

To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy. X

# Developers Guide to the Qtum GitHub



Dev Bharel

[Follow](#)

Jul 25, 2018 · 6 min read

This article will serve as a quick-start guide for developers of all skill levels interested in playing around with Qtum and the projects they offer on their GitHub.

## What is Qtum?

The Qtum developers describe Qtum as “a decentralized blockchain project built on Bitcoin’s UTXO model, with support for Ethereum Virtual Machine based smart contracts, and secured by a proof-of-stake consensus model. It achieves this through the revolutionary Account Abstraction Layer which allows the EVM to communicate with Qtum’s Bitcoin-like UTXO blockchain.” Breaking this down, Qtum provides you with the ability to create and manage DAPP’s through smart contracts, in a similar fashion to Ethereum. Unlike Ethereum however, Qtum has the added benefit of using the Bitcoin UTXO model for added security. For information about the team and the organization head over to Qtum’s home page, once there I would suggest checking out their video explaining the long-term goals of Qtum. If you are looking for more technical and detailed information check out their published white papers. If however, you are looking to find guides on developing with Qtum and how to get involved, read on!!

## Qtum as a user

To get acquainted with Qtum, there are two starter guides which will be very useful to start developing and working on Qtum projects. The user guide serves as a good launch point for those just getting acquainted with Qtum. It provides details on how to host your own Qtum node and how to manage said node. It has ample information about the different types of wallets Qtum offers and how to set them up, along with guides on locking and unlocking your Qtum wallet. It also has a guide on how to stake your qtum.

To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy. X

uses Proof of Authority, should you be interested in learning about that.

Note: there is also a simple guide to ERC20 token creation and deployment, however since this is a development tutorial you will want to look into the second Qtum starter guide to learn about how to get involved in Qtum development.

## Qtum as a developer

The second guide is a developer guide, which should be your go-to reference when beginning to develop in a Qtum environment. It will guide you through several useful things including setting up a local Qtum network to play around with. It shows you how to connect to the testnet for more verbose testing as well as the mainnet for when you are ready to deploy. Using an ERC20 token as an example, it will guide you through the creation of the token as a smart contract with all the important features it may need, and how to deploy it. Perhaps the most useful aspect for developers, however, is the section on how to manipulate said token contract using the Qtum JS library, and even cooler is a guide on how to create a simple DAPP using React. Give it a read through and follow its guides for an excellent introduction to developing in a Qtum environment. Once you start to feel comfortable with these tools, you might be interested in exploring more of what Qtum has to offer developers, to help you with that let's go through some of the key GitHub projects Qtum has to offer.

## Qtum Github Projects for Developers

The projects highlighted below will prove to be extremely useful for the development of any Qtum DAPP or Qtum related project. Make sure you get acquainted with each of them.

### Qtum Boilerplate Project

This is a quick project that the Qtum developers made in order to help you get started on creating DAPP's on Qtum. Follow the instructions on this repo to dip your toes into many aspects of Qtum and what it means to create a Qtum DAPP.

### Qtum Docker

To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy. X

developers can run a local Qtum network on their machine regardless of your OS! The images are available from the docker hub; so, if you already have docker downloaded, you are only a few steps away from getting started. If you need a helping hand to get started with setting up a Qtum regtest environment take a look at this guide, which will explain step by step how to set up the environment. The regtest is easy to use and is a great way to get more acquainted with how Qtum works and will prove to be a pivotal tool for testing your projects whether it be a smart contract or a DAPP.

## Qmix

Developed as the Qtum version of Ethereum's remix, Qmix is an in-browser IDE through which you can write Solidity smart contracts. Not only will you be able to write your contracts and compile them to ensure they are correct, but you can also deploy them and interact with them directly with the Qmix UI. To be able to deploy and interact with the smart contracts you need to connect to a Qtum network, which is why in the help tab of the application you will find instructions on how to connect Qmix to your local regtest network. Once connected, you will be able to test your smart contracts thoroughly before actually deploying them on the mainnet. It is likely that you will become acquainted with this tool if you are planning on developing Qtum DAPPS.

\*Qmix was developed as a contract for Qtum, as such the GitHub repository for this project is not on Qtum's GitHub but rather can be found [here](#).

## Qtum JS

Qtum JS is a JavaScript library built specifically to aid in Qtum smart contract development, it will be extremely beneficial for your DAPP's. Use this tool to be able to build applications and interact with your smart contracts, as well as directly to the Qtum RPC using your favorite frameworks such as React or Angular. Find everything you can do with this library in the documentation.

## Qtum JS Wallet

Similar to Qtum JS, this is a simplified library that will allow you to build lite wallets. It is extracted from another Qtum project (the official web wallet) and uses the Qtum

To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy. X

## Qtum API

The Qtum API is another very useful tool for any application you may be working on. Use the API to get the information from the Qtum network that you may need for your project, on the GitHub page is a full list of all the API calls you can make along with the response that they will return. Qtum explorer, for example, uses this tool and is an example of the kind of project you could build using it.

## Qtum Projects in Development

Everything up to this point has focused on getting to know the Qtum environment and developing your own tools and applications to interact with it. If however, you are interested in being involved in ongoing Qtum projects, take a look at some of the projects that Qtum is still working on. Remember that a good way to get involved on any project is to clone the projects and play around with them, so feel free to dive into any of the Qtum projects old or new and see what you can do. If you find any problem add them to the appropriate GitHub issues or if you find a solution make a pull request, which is in itself another great way to get involved and gain experience in development. To stay involved and informed also consider heading over to the Qtum forum.

Additional Qtum Projects worth looking into:

**Qtum Electrum** a lightweight Qtum wallet.

**Qtum Enterprise** currently in development, and is Qtum's enterprise environment so that companies can run their own closed network for whatever solution they may be looking for.

**Qtum Explorer** this is the GitHub repo of the explorer discussed previously in the article.

**Qtum Solar** is a prototype project that Qtum is working on for the deployment of smart contracts.

To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy. X

contracts on popular programming languages rather than being bound to solidity.

**Qtum Lightning** is the Qtum implementation of Bitcoin's lightning network. It is still in a development stage.

**Qtum Portal** is a web server through which to run third-party DAPPS.

**Qtum IOS wallet** is the repo for the wallet found in the apple store.

**Qtum android wallet** is the repo for the wallet found on the google play store.

Ethereum    Qtum    Blockchain    Smart Contracts    Qtum 101

About    Help    Legal

Get the Medium app

