Web3 Enabler:

XRP for Salesforce

User Guide

V1.0

Overview

Web3 Enabler: XRP for Salesforce enables users to accept XRP on the XRP Ledger in Salesforce.

XRP for Salesforce is safe and secure by design. Neither your Salesforce Org nor Web3 Enabler ever has access to your private key. This application is completely open source. We even use open source software to safely connect your chosen wallet to our network and monitor for transactions on the blockchain.

Your users do not need to be Web3 knowledgeable to use XRP for Salesforce. With a few button clicks connect your and clients' wallets to begin accepting crypto payments. We track all your payments for you.

We urge Salesforce Administrators to read through this entire document at least twice. If you are new to Web3, some of the details may be unfamiliar to you. If you are experienced with Web3, you may find some of the simplifications we make in the interface shocking. We have focused on making the process as simple as possible.

Table of Contents

Quick Start Guide

For security reasons, we recommend a limited access Integration User with Web3 Enabler Integration Permissions. Because of the sensitive nature of Blockchain Financial Transactions, a clear paper trail of this user is recommended. We recommend using an Integration User, but a Standard User with Admin Profile and the Integration Permission set will also work.

The Integration User needs the following Permissions:

- Salesforce Developer Edition account from Salesforce
- XRPL Webhooks account
- Authorization on the XRP for Salesforce Setup App

Setup instructions:

1. Install XRP for Salesforce.
2. Setup XRPL Webhooks account.
3. Setup XRP for Salesforce.
    a. Create a site for the application use case.
    b. Add Apex class access to the site.
    c. Assign permission set to users who should have access to the XRP for Salesforce (Optional).
4. Configure XRP for Salesforce.
    a. Provide API Key and Secret from XRPL Webhooks.
    b. Provide site domain created during setup.
    c. Register Webhook.
    d. Initialize asset tokens.
5. Use XRP for Salesforce.
    a. Create Accounts.
    b. Create Org Wallets.
    c. Create Account Wallets.
    d. Capture inbound and outbound blockchain transactions.

Detailed explanation for the Integration User: Create the Integration User - We highly recommend the use of a Salesforce Integration user for this program. Requirements of this User are as follows:

1. Enter Setup, and Create Users.
2. Give your new Integration User reasonable names and logins.
3. Use an email you can monitor.
4. Salesforce License: Salesforce Integration.
5. Salesforce Profile: Salesforce API Only System Integrations.
6. After saving, Edit Permission Sets for this user and assign:
    a. XRP for Salesforce Integration User (Managed).

Users will also need appropriate permissions based on their use cases.

Permission Levels: User, Admin

User level settings: XRP for Salesforce User (Managed) Permissions grant access to the tools a typical Salesforce Salesperson or Customer Service Representative needs to manage Web3 exchanges. They can access the Org Wallet addresses, create the Account Wallets for connection and otherwise enable sales to occur.

Admin level Settings: XRP for Salesforce Admin (Managed) Permissions. This user should be familiar with Web3 concepts like blockchains, tokens and contracts. The Admin User can edit any of these settings. If nobody in your organization has this knowledge, the defaults are probably sufficient for your needs.

Best Practices (Cryptocurrencies)

As you and your organization become more comfortable with the world of digital assets, you may branch out. However, for most organizations new to accepting cryptocurrencies, a few simple operations will alleviate risk and simplicity.

Understand the Basics of Public Key Encryption

Public key encryption relies on a series of mathematical equations that connects a private and public key (wallet address). Anyone with access to your public key can send a message to you and only you can decode it. You can “digitally sign” a message with your private key, and anyone with your public key can verify it. This key pairing is among the foundations of cryptocurrencies.

As a result, your organization can publish its public key to receive payments. However, only the possessor of the private key can authorize the “spending” of those coins, by sending a signed message to the network. In common usage, the private key is managed by the “wallet holder”.

Create Policies around Wallet Access

Whoever controls the private keys controls the coins. You should generally have at least two people with access to the wallet to avoid losing your coins. You should decide how much crypto exposure you want to have, and convert to fiat when your coins on hand exceed it. Web3 Enabler uses your public key only, and does not have access to your coins.

