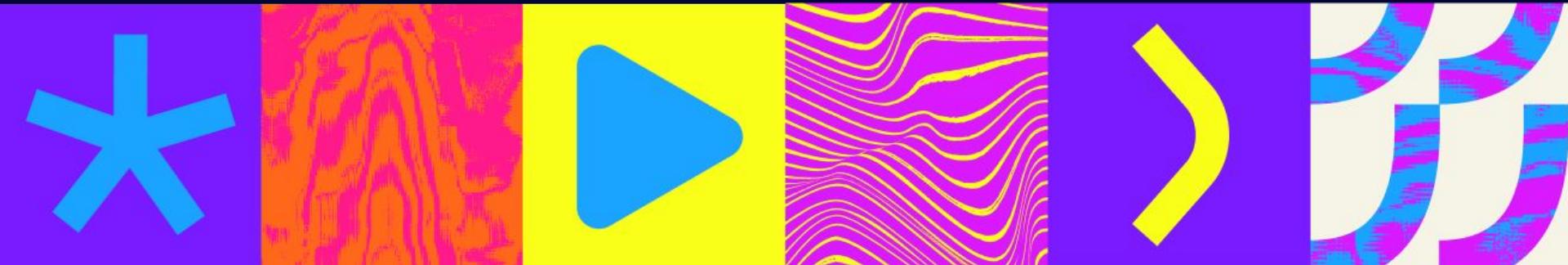




Workshop Introduction to the XRP Ledger





Workshop I

Introduction to the XRP Ledger

Hack4Good Hackathon 2025

November 29th-30th 2025
XRPL Commons & ECE Paris



Commons



ECE





Agenda

- 1 Introduction to the XRP Ledger**
- 2 Interaction with XRPL**
- 3 Introduction to Scaffold XRP**
- 4 More documentation**

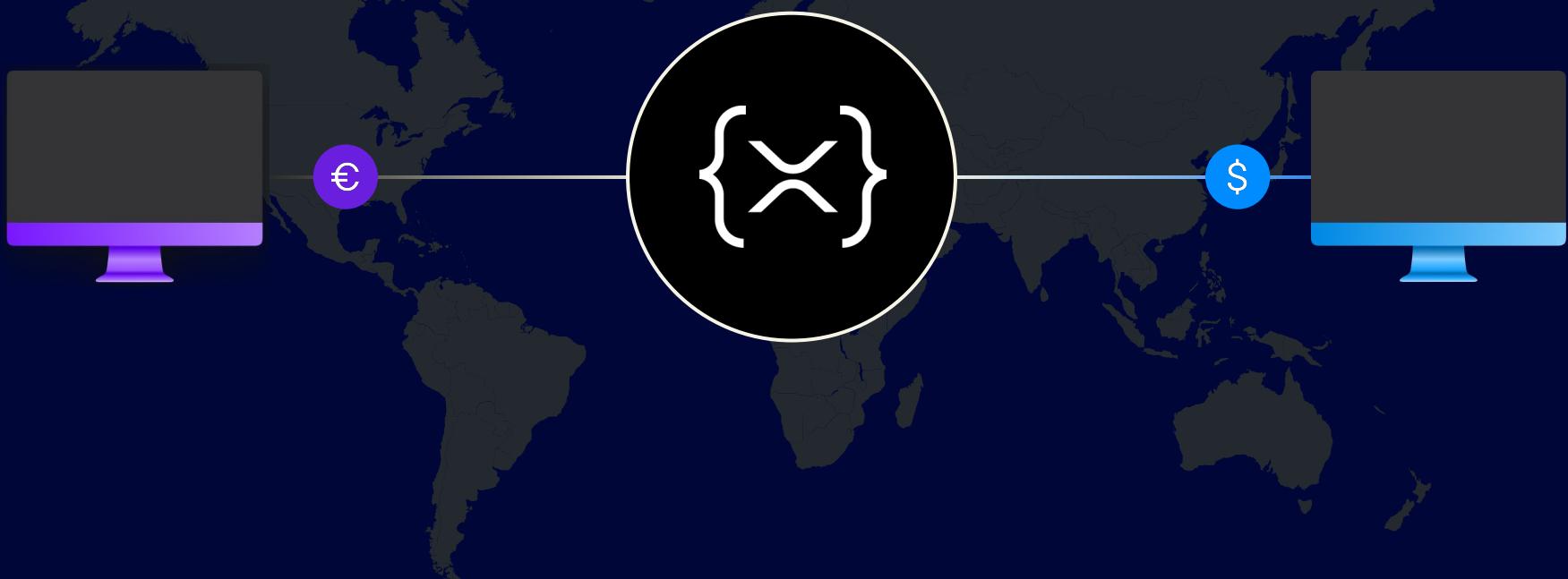
1

Introduction to the XRP Ledger

1. Introduction to the XRP Ledger



XRP Ledger (XRPL) launched in 2012 to address limitations of crypto and fiat currencies for financial use cases, specifically payments



