those libraries and tools, external scripts (if necessary), as well as helpful README's and documentation. All Truffle boxes are directly integrated into the Truffle command line, and you need only `truffle unbox` to download and prepare your box of choice.

[Click here for a list of all available boxes.](#) Ensure you've installed the latest version of Truffle before opening your first box.

Today's release comes with three officially-supported boxes, as well as one community-supported box. Our officially-supported boxes are:

- [react](#)
 - : A box that pairs Truffle with React in an easy to understand structure, meant for a web-based Ethereum application.
- [react-auth](#)
 - : Ourreact
 - box with extra goodies for authentication powered by smart contracts.
- [react-uport](#)
 - : Likereact-auth
 - , but authentication is powered by Uport.

Perhaps the most exciting part of this release -- our pièce de résistance -- is the [Status Truffle box](#), our first community-supported box. This box was created by the Status team, and instantly links Truffle and Status together with an example application and all the tools and libraries you need to build your own Status dapp.

Get Involved: Make Your Own Box

We've released four boxes out of the gate to get you started, but this isn't just our candy store. We want you to contribute to the boxes on our site. If you have boilerplate code, an example dapp, a helpful integration or an exciting dapp stack (say, an Ethereum-enabled iOS app built with Truffle), send us an email at boxes@trufflesuite.com and we'll look it over. If it's awesome, we'll list it on our site with your name front and center. If it's better than awesome, we'll help contribute and maintain it and make it an officially-supported Truffle box. With Truffle, you can share your code with others and help build the community, one box at a time.

One More thing...

There's always one more thing. Fair warning, this one's techy. Right now, all the boxes you're seeing are static pages showing off what we and the Status team have built. Soon, however, we'll be taking on the ultimate challenge and will be creating a Truffle box registry, built on top of Ethereum, to help market and promote our community-supported boxes. Of course we'll be using Truffle, which adds a whole new layer of sweetness. Stay tuned for more details about the registry and how you can use the Ethereum blockchain to become a Truffle contributor.

As always, we thank everyone for your support, and would love to hear from you. Don't hesitate to reach out on our [community GitHub Discussions channel](#) where hundreds of your fellow Trufflers congregate to answer your questions.

Cheers!

-- Tim & the Truffle Team