User Guide

Working with Accounts and Opportunities

Web3 Enabler tracks payments to and from your Organization and your Accounts. The nomenclature we use is as follows:

- Org Wallet: Your company's wallet(s)
- Account Wallet: The third party Vendor or Customer wallet

Account Wallets are connected to Accounts and link all transactions to those Accounts.

The Account Wallet features a Lookup reference to a Contact, which may be important for automations in your organization.

Transactions

Transactions are referred to as Inbound Blockchain Transactions or Outbound Blockchain Transactions. If a transaction is sent from the address in an Account Wallet (i.e. a customer payment) to your Org Wallet, that is an Inbound Blockchain Transaction. If the transaction starts from your Org Wallet and goes to the Account Wallet, that is an Outbound Blockchain Transaction.

Your Salesforce Admin may further add custom fields to connect Opportunities, Orders, or other Salesforce Objects to transactions.

Receiving a Payment

Your XRP for Salesforce Admin should have set up one or more Org Wallets for you to receive payment.

They will have a wallet address on the Org Wallet that you may be given read access to. Once your client has connected their account wallet, transactions sent between the two wallets will be recognized and recorded by XRP for Salesforce.

Accounting/Finance Guide

If you are closing transactions via Blockchain and your transactions are not integrated with your accounting and/or ERP systems, you can export them from Salesforce and import them into your systems.

Transaction Logs

Transactions are viewable via Inbound Blockchain Transactions and Outbound Blockchain Transactions.

Most commonly, Inbound transactions are revenue and Outbound transactions are refunds, but each transaction and business is unique. XRP for Salesforce will match live transactions as best it can, but your Salesforce Administrator will need to work with you for detailed mappings that match your business usage.

Exporting Transactions

Transaction Reports are included that are exportable as CSV or XLS for import into your accounting and ERP systems.

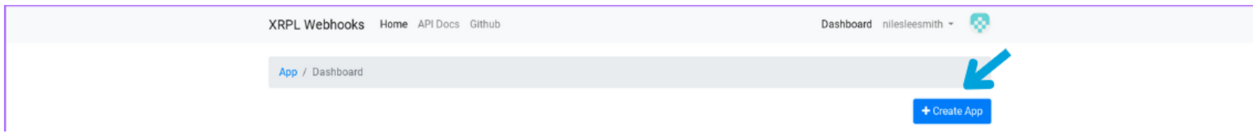
Administrators Guide

Install XRP for Salesforce

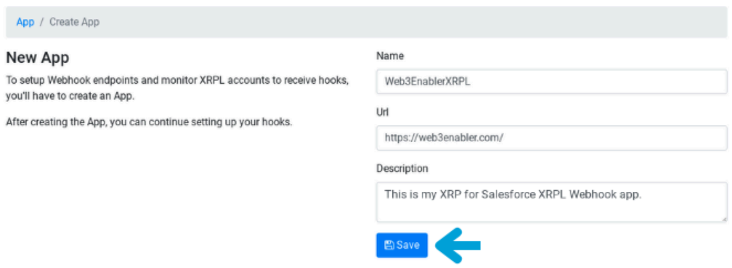
1. Checkout the Main Branch.
2. Log into the Dev Hub Org by running sf org login web --set-default-dev-hub --alias DevHub --instance-url https://login.salesforce.com and entering your username and password.
3. Create a Scratch Org by running sf org create scratch -f ./config/project-scratch-def.json -a dev -d -y 30.
    a. The -f flag is a path to config file (no need to change it).
    b. The -a flag is an alias of the scratch org, if you create multiple scratch orgs you can give them unique aliases to easier refer to them.
    c. The -d flag marks the newly created scratch org as default. If you don't mark it as default you will have to reference it by username or alias, or you will have to use sf config set target-org YourAliasOrUsername to set it as default.
    d. The -y flag sets the number of days before the org expires.
    e. Use the -h flag for help.
    f. For more details: developer docs scratch orgs create.
4. Push the code to the Scratch Org: sf project deploy start
5. Connect to the Salesforce Scratch Org: sf org open

Set up XRPL Webhooks Account

1. Go to XRPL Webhooks.
2. Login or create an account.
3. Create an app.



Click the *Create App* button.


