

Open SSO Documentation

Description

Open SSO is an User Management System with Single Sign On. Easier to manage your user and make them able to sign-in to all of your applications.

Open SSO was designed for high scalable purpose. Built with [Fastify Framework](#), one of the fastest JavaScript framework. Open SSO is very small application (only user management system) and isolated, so this will make you easier to integrate to your current web/application.

Open SSO use `denormalise` database to achieve high performance speed for read and write also to make easier for migration to clustered or sharding database. To reduce the request into database, there is built-in cache using `memory` as default but easily to swith to `redis` in the future.

About authentication security, we follow the JWT standard `Oauth 2.0` as describe in [RFC7519](#) and [RFC9068](#). Very fast performance because everything is stateless.

Documentation

Below here is the full documentation about Open SSO project.

Open SSO Documentation	1
Get Started	1
A. Requirement	2
B. Restore Database [Optional, you can skip this part]	2
C. Build App and Run it	3
D. Update	3
E. Deploy to Server	3
Learn more	8
Help	8
Credits	9
Changelog	9

Get Started

Open SSO is a NodeJS application. Make sure you've done already to install the NodeJS server on your server or on your local computer.

A. Requirement

Minimum requirement to install Open SSO project:

- NodeJS min v.14.19.3 - [Download NodeJS here](#).
- Database [Optional]
- Redis [Optional]

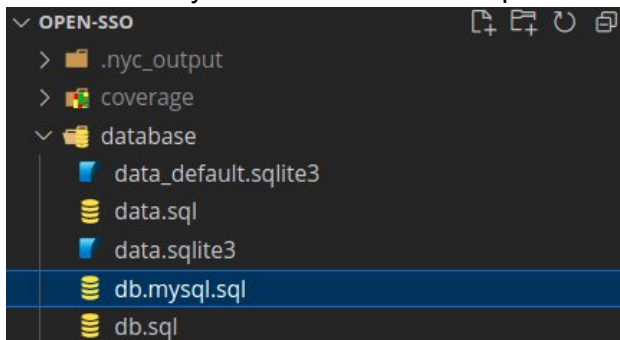
Note:

- The default database is using SQLite, but you can switch to another database like MariaDB, MySQL or PostgreSQL for easy monitoring, maintaining and scaling in the future.
- The default cache engine is using memory, but you can switch to Redis easily from config file. Using Redis is optional but it will become required if your application working on container service, clustered or behind the proxy load balancer.

B. Restore Database [Optional, you can skip this part]

This is optional, Open SSO using SQLite as default and all features works well with it. You need to setup database if you have a plan to run Open SSO using MariaDB or MySQL database server. You can skip this if SQLite is enough for you.

1. Install **MariaDB** or **MySQL**. You can search how to install this by using Google search engine.
2. Create a New Database, then execute or import the file **db.mysql.sql** located inside directory database into your database. See the picture below.



3. Edit **config.js** like this below

```
12   sequelizeOption: {
13     dialect: 'mariadb', /* one of 'mysql' | 'mariadb' | 'postgres' | 'mssql' */
14     dbname: 'DATABASE_NAME',
15     username: 'DATABASE_USER_NAME',
16     password: 'DATABASE_USER_PASSWORD',
17     options: {
18       host: 'DATABASE_HOST_OR_SERVER',
19       port: 'DATABASE_PORT'
20     }
21     // dialect: 'sqlite',
22     // storage: './database/data.sqlite3'
23   },
24 }
```

4. Done.

Note:

- File **db.mysql.sql** is only works for MariaDB and MySQL only.
- File **db.postgre.sql** is only works for PostgreSQL only.
- Working with **MSSQL** should works too, but we don't create the database structure for it, so you have to create it by yourself.

C. Build App and Run it

You have to build the app for the first time.

1. Go to the directory path then typing.
npm install
2. Wait until the process complete.
3. Then you can run the app by typing
node server.js
4. Now, Open the url <http://localhost:3000>.

Note:

- There is no default account, so when you're running this application for the first time, you have to register a new account to get the admin role.

