

DIVA.EXCHANGE

Javascript for Privacy-Protecting Peer-to-Peer Applications



Usage of the I2P-SAM Javascript Library: Anonymized and End-to-End Encrypted Communication

Author: Konrad Bächler

Twitter: @DigitalValueX, Web: <https://diva.exchange>



DIVA.EXCHANGE

Git repos to fork...



About diva.exchange



- Non profit association, open to everyone
- A loose bunch of Devs & Researchers - spread all over the world
- «DIVA - Free Banking Technology for Everyone» means:
handle all kind of Digital Values under your own control and responsibility and apply your very own philosophy of privacy without being nudged by others
- No centralized business model (pointless); no token/coin.



Agenda

- Motivation: why I2P-SAM got developed
- Setting up the distributed overlay I2P network for testing
- Creating applications using the I2P-SAM library
- Use cases: wide range of peer-to-peer applications
- Discussion



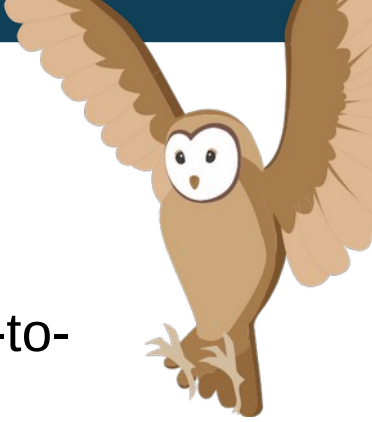
Hello I2P Network



- A few basic facts (some are simplified - educational reasons):
 - I2P is an overlay network (misleading name «darknet» is just used by dubious media desperately in need for clicks)
 - It's a peer-to-peer network where every node in the network acts as a router
 - I2P itself has no storage capabilities – it is a transport layer
 - Messages travelling through the network are multiple times encrypted (like a garlic: it has multiple layers) – call it «Confidentiality feature»
 - Messages hop over several routers within the network to their final destination (using «tunnels») – call it «Anonymity feature»
- **In a nutshell: I2P = confidential & anonymous message transport**



Motivation: Why I2P-SAM?



- I2P is one of the most reliable, fully private, well-researched peer-to-peer networks
- There was no maintained and modern JavaScript/TypeScript library available for I2P
- Divachain (blockchain = fully and truly distributed storage layer designed for I2P) needed a modern library to handle the networking
- SAM itself is a great interface to communicate with the local I2P router, to do things like:
 - Create new local destinations
 - Send messages over the I2P network



