

BigLedger Handbook

Vincent Lee <vincent@akaun.com>, Diana Filimonova <diana@wavelet.net>

Version 1.0, 2022-05-15



This is the public guide that provides all public use cases of the Akaun Platform. To view this in other formats, click on the following:

- [Epub Format](#)
- [Single File HTML](#)
- [PDF Format](#)

Table of Contents

1. Guide by User Types	3
1.1. General Users.....	3
1.1.1. Your AkaunID.....	3
1.1.2. The Production, Staging and Development Environment.....	3
1.1.3. Understanding Root vs Tenant.....	3
1.2. Project Implementors	5

Chapter 1. Guide by User Types

This repository contains all the source code for publishing the Akaun Platform Public Documentation.

1.1. General Users

The very first thing that every akaun-platform users would need to have, is to sign up for and AkaunID. You can do so by visiting <https://akaun.com/>

1.1.1. Your AkaunID

It is important for you to understand that with a single AkaunID, it is possible for you to have multiple Sign-In methods. You could be using an email with password, a phone number with the same password, you could also be adding access keys to your AkaunID for integration purposes.

1.1.2. The Production, Staging and Development Environment

The very first thing to know is to understand the usage of each of the following environments.

Environment	URL	Custom Domain	Description
Production	https://akaun.com	<a href="https://<something>.akaun.net">https://<something>.akaun.net	This is the life environment, used by real businesses in their day to day operations.
Staging	https://akaun.com	<a href="https://<something>.akaun.app">https://<something>.akaun.app	This environment is usually used for testing by the customers and project managers. New features are tested by the end users here before it gets deployed to the production environment.
Development	https://akaun.dev	<a href="https://<something>.akaun.xyz">https://<something>.akaun.xyz	This environment is usually used by the software developers to do their experiments.

If you have registered an AkaunID to login to the Production environment, that doesn't mean you have the same AkaunID registered in the Staging Environment and vice versa.

1.1.3. Understanding Root vs Tenant

Within a single environment (Production vs Staging vs Development) there's one and only one root database, and there could be MANY tenant databases. Each user could be joining MULTIPLE tenants

and they could have different "RANK" when joining each tenant, for example, OWNER, ADMIN, MEMBER, GUEST, VISTOR.

Rank of a users in a Tenant	Description
OWNER	The OWNER of a tenant is able to add any applets to an "applet catalog" if the person is also the owner/admin of the "applet catalog". The OWNER is able to add, remove or promote another user in the tenant to be the OWNER of the tenant. This RANK is usually granted to the business owners or MIS Manager of a company.
ADMIN	The ADMIN of a tenant can do almost everything that the OWNER can do, except adding , removing or promoting another user in the tenant to become the OWNER. This role is usually granted to the MIS Executive or managers in a company.
MEMBER	The MEMBER Rank does not entitle a user to administer the Tenants , Applet Catalogs, they can however, become the OWNER/ADMIN or MEMBER of an Applet after they installed the specific applet. It is important to understand that the "OWNER/ADMIN of the Tenant" is NOT the same as the "OWNER/ADMIN" of the "Applet"

1.2. Project Implementors

Implementation of BigLedger projects requires a deep understanding of the Akaun platform features and functions. Depends on which module the business

some sections here

==== Section 2

=== Tenant Owners

Platform Guide for Tenant Owners

==== Section 1

==== Section 2

=== Applet Developers

Guid for applet Developers

==== Signing up your bigledger account

=== Reseller Guide

Guide for resellers

==== Section 1

==== Section 2

=== System Integrators

Guide for system integrators

==== Section 1

some sections here

==== Section 2

== Guide by Industry

This guide by industry provide :

- use cases
- case study
- key paint points and feature of each industry

=== Consumer Electronics Industry

==== Problems

==== Solutions

==== Features

=== Pharmacy Industry

==== Problems

==== Solutions

==== Features

=== Car Workshops

==== Problems

==== Solutions

==== Features

=== Retail Chainstores

==== Problems

==== Solutions

==== Features

=== Logistics Industry

==== Problems

==== Solutions

==== Features

=== Manufacturing Industry

==== Problems

==== Solutions

==== Features

=== Pet Industry

==== Problems

==== Solutions

==== Features

== Guide by Modules

The Akaun platform is like lego blocks, users will be able to mix and match different applets to formulate a new module, even if it is not in the list below.

=== Core Module

==== Overview

==== Benefits

==== Related Applets

==== Pricing

=== Accounting Module

==== Overview

==== Features

=== Point of Sales Module

=== Manufacturing Module

=== Membership Module

=== Multilevel Marketing Module

=== Unified Contact Center Module

=== Inventory Module

=== EcomSync Module

=== Delivery Module

=== Cross Platform Commerce

=== Procure to Pay Module

=== Order To Cash Module

== Applet Directory

=== Applet Listing

== Cloud Native Architecture

Common misunderstanding * thinking web based = cloud based. * everything has A.I. * online / networked = cloud based

Our aspirations * To be the business operating systems

=== Background

- Outdated JBoss application server, EJB 1.0 / 2.0
- Servers idling
- Hackers attack
- Certain minimum costs to power up a tenant
- Self service
- Internal employee users vs customers + suppliers
- Integration pains

=== Motivation

Platform Business Model

Multiple offerings → as middleware - backend as a service → as applets → as SAAS → as communication hub

Cross Platform development

Scalability

Flexibility

Growth

Composable Enterprise

Utility based billing

==== Integrations

spikes in loads

==== Infrastructure and Toolchain

Various tools ready to be used out of the box... example : OCR, Lambda, RDS, NoSQL, API Gateway

==== Network Effect

==== Cloud Cost Savings

==== Artificial Intelligence

==== Human Resource

=== Journey

=== Challenges and Lessons

=== Technical Considerations

- managing complexities
- Sharing about our journey to transform from monolith architecture to cloud native architecture
- Sharing about the problems and challenges implementing stateless applications
- Sharing about the implementation of Micro-Frontend Architecture
- Sharing about the implementation of Web Socket using API Gateway and DynamoDB
- Sharing about challenges of securing the API, permissions / configurations etc
- Sharing about the challenges of multi-tenant environments
- Sharing about the design of composable enterprise architecture
- Sharing about the tricky part of networked tenants

== Documentation Authoring

=== Authoring this Documentation

This documentation is written using asciidoc, please take a few minutes to go through the following guide:

- <https://docs.asciidoctor.org/asciidoctor/latest/>
- <https://docs.asciidoctor.org/asciidoc/latest/>

Some useful videos:

- <https://www.youtube.com/watch?v=KbmpBZO4EoY>
- <https://www.youtube.com/watch?v=GL1SEYOhEiQ>
- https://www.youtube.com/watch?v=wca5m7JpS_U

=== Setup and