D. Update

Open SSO is using many dependencies, please try to update some security patch before you deploy Open SSO to your server. This would be better if you repeat doing this periodically for every 6 months.

Update App

1. Go to the directory path then typing
npm update
2. Done.

E. Deploy to Server

If you finished doing from point A to D, now we prepared to deploying to server. In this case, I'll using **cPanel** and **SQLite** database because it's very cheap and easy.

a. Preparation

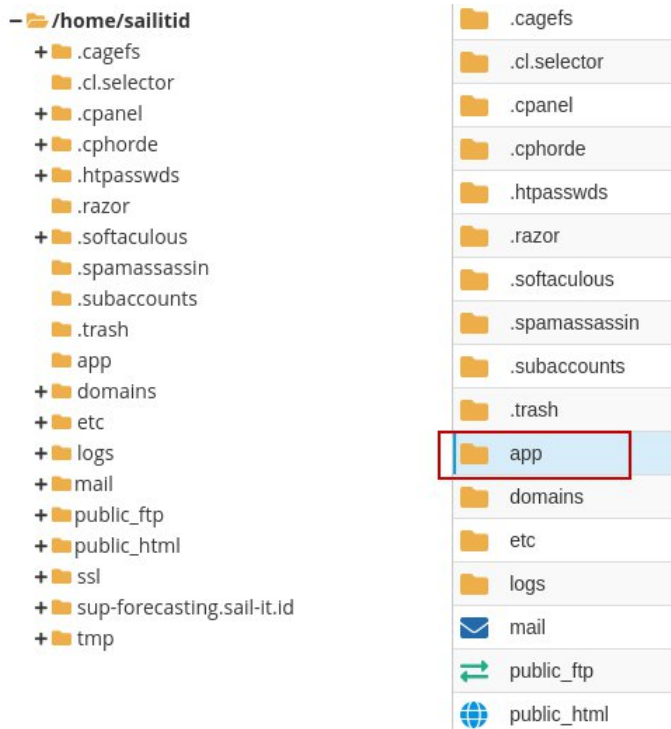
1. Edit the **config.js** file, then change this part with your domain or subdomain, in this case I'll using my subdomain **sso.sail-it.id**.

```
// WEBSITE
// This is the base url, used for template engine
// Change with your domain without trailing slash.
// Example : https://yourdomain.com
baseUrl: 'https://sso.sail-it.id',
baseAssetsUrl: 'https://sso.sail-it.id',
siteName: 'Open SSO',
```

2. Then save it.
3. Now select all files then compressed to zip, give it name **app.zip**.
4. Done.

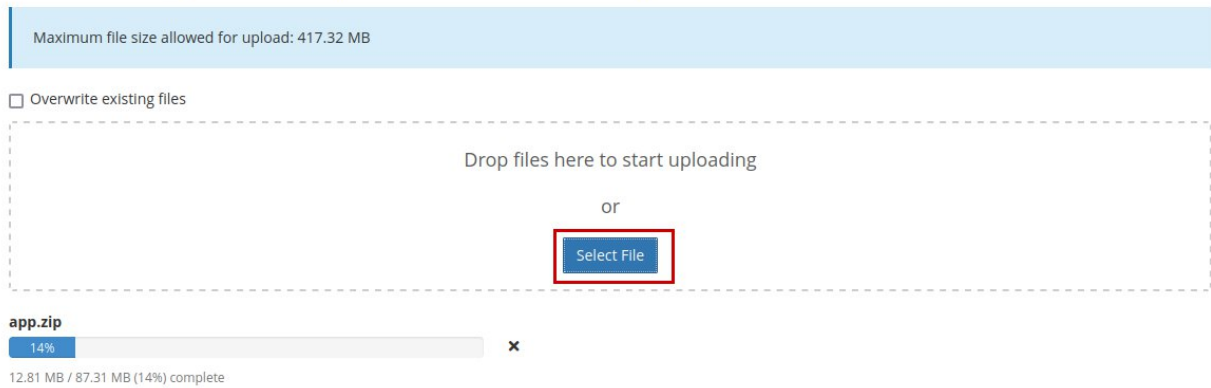
b. Upload to hosting

1. Go to your **cPanel** and look for **File Manager**.
2. Create new folder name **app** on root.



3. Go inside **app** folder, then click **Upload** button at top left. Then you will be redirected to the new page for upload, now choose your zip file and start upload it.