The XRP Ledger



Open Source



Decentralized



**Many
libraries for
coding**



Safe (12y w/o interruption)



**Low carbon
footprint**

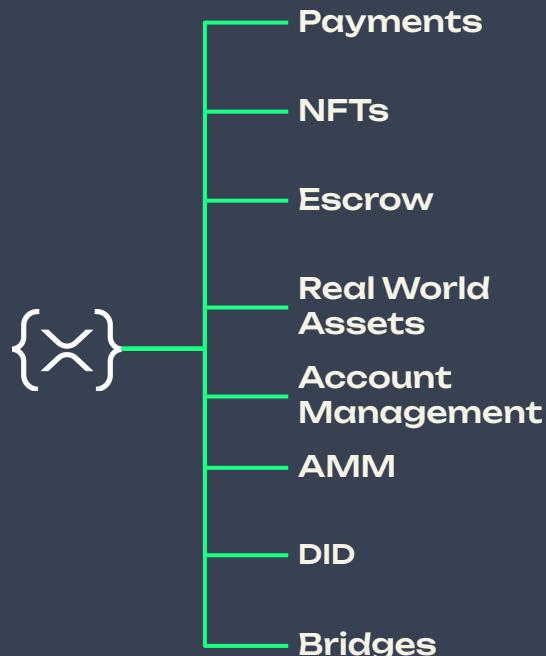


Fast (4s finality)

Specificities of the XRPL



One Endpoint to do all the things



Single API

No need to stitch together disparate systems or spend months integrating complex technology - simply connect into XRPL through a single API