App / Create App

New App

To setup Webhook endpoints and monitor XRPL accounts to receive hooks, you'll have to create an App.

After creating the App, you can continue setting up your hooks.

Name: Web3EnablerXRPL

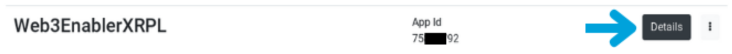
Url: https://web3enabler.com/

Description: This is my XRP for Salesforce XRPL Webhook app.

Save

Enter the app name in the *Name* field. Enter a website address in the *Url* field. Enter a description in the *Description* field. Click the *Save* button.

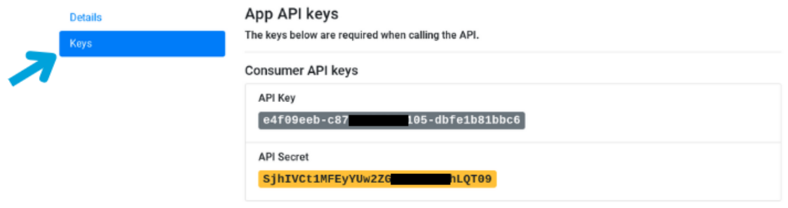
4. View the app API keys.


Web3EnablerXRPL

App Id 75[redacted]92

Details

Click the *Details* button for your app.


Details

Keys

App API keys

The keys below are required when calling the API.

Consumer API keys

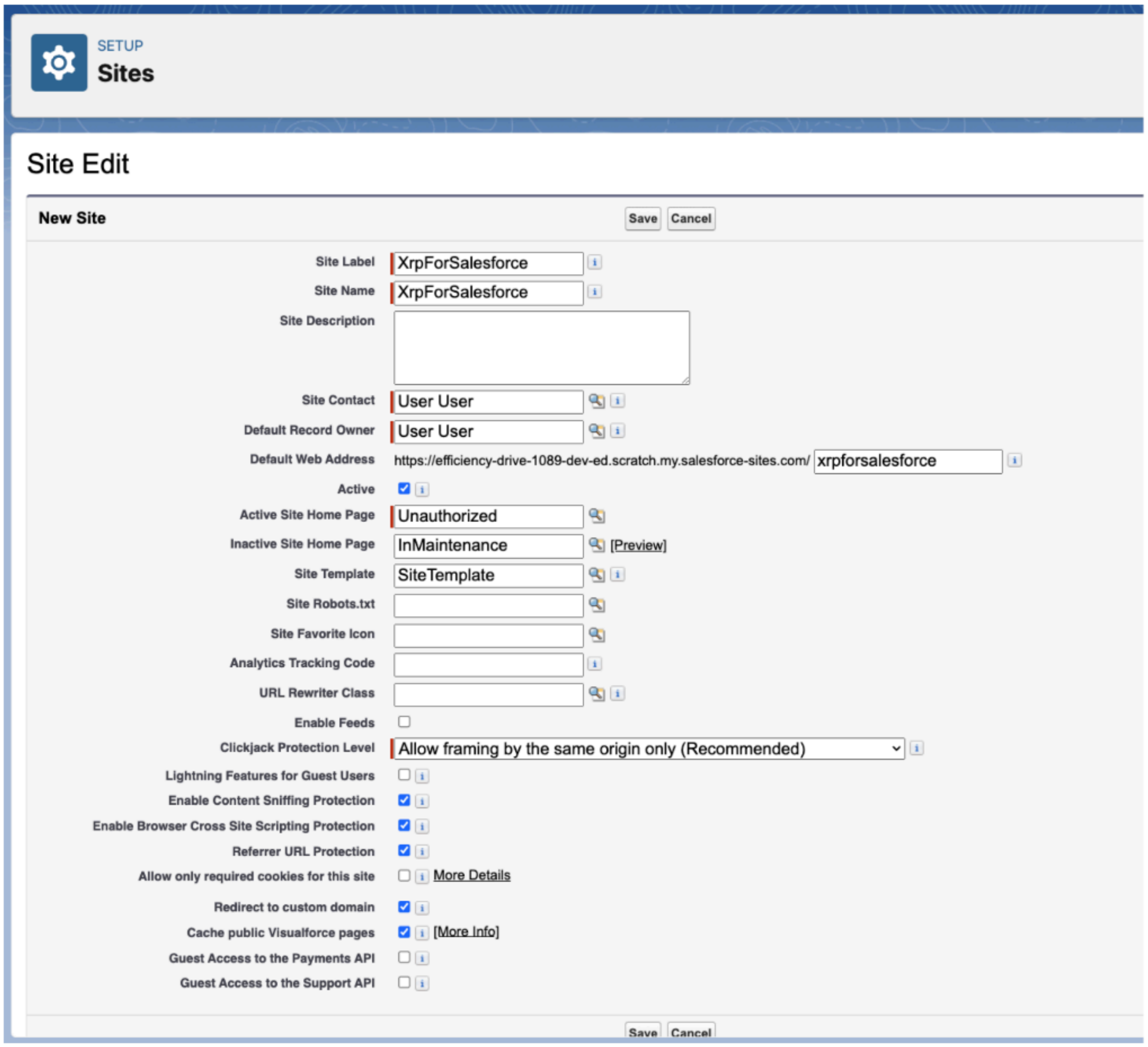
API Key
e4f09eeb-c87[redacted]105-dbfe1b81bbc6