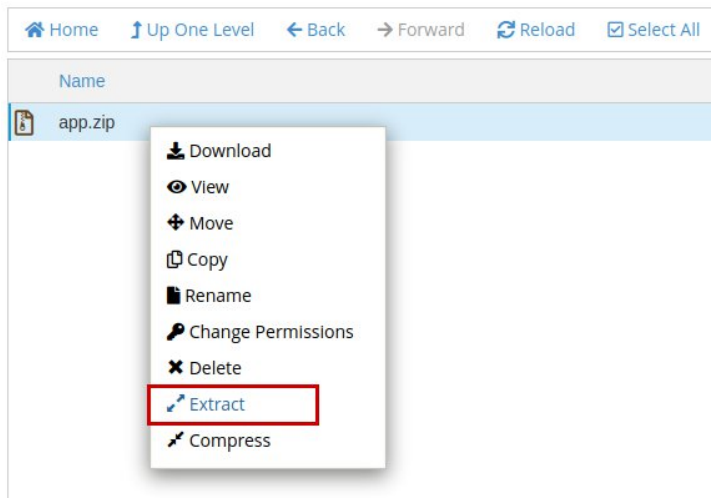
Select the file you want to upload to “/home/sailitid/app”.



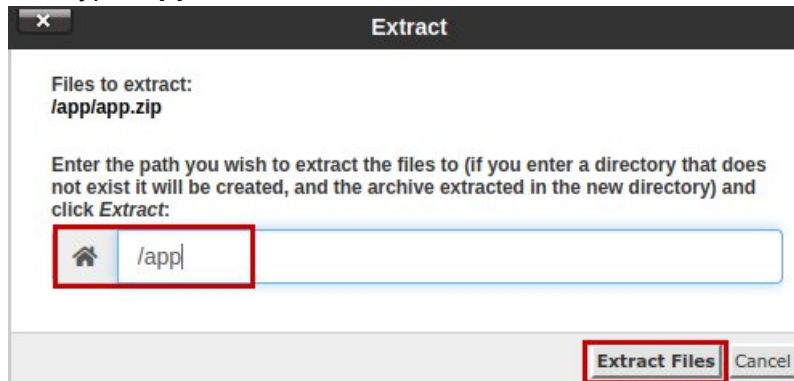
When the upload complete, you can close it page and back to **File Manager**.

4. Click **Reload** button then you will see your **app.zip** file on there.

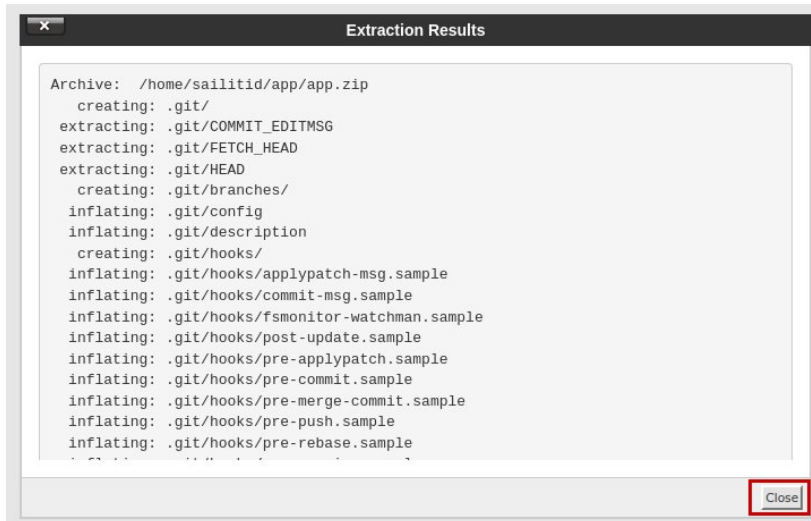
Now right click on that file, then choose **Extract**.