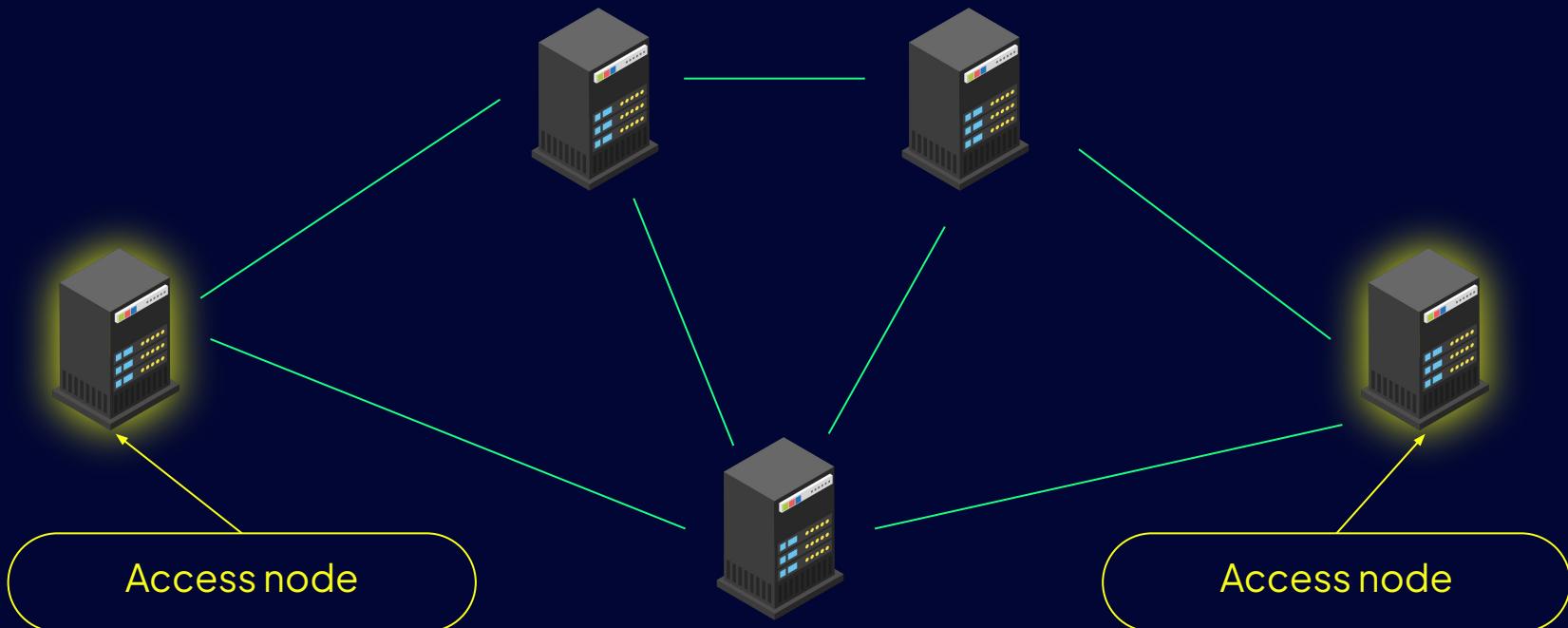
Minimal Code Required

Astonishingly simple, you can get up and running on the XRP Ledger in as little as few lines of code using familiar programming languages (JS, Python, Java, and many more)

1. Introduction to the XRP Ledger



XRPL Network



2

Interaction with XRPL

Languages with an XRPL SDK



Java



TypeScript
Javascript



Python



Rust



Go

Recommended: JavaScript/TypeScript or Python (most up to date)

2. Interaction with XRPL



Connect to a Node

```
import { Client } from "xrpl";

const client = new Client("wss://s.altnet.rippletest.net:51233");

async function main() {
    await client.connect();
    await client.disconnect();
}
```

2. Interaction with XRPL

Create a Wallet



```
// Generate key pair and call the faucet
async function createWallet(client: xrpl.Client) {
  const { wallet, balance } = await client.fundWallet();
}
```

Prepare & Send the Transaction

```
async function sendPaymentTx(  
    client: xrpl.Client,  
    wallet: xrpl.Wallet,  
    address: string,  
    amount: number,  
) {  
    const tx: xrpl.Payment = {  
        TransactionType: "Payment",  
        Account: wallet.classicAddress,  
        Destination: address,  
        Amount: xrpl.xrpToDrops(amount),  
    };  
  
    return await client.submitAndWait(tx, {  
        autofill: true,  
        wallet,  
    });  
}
```

2. Interaction with XRPL

Get the Result

```
{  
    api_version: 2,  
    id: 22,  
    result: {  
        close_time_iso: "2024-12-06T13:39:10Z",  
        ctid: "C02C0E9E00010001",  
        hash: "A2D8910A45D19A91755F3BBC1E29F5FC97C438B0E117E3FBD042DAD1C9A94B98",  
        ledger_hash: "8DAABD0B422F691F3E806EF0E78B8D3F5D7F7D831E97661754A6E31B5E7CEA7C",  
        ledger_index: 2887326,  
        meta: {  
            AffectedNodes: [ { ModifiedNode: [Object] }, { CreatedNode: [Object] } ],  
            TransactionIndex: 1,  
            TransactionResult: "tesSUCCESS",  
            delivered_amount: "10000000"  
        },  
        tx_json: {  
            Account: "rELCjuVzgjBrBUexXoytR59gzEMTU3BwBc",  
            DeliverMax: "10000000",  
            Destination: "rf26gfMAfxaSK8cRJ8b3HpSn11N4v5xD9h",  
            Fee: "12",  
            Flags: 0,  
            LastLedgerSequence: 2887344,  
            Sequence: 2887324,  
            SigningPubKey: "ED598042DFC6F9C65B16002F042B57AA4372EC6B12B9F3862F772A4FCF7B331BBC",  
            TransactionType: "Payment",  
            TxnSignature: "4ABCFD76572234A1B51BE6509C6364835530EE44C2B59005D4A5ACE4C84B13F9EF0025CF84A5C1F922F500877E2BCFA  
0C6696A434024F02E3A776E0F7B485C07",  
            date: 786807550,  
            ledger_index: 2887326  
        },  
        validated: true  
    },  
    type: "response"  
}
```

2. Interaction with XRPL

The Entire Code

```
import xrpl from "xrpl";

const client = new xrpl.Client("wss://s.altnet.rippletest.net:51233");

async function main() {
    // 1. we connect to a node
    await client.connect();

    // 2. we create a wallet
    const wallet = await createWallet(client);

    // 3. we prepare and send the tx
    // Here we want to send 10 XRP to rf26gfMAfxaSK8cRJ8b3HpSn11N4v5xD9h
    const amount = 10;
    const tx = await sendPaymentTx(
        client,
        wallet,
        "rf26gfMAfxaSK8cRJ8b3HpSn11N4v5xD9h",
        amount,
    );

    // 4. we get the result
    console.log(tx);
    await client.disconnect();
}
```

3

Introduction to Scaffold XRP

3.