API Secret
SjhIVCt1MFEyYUw2ZG[redacted]hLQT09

Click the *Keys* tab to view your API Key and API Secret.

Set up XRP for Salesforce

1. Go to Setup.
2. Create a new Salesforce site.
   a. Click on the *User Interface* dropdown.
   b. Click on *Sites and Domains* dropdown.
   c. Click on *Sites*.
   d. Register your Salesforce site domain if necessary.
   e. Click the *New* button.


SETUP
Sites

Site Edit

New Site

Site Label	XrpForSalesforce
Site Name	XrpForSalesforce
Site Description	
Site Contact	User User
Default Record Owner	User User
Default Web Address	https://efficiency-drive-1089-dev-ed.scratch.my.salesforce-sites.com/ xrpforsalesforce
Active	✓
Active Site Home Page	Unauthorized
Inactive Site Home Page	InMaintenance [Preview]
Site Template	SiteTemplate
Site Robots.txt	
Site Favorite Icon	
Analytics Tracking Code	
URL Rewriter Class	
Enable Feeds	☐
Clickjack Protection Level	Allow framing by the same origin only (Recommended)
Lightning Features for Guest Users	☐
Enable Content Sniffing Protection	✓
Enable Browser Cross Site Scripting Protection	✓
Referrer URL Protection	✓
Allow only required cookies for this site	☐ More Details
Redirect to custom domain	✓
Cache public Visualforce pages	✓ [More Info]
Guest Access to the Payments API	☐
Guest Access to the Support API	☐