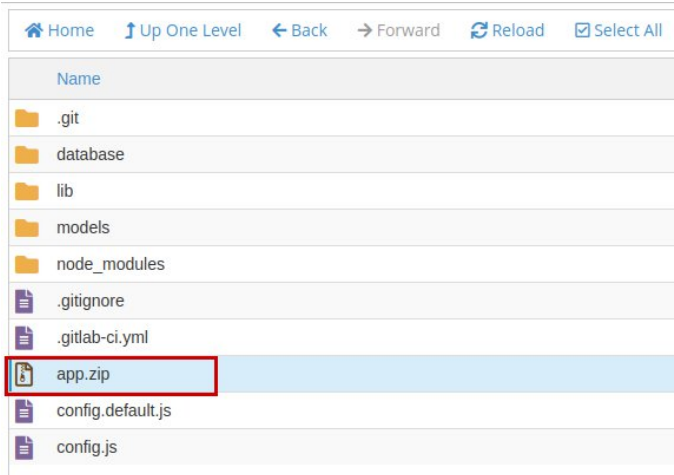
Just type **/app** on it, then click **Extract Files**.



When the extract successfully, it will shown like this below



Just click **Close** and click **Reload** button again. Then you will see app files already extracted.



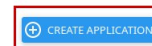
5. Done, you can close the **File Manager** page.

c. Setup NodeJS App

1. Go to your **cPanel** again.
2. Look for **Setup NodeJS App** then click **CREATE APPLICATION** button.



WEB APPLICATIONS



3. Configuration should look like this below, then click **CREATE** button

WEB APPLICATIONS [+ CREATE APPLICATION](#) CANCEL CREATE

Node.js version: 14.20.1

Application mode: Development

Application root: app

Application URL: sso.sail-it.id

Application startup file: server.js

Passenger log file: /home/sailitid/

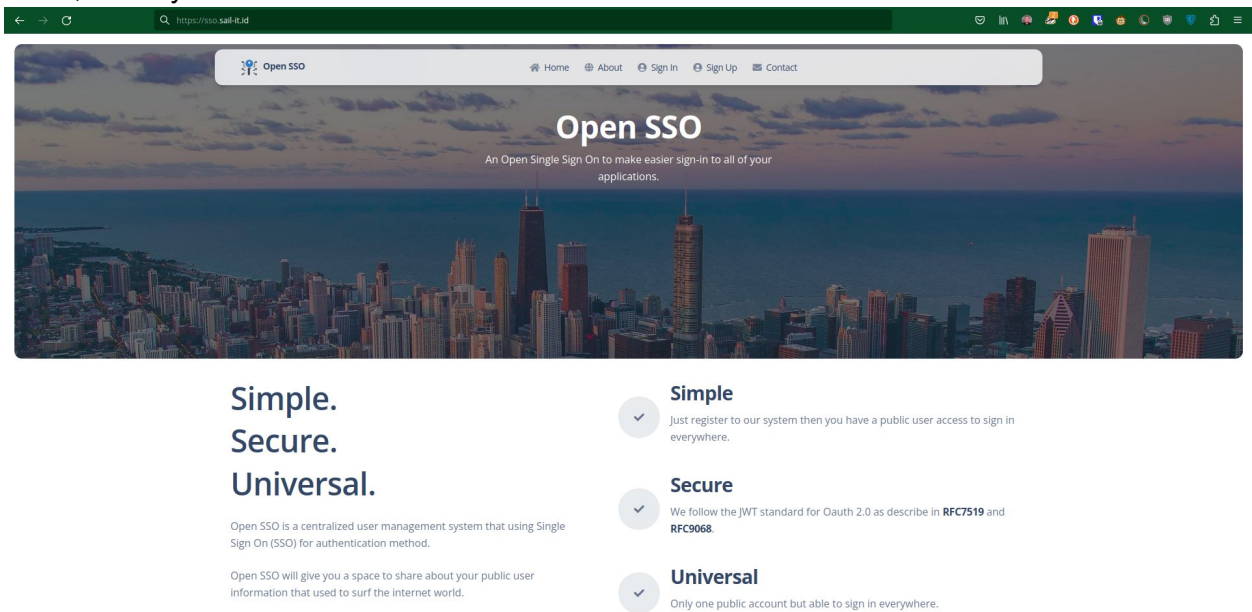
4. Then click **Start App**, if there is no error, the button will display like this below