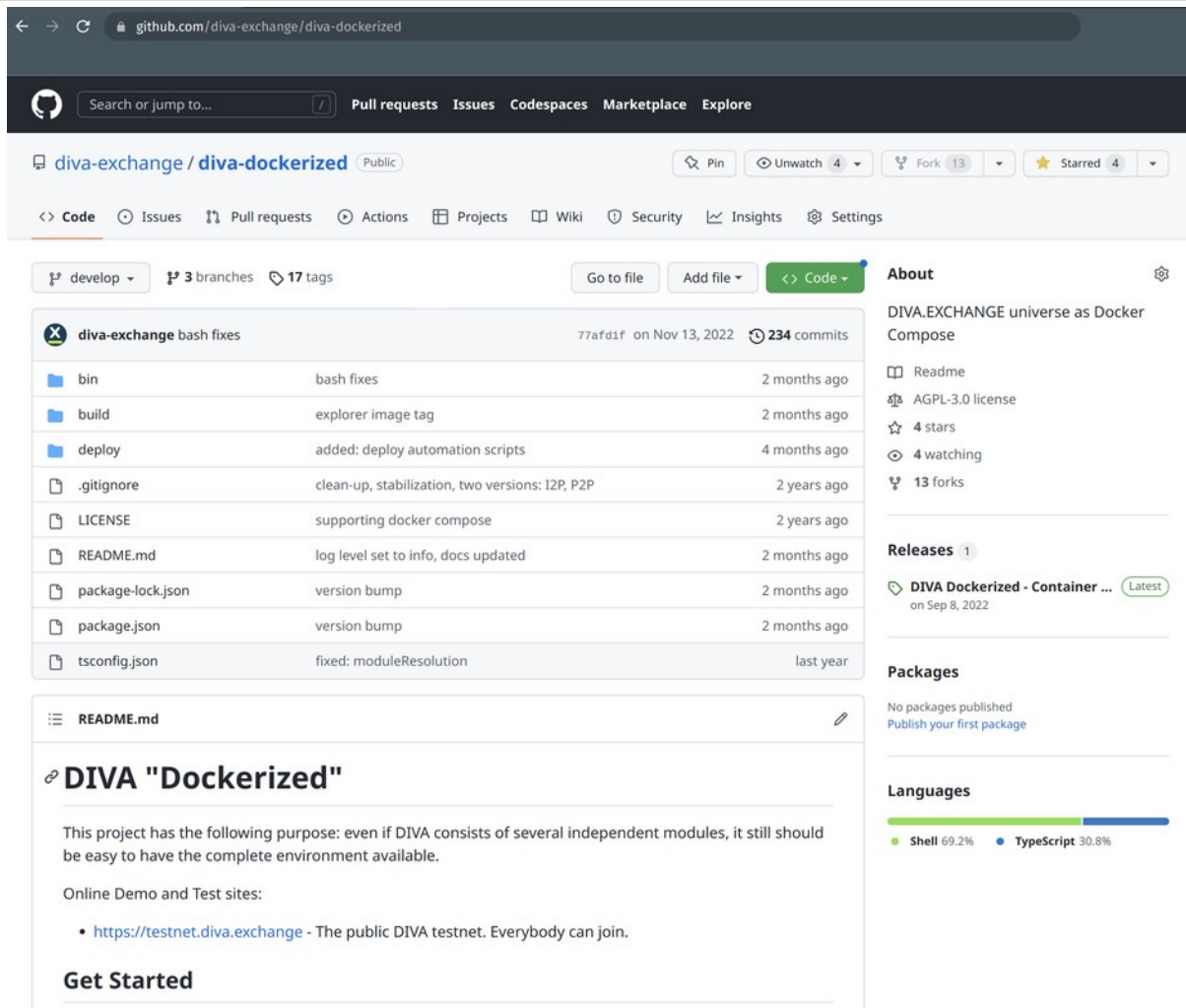
How to Setup a Test Network



- Build Peer-to-Peer applications in a fully distributed system: use I2P-SAM
- In general, it is difficult to develop against to test fully distributed systems
- **Here is the solution**: instantly create a local developer and test network with storage capabilities by using docker and <https://github.com/diva-exchange/diva-dockerized>



Setup



github.com/diva-exchange/diva-dockerized

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

diva-exchange / diva-dockerized (Public)

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

develop 3 branches 17 tags

Go to file Add file <> Code

diva-exchange bash fixes 77afd1f on Nov 13, 2022 234 commits

bin	bash fixes	2 months ago
build	explorer image tag	2 months ago
deploy	added: deploy automation scripts	4 months ago
.gitignore	clean-up, stabilization, two versions: I2P, P2P	2 years ago
LICENSE	supporting docker compose	2 years ago
README.md	log level set to info, docs updated	2 months ago
package-lock.json	version bump	2 months ago
package.json	version bump	2 months ago
tsconfig.json	fixed: moduleResolution	last year

README.md

DIVA "Dockerized"

This project has the following purpose: even if DIVA consists of several independent modules, it still should be easy to have the complete environment available.

Online Demo and Test sites:

- <https://testnet.diva.exchange> - The public DIVA testnet. Everybody can join.

Get Started

About

DIVA.EXCHANGE universe as Docker Compose

Readme

AGPL-3.0 license

4 stars

4 watching

13 forks

Releases 1

DIVA Dockerized - Container ... Latest on Sep 8, 2022

Packages

No packages published
Publish your first package

Languages

Shell 69.2% TypeScript 30.8%



DIVA.EXCHANGE

Page: 8
Author: Konrad Bächler, Twitter: @DigitalValueX

Git repos to fork...




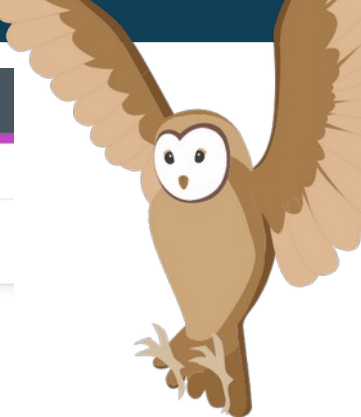
How to Setup a Test Network



- There exists a also a **simplified** option to have access to the I2P network for local development:
 - Use the package (`npm i @diva.exchange/i2p-sam`)
 - Use docker compose to start and stop/purge `test/sam.diva.i2p.yml`




NPM



npmjs.com/package/@diva.exchange/i2p-sam?activeTab=readme

Never Panic Much Pro Teams Pricing Documentation

npm Search packages Search Sign Up Sign In

@diva.exchange/i2p-sam 

4.1.6 • Public • Published a month ago

Readme Code Beta 2 Dependencies 4 Dependents 31 Versions

I2P SAM

An I2P SAM library: enabling applications to communicate through the I2P network. I2P is a fully distributed, "privacy-by-design" network.

To get I2P up and running, take a look at the project: <https://github.com/diva-exchange/i2p>

Use Cases

I2P is an instantly available peer-to-peer network which can be used for things like:

- chat, social media and alike - all private and secure
- distributed databases, aka blockchains (see <https://testnet.diva.exchange> as an example)
- gaming, file sharing and ... whatever else you come up with

I2P is fully distributed, well researched and gets further developed by a competent community.