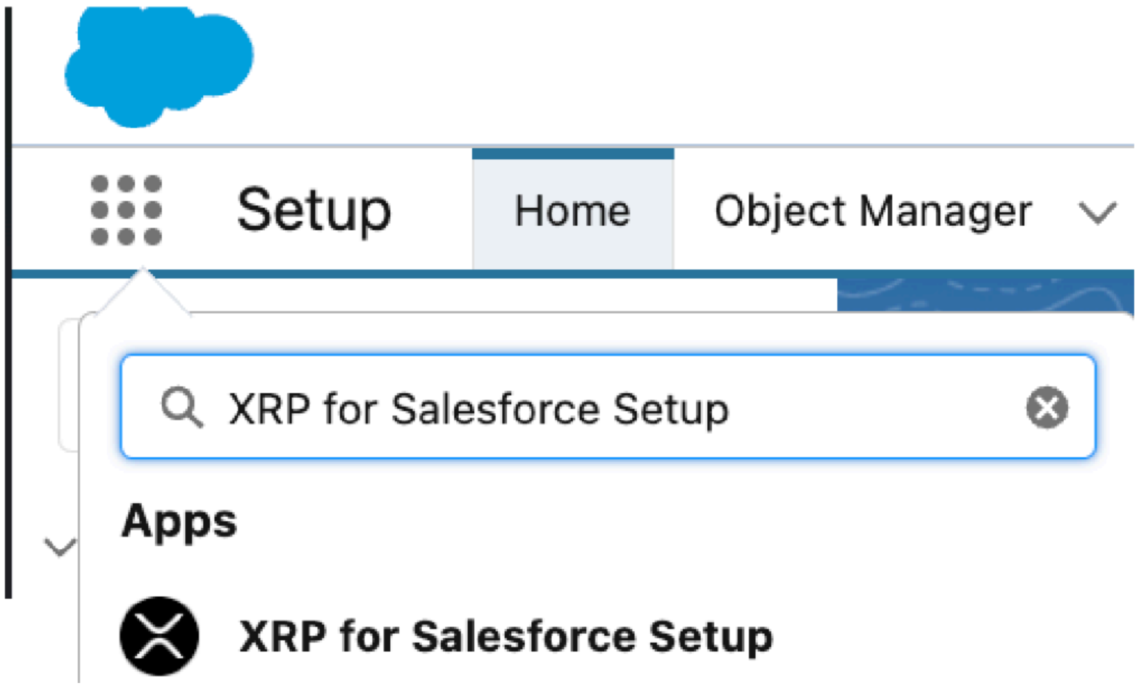
Enter “XrpForSalesforce” in the *Site Label* field. Enter “XrpForSalesforce” in the *Site Name* field. Select a *Site Contact* user. Select a *Default Record Owner* user. Enter “xrpforsalesforce” in the *Default Web Address* field. Check the *Active* checkbox. Set the *Active Site Home Page* to Unauthorized. Copy the *Default Web Address* provided. Click the *Save* button.

3. Enable Apex class access.
   a. Select the site.
   b. Click the *Public Access Settings* button.
   c. Click the Enable Apex Class *Edit* button.
   d. Add the site to *Enabled Apex Classes*.
   e. Click the *Save* button.
4. Assign XRP for Salesforce permission set (Optional).
   a. Click the *User* dropdown.
   b. Click on *Permission Sets*.
   c. Click on *XRP Admin Permission Set*.
   d. Click the *Add Assignment* button.

    e. Select the user.
    f. Click the *Save* button.

Configure XRP for Salesforce

1. Open XRP for Salesforce Setup from App Launcher.



Click on the App Launcher. Search “XRP for Salesforce Setup” in the search bar.

2. Enter credentials.
    a. Enter the XRPL Webhook API Key in the *API Key* field.
    b. Enter the XRPL Webhook API Secret in the *API Secret* field.
    c. Enter the Salesforce site domain in the *Site Domain* field.
    d. Click the *Save* button.
3. Register the Webhook.
    a. Click the *Register Webhook* button.
4. Initialize the asset tokens.
    a. Click the *Initialize Asset Tokens* button.
5. Add webhook to XRPL Webhooks app.
    a. Click the triple dot icon on your XRPL Webhook app.
    b. Click *Manage webhooks*.
    c. Enter your Salesforce site domain in the *URL field*.
    d. Click the *Add* button.