WEB APPLICATIONS [SSO.SAIL-IT.ID/](#) DESTROY CANCEL SAVE

Enter to the virtual environment. To enter to virtual environment, run the command: `source /home/sailitid/nodeverw/app/14/bin/activate && cd /home/sailitid/app`

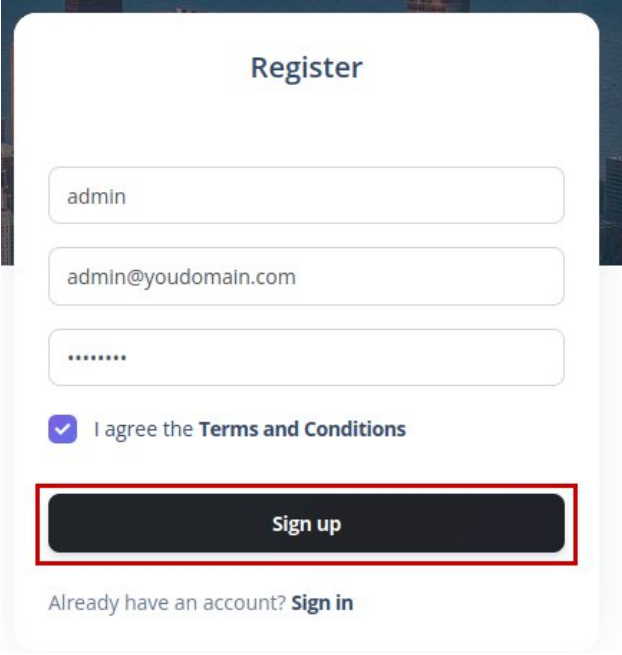
Node.js STOP APP RESTART

5. Done, now try to visit the website.



F. Create Admin User

1. Just go to **Sign Up** page.
2. Fill the form and click **Sign Up** button.



The image shows a registration form titled "Register". It contains the following fields and elements:

- A text input field containing "admin".
- A text input field containing "admin@youdomain.com".
- A password input field with masked characters "*****".
- A checkbox that is checked, with the text "I agree the **Terms and Conditions**".
- A dark grey button with the text "Sign up" in white, which is highlighted with a red rectangular border.
- At the bottom, the text "Already have an account? **Sign in**".

3. Done, the first user will get admin user automatically.

Learn more

Open SSO is having many features, if you want to learn more deeply about Open SSO project, please see:

- [Advanced Configuration Documentation](#)
- [API Documentation](#)

Help

If this documentation didn't help to your specific problem, just feel free sending email to aalfiann@gmail.com.

We always try to update this documentation in the future. If you want to get the faster update, please check our [online documentation](#).

Credits

All Open SSO libraries or external scripts is using an open source library.

Frontend we using:

- [Argon Dashboard 2](#)
- [Bootstrap 5](#)
- [reCaptcha V3](#)
- [Dom.js](#)
- [Axios Http](#)
- [Browser Storage Class](#)
- [Fly Json ODM](#)
- [Momentjs](#)
- [Native Form Validation](#)
- [ReefJS V12](#)
- [Sweetalert](#)

Note:

- For backend, you can see the dependencies at package.json.
- There is no jQuery in this application, but you are able to add it by yourself if you need it.

Changelog

- **v1.0.0 - 03/2023**
 - First Release