This I2P SAM library helps developers to create an I2P application quickly and hassle-free.

Get Started


```
npm i @diva.exchange/i2p-sam
```

or, lighter, without developer dependencies:


Install

```
> npm i @diva.exchange/i2p-sam
```

Repository

 github.com/diva-exchange/i2p-sam

Homepage

 [github.com/diva-exchange/i2p-sam#rea...](https://github.com/diva-exchange/i2p-sam#readme)



Weekly Downloads

4,932

Version	License
4.1.6	Apache-2.0

Unpacked Size	Total Files
90.2 kB	34

Issues	Pull Requests
0	0



Docker



```
#
# Apache License - Copyright (c) 2021-2022 diva.exchange
#
# Author/Maintainer: DIVA.EXCHANGE Association <contact@diva.exchange>
#

version: "3.7"

services:
  sam.diva.i2p:
    container_name: sam.diva.i2p
    image: divax/i2p:current
    restart: unless-stopped
    environment:
      ENABLE_SAM: 1
      BANDWIDTH: P
    volumes:
      - sam.diva.i2p:/home/i2pd/
    networks:
      network.sam.diva.i2p:
        ipv4_address: 172.19.74.11

networks:
  network.sam.diva.i2p:
    name: network.sam.diva.i2p
    ipam:
      driver: default
      config:
        - subnet: 172.19.74.0/24

volumes:
  sam.diva.i2p:
    name: sam.diva.i2p
```



Create an Application (1)



- Assumption: create a simple **I2P browser** (=reading data from the network)
- Step one: start a local I2P node to be part of the I2P network

```
docker compose -f test/sam.diva.i2p.yml up -d
```

- Step two: use the first quick start example



Quick Start - Examples

How to Use Streams

Send an HTTP GET request to diva.i2p and output the response:

```
import { createStream } from '@diva.exchange/i2p-sam';

createStream({
  stream: {
    destination: 'diva.i2p'
  },
  sam: {
    host: '127.0.0.1',      # your local I2P SAM host
    portTCP: 7656          # your local I2P SAM port
  },
}).then((sam) => {
  sam.on('data', (data: Buffer) => {
    console.log('Incoming Data: ' + data.toString());
  });
  sam.send(Buffer.from('GET / HTTP/1.1\r\nHost: diva.i2p\r\n\r\n'));
});
```

172.19.74.11

Why?

See docker compose .yaml file



Create an Application (2)



- Assumption: create a simple **I2P service** (=serving data to others, like a website)
- Step one: start a local I2P node to be part of the I2P network

```
docker compose -f test/sam.diva.i2p.yml up -d
```
- Step two: use the second quick start example



Forward incoming streaming data to a local socket server:

```
import { createStream, createForward, I2pSamStream } from '@diva.exchange/i2p-sam';
import net from 'net';
```

```
(async () => {
  const serverForward = net.createServer((c) => {
    console.debug('client connected');
    c.on('end', () => {
      console.debug('client disconnected');
    });
    c.on('data', (data: Buffer) => {
      console.debug(data.toString());
      c.write(`Hello Client!\n`);
    });
  });
  serverForward.listen(20222, '127.0.0.1');

  const samForward: I2pSamStream = await createForward({
    sam: {
      host: '127.0.0.1',      # your local I2P SAM host
      portTCP: 7656          # your local I2P SAM port
    },
    forward: {
      host: '127.0.0.1',      # your local listener, see above
      port: 20222,           # your local listener, see above
    },
  });
});
```

172.19.74.11

```
const samClient: I2pSamStream = await createStream({
  sam: {
    host: '127.0.0.1',      # your local I2P SAM host
    portTCP: 7656          # your local I2P SAM port
  },
  stream: {
    destination: samForward.getPublicKey()
  },
});
// event handler
samClient.on('data', (data: Buffer) => {
  console.debug(data.toString());
});
// send some data to destination
samClient.stream(Buffer.from(`Hi Server!\n`));
})();
```



Examples of Use Cases



- DIVA is working on «Free Banking For Everyone» - join in :)
- Chat – I2P could really use a chat with a great user interface
- Games – What about implementing one using some fully distributed source of randomness?



DIVA.EXCHANGE

Discussion / Links



Package: <https://www.npmjs.com/package/@diva.exchange/i2p-sam>

Web: <https://diva.exchange/>

Twitter: [@DigitalValueX](https://twitter.com/DigitalValueX)

Mastodon: [@social@social.diva.exchange](https://social.diva.exchange/@social)

Telegram Group: https://t.me/diva_exchange_chat_de

Source Code (AGPL3 or better; Apache 2.0) & Research/Academia:
<https://github.com/diva-exchange>

I2P & Docs: <https://geti2p.net>

