---

title: Ethereum for Rust developers description: Learn how to develop for Ethereum using rust-based projects and tooling lang: en incomplete: true

---

Learn how to develop for Ethereum using Rust-based projects and tooling

Use Ethereum to create decentralized applications (or "dapps") that utilize the benefits of cryptocurrency and blockchain technology. These dapps can be trustworthy, meaning that once they are deployed to Ethereum, they will always run as programmed. They can control digital assets in order to create new kinds of financial applications. They can be decentralized, meaning that no single entity or person controls them and are nearly impossible to censor.

Getting started with smart contracts and the Solidity language {#getting-started-with-smart-contracts-and-solidity}

Take your first steps to integrating Rust with Ethereum

Need a more basic primer first? Check outethereum.org/learn or ethereum.org/developers.

- Blockchain Explained
- Understanding Smart Contracts
- Write your First Smart Contract
- Learn How to Compile and Deploy Solidity

Beginner articles {#beginner-articles}

- Choosing an Ethereum Client
- The Rust Ethereum Client * **Note that OpenEthereum has been deprecated and is no longer being maintained.** Use it with caution and preferably switch to another client implementation.
- Sending Transaction to Ethereum Using Rust
- A step-by-step tutorial on how to write contracts in rust Wasm for Kovan

Intermediate articles {#intermediate-articles}

Advanced use patterns {#advanced-use-patterns}

- pwasm_ethereum externs library to interact with Ethereum-like network
- Build A Decentralized Chat Using JavaScript and Rust
- Build a Decentralized Todo App Using Vue.js & Rust
- Build a blockchain in Rust

Rust projects and tools {#rust-projects-and-tools}

- pwasm-ethereum - *Collection of externs to interact with ethereum-like network*
- Lighthouse - *Fast Ethereum consensus layer client*
- Ethereum WebAssembly - *Proposed redesign of the Ethereum smart contract execution layer using a deterministic subset of WebAssembly*
- oasis_std - *OASIS API reference*
- Solaris - *Solidity Smart Contracts unit test harness using the native Parity Client EVM.*
- SputnikVM - *Rust Ethereum Virtual Machine Implementation*
- Wavelet - *Wavelet smart contract in Rust*
- Foundry- *Toolkit for Ethereum application development*
- Ethers_rs- *Ethereum library and wallet implementation*
- SewUp - *A library to help you build your Ethereum webassembly contract with Rust and just like develop in a common*

backend

- Substreams - *Parallelized blockchain data indexing technology*
- Reth Reth (short for Rust Ethereum, pronunciation) is a new Ethereum full-node implementation

Looking for more resources? Check outethereum.org/developers.

Rust community contributors {#rust-community-contributors}

- Ethereum WebAssembly
- Oasis Gitter
- Parity Gitter
- Enigma