Introduction to Scaffold XRP



Scaffold XRP



🔗 <https://xrpl.at/ece2025/scaffold>

The screenshot shows the Scaffold-XRP dApp interface. At the top, there's a header with a blue 'X' icon and the text 'Scaffold-XRP'. To the right, a dark button displays a truncated address: 'rfsK...fy5w'. Below the header, the title 'Scaffold-XRP' is centered, followed by the subtitle 'A starter kit for building dApps on XRPL with smart contracts'. A large section titled 'Account Info' contains fields for 'Address' (showing 'rfsK...fy5w'), 'Network' (set to 'Testnet (testnet)'), and 'Wallet' (set to 'GemWallet'). A note below the wallet field says 'Click the button showing your address to disconnect'. Another large section titled 'Send Transaction' includes a 'Destination Address' input field containing 'rN7n7otQDd6Fc...'. An 'Amount (drops)' input field is partially visible below it.

Live Demo!



4

More documentation

4. More documentation



Ecosystem Map



<https://map.xrpl-commons.org>

XRPL Core Protocol Tools ↗

Filters ↴
subcategory ↴ status ↴ tags ↴


XRP Ledger Foundation ↗

The XRP Ledger Foundation is an independent non-profit entity which works to support development and usage of the XRP Ledger.


XRP Services ↗

Execute different XRP Ledger transactions with XUMM


XRP BALANCE . INFO

XRP Balance ↗

Track your XRP balance


XRPL LABS

XRPL Labs ↗

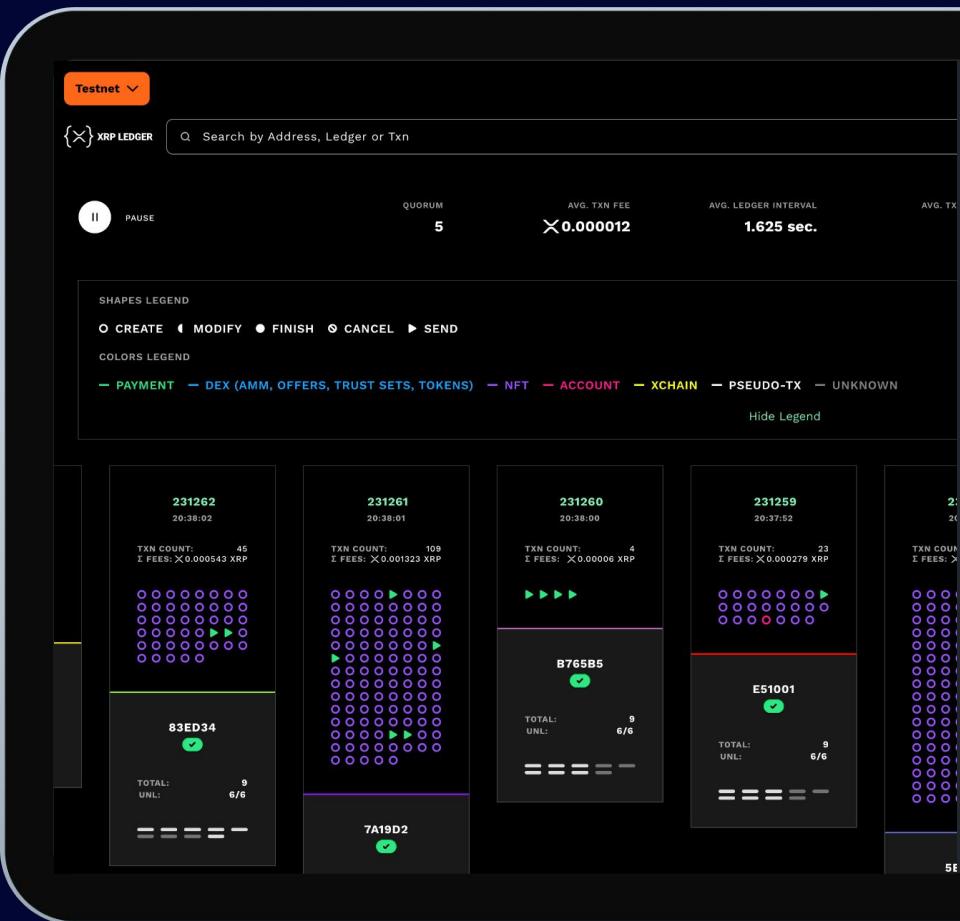
The XRPL Labs flagship product is XUMM, a non custodial XRP Ledger client. The XUMM app relies on the XRP Ledger and the XUMM platform...