Using XRP for Salesforce

1. Open the XRP for Salesforce from the App Launcher.
2. View Asset-Tokens.

   a. Click on the *Asset-Tokens* tab. Note that since you've registered the XRPL Webhook and initialized the asset-token, the Asset Token Name will be automatically set to *XRP (FKA Ripple)*, and the Asset will be automatically set to *XRP*.
3. Create an Account.
   a. Click the *Accounts* tab.
   b. Click the *New* button.
   c. Enter the name in the *Account Name* field. Click the *Save* button.
4. Create an Org Wallet.
   a. Click on the *Org Wallets* tab.
   b. Click the *New* button.
   c. Enter name in the *Wallet Name* field. Enter address in the *Wallet Address* field. Click the *Save* button.
5. Create an Account Wallet.
   a. Click on the *Accounts Wallet* tab.
   b. Click the *New* button.
   c. Select an account in the *Account* field. Enter name in the *Wallet Name* field. Click the *Save* button.
6. Create an Inbound Blockchain Transaction.
   a. Click the triple dot icon on the webhook you added to your *XRPL Webhooks* app.
   b. Click *Manual Trigger*.
   c. This will send an HTTP POST request to the URL specified in the webhook. Once you have done this, check to confirm that the following updates have been made.
      i. A subscription, associated with the wallet address set in your XRP for Salesforce Org Wallet, has been added for that Webhook: Check by clicking the triple dot icon next to your XRPL Webhooks app, followed by *Manage subscriptions*.
      ii. The *XRP for Salesforce Org Wallet* has been connected: Check this by ensuring that the *Wallet Connected* checkbox in your *XRP for Salesforce Org Wallet* is now checked.
7. View an Inbound Blockchain Transaction.
   a. Click on the *Inbound Blockchain Transactions* tab.
   b. Click on the *Inbound Blockchain Transaction Id*.
   c. Your first *Inbound Blockchain Transaction* will appear here once you've added your first *Manual Trigger* from your XRPL webhook.
8. Create an Outbound Blockchain Transaction.
   a. Disconnect your XRP for Salesforce Org Wallet by clicking *Disconnect Wallet* in your *XRP for Salesforce Org Wallet*.
   b. Swap the wallet address in your *XRP for Salesforce Org Wallet* with the wallet address in your *XRP for Salesforce Account Wallet*.
   c. Create a second XRPL Webhook Trigger by clicking the triple dot icon on the webhook you added to your XRPL Webhooks app, and then clicking *Manual Trigger*. This will send another HTTP POST request to the URL specified in the

webhook. Confirm that this has successfully reconnected your *Salesforce Org Wallet* by ensuring the following updates have been made.

   i. A subscription, associated with the newly set wallet address in your XRP for Salesforce Org Wallet, has been added for that Webhook: Check by clicking the triple dot icon next to your XRPL Webhooks app, followed by *Manage subscriptions*.
   ii. The *Wallet Connected* checkbox in your *XRP for Salesforce Org Wallet* is now checked again.

9. View an Outbound Blockchain Transaction.
   a. Click on the *Outbound Blockchain Transactions* tab.
   b. Click on the *Outbound Blockchain Transaction Id*.
   c. Your first *Outbound Blockchain Transaction* will appear here once you've added your second *Manual Trigger* from your XRPL webhook.

Salesforce Admin Primer on Cryptocurrencies

Many Salesforce Admins may only have a cursory understanding of cryptocurrencies and digital assets when asked to embark on this process. This primer is designed to provide some basic terminology and understanding.

Definitions

Blockchain - A distributed ledger (series of transactions) stored in data elements called blocks. These blocks contain references to the prior blocks, creating a "chain" of data. The blockchain costs resources to maintain. The maintainers are compensated for validating or mining.

Coin - The native digital asset of a blockchain. It is used to pay for transactions.

Transaction - An entry on the blockchain.

Private Key (Seed phrase) - A master password created specifically for crypto assets to be released from a wallet. The private key may be displayed as an alphanumeric string or a set of words (usually 12 or 24) known as a seed phrase corresponding to the private key. Whoever possesses the private key has control over the crypto assets in the crypto wallet. NEVER SHARE YOUR PRIVATE KEY.

Public Key (Wallet address) - An alphanumeric address shared with the blockchain to denote sending and receiving parties. A public key is derived from the private key. A public key and transactions are public and viewable on the blockchain.