

## Tessera

- How to
  - Get started with Tessera
    - Dependencies
    - Install distribution jar
    - Start Tessera
  - Generate keys
  - Configure
  - Use
  - Migrate from Constellation
- Tutorials
- Concepts
- Reference

## Getting started

## Prerequisites

1. Install necessary dependencies
2. Install Tessera

## Starting Tessera and sending a payload

## 1. Generate keys

Generate a key pair in files called `myKey.pub` and `myKey.key`.

```
Bash
tessera -keygen -filename mykey
```

When prompted to enter a password, press enter to generate an unlocked key.

**Caution**

We are using unlocked keys for educational purposes only. Ensure private keys are secured appropriately in production environments.

## 2. Create a configuration file

Create a file called `tessera.conf` and add the following properties.

```
Tessera configuration file
{
  "useWhiteList": false,
  "jdbc": {
    "username": "sa",
    "password": "",
    "url": "jdbc:h2://target/h2/tessera1",
    "autoCreateTables": true
  },
  "serverConfigs": [
    {
      "app": "ThirdParty",
      "enabled": true,
      "serverAddress": "http://localhost:9081",
      "communicationType": "REST"
    },
    {
      "app": "QZT",
      "enabled": true,
      "serverAddress": "unix:/tmp/test.ipc",
      "communicationType": "REST"
    },
    {
      "app": "PPP",
      "enabled": true,
      "serverAddress": "http://localhost:9081",
      "sslConfig": {
        "tls": "QZT"
      },
      "communicationType": "REST"
    }
  ],
  "peer": [
    {
      "url": "http://localhost:9081"
    },
    {
      "url": "http://localhost:9083"
    }
  ],
  "keys": {
    "passwords": [],
    "keyData": [
      {
        "privateKeyPath": "myKey.key",
        "publicKeyPath": "myKey.pub"
      }
    ]
  },
  "alwaysSendTo": []
}
```

## 3. Start Tessera node

```
tessera -configfile tessera.conf
```

**Tip**

If the command line help is displayed, enable debugging and try again.

```
tessera --debug -configfile tessera.conf
```

## 4. Upcheck

Use the `upcheck` method to confirm Tessera is up and running.

```
Request      Result
curl http://localhost:9081/upcheck
```

## Prerequisites

## Starting Tessera and sending a payload

1. Generate keys
2. Create a configuration file
3. Start Tessera node
4. Upcheck



# Quorum is extensible. Develop with Ethereum's familiar tools tailored for enterprise.

GET STARTED

## Get Started with Quorum

BUILD WITH

### Hyperledger Besu

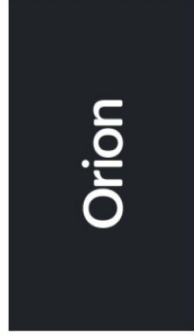


#### Hyperledger Besu

Hyperledger Besu is an open source Ethereum client maintained by the Hyperledger community, including ConsensusSys. Besu is Mainnet compatible, Java-based, Apache 2.0 licensed, and extensible through its Java plugin framework.

[HYPERLEDGER BESU DOCS →](#)

ConsensusSys Quorum is an open-source protocol layer that provides developers with the flexibility and reliability needed to make their blockchain applications successful. ConsensusSys Quorum comprises a suite of configurable components and APIs, enabling you to customize your use case and production environment.



#### Orion

Orion is a private transaction manager for the Hyperledger Besu client, developed under the Apache 2.0 license and written in Java.

[ORION DOCS →](#)

BUILD WITH  
**GoQuorum**

## EthSigner

### EthSigner

EthSigner is a client-agnostic Ethereum transaction signer written in Java and Apache 2.0 licensed. EthSigner offers private key storage and management by separating private key management from transaction validation.

[ETHSIGNER DOCS](#) →

## Besu Bundle

### Hyperledger Besu Bundle

The Besu bundle contains the latest production-ready versions of Hyperledger Besu, Orion, and EthSigner. It is recommended for users who have completed the quickstart for these products and want to set up their own networks.

[DOWNLOAD LATEST VERSION](#) →  
[ALTERNATE INSTALLATION OPTIONS](#) →

## GoQuorum

### GoQuorum

GoQuorum is an open-source Ethereum client maintained by ConsenSys. GoQuorum is Go-based and GPL licensed.

[GOQUORUM DOCS](#) →

## Tessera

### Tessera

Tessera is a private transaction manager for the GoQuorum client, developed under the Apache 2.0 license and written in Java. You can use Tessera to send private transactions, including credit default swap agreements and other financial contracts

[TESSERA DOCS](#) →

## EthSigner

### EthSigner

EthSigner is a client-agnostic Ethereum transaction signer written in Java and Apache 2.0 licensed. EthSigner offers private key storage and management by separating private key management from transaction validation.

[ETHSIGNER DOCS](#) →

## GoQuorum Bundle

### GoQuorum Bundle

The GoQuorum bundle contains the latest production-ready versions of GoQuorum, Tessera, and EthSigner. It is recommended for users who have completed the quickstart for these products and want to set up their own networks.

[DOWNLOAD LATEST VERSION](#) →  
[ALTERNATE INSTALLATION OPTIONS](#) →

# We've Built Quorum with Everything You Need



## Permissioning and Privacy

The Quorum open source protocol layer enables businesses to build on public or private Ethereum networks, ensuring our solution fits any potential regulatory and security requirements.



## Consensus Mechanisms

Both the Hyperledger Besu and GoQuorum variants support Proof of Authority (IBFT and Clique) consensus mechanisms, while Besu also supports Proof of Work (EthHash) for the Ethereum Mainnet, and GoQuorum also supports RAFT ordering.



## Production-Ready

Quorum provides familiar connectivity to monitor the performance and uptime of enterprise solutions. Monitoring, logging and event streaming for High Availability, and database backups for Disaster Recovery are all supported by Quorum projects.



## Ecosystem Compatibility

We work with the existing tools you know: Truffle, MetaMask, Remix, OpenZeppelin, and more. Quorum now has compatible options for the Ethereum Mainnet.



## Quickstarts and Support

Robust documentation is available for the entire Quorum open source layer. Documentation is maintained by the ConsenSys team and support is always available.



## Enterprise Add-ons

Building with Quorum means additional access to a number of market-ready enterprise product modules ready for Mainnet and private network deployment.

## Receive support for your Quorum implementation

### ConsenSys Quorum Support

ConsenSys offers a range of support services, from developer support to 24x7 production support. Contact us to discuss our support for our Quorum open-source layer.

[CONTACT US](#)

### ConsenSys Quorum Helpdesk

Submit a ticket for help with specific ConsenSys Quorum issues and our Customer Service team will reach out.

[SUBMIT A TICKET](#)

Upgrade Your Business  
Blockchain

Ethereum is equipped to optimize your business. On top of our Quorum open-source protocol layer, ConsenSys offers product modules that can be deployed across industries and use cases.

[EXPLORE OUR STACK](#)

## Join the ConsenSys Quorum Community



- PRODUCTS
- DEVELOPERS
- ENTERPRISE
- BLOG
- SUPPORT
- FAQ
- LOGIN



### Subscribe to Quorum

Sign up to get our latest product updates, resources, and more straight to your inbox.

e-mail address