4. More documentation

Block Explorer



🔗 <https://testnet.xrpl.org>



4. More documentation



Faucets



<https://xrpl.org/resources/dev-tools/xrp-faucets/>

The screenshot shows the XRP Faucets documentation page. At the top, there's a banner for "XRP Ledger Apex is back in Amsterdam" with a "Register Now" button. Below the banner, there are navigation links for "Docs", "Resources", and "Community", along with a search bar.

XRP Faucets

These parallel XRP Ledger test networks provide platforms for testing changes to the XRP Ledger and software built on it, without using real funds.

These funds are intended for **testing** only. Test networks' ledger history and balances are reset as necessary. Devnets may be reset without warning.

All balances and XRP on these networks are separate from Mainnet. As a precaution, do not use the Testnet or Devnet credentials on the Mainnet.

Choose Network:

- Testnet**: Mainnet-like network for testing applications.
- Devnet**: Preview of upcoming amendments.
- Xahau-Testnet**: Hooks (L1 smart contracts) enabled Xahau testnet.

[Generate Testnet credentials](#)

Testnet Servers

```
// WebSocket  
wss://s.altnet.ripplestest.net:51233
```

```
// JSON-RPC  
https://s.altnet.ripplestest.net:51234/
```

Devnet Servers

```
// WebSocket  
wss://s.devnet.ripplestest.net:51233
```

```
// JSON-RPC  
https://s.devnet.ripplestest.net:51234/
```

Xahau-Testnet Servers

```
// WebSocket  
wss://xahau-test.net/
```

```
// JSON-RPC  
https://xahau-test.net/
```

4. More documentation

Hackathon GitHub



🔗 <http://xrpl.at/ece2025>

HAKS Hackathon 2025



Welcome to the Hack4Good 2025 with ECE Paris !

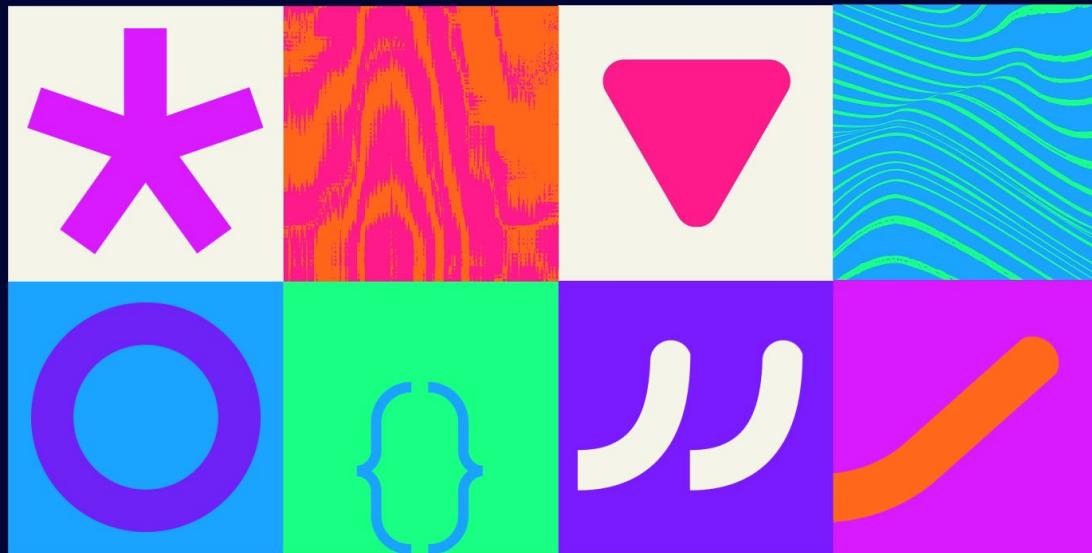
As part of this hackathon, the theme is Crypto for Good. You are free to develop a project around this topic and to unleash your creativity by approaching it in an original and meaningful way.

Crypto for Good

This track encourages creating blockchain solutions for the greater good, financial inclusion, humanitarian aid, climate resilience, and supply chain transparency. Here are some project examples :

- Financial Inclusion: payments, savings, and transfers accessible to the unbanked, even through SMS or WhatsApp, without the need for a smartphone or bank account.

Ask me anything!



Thank you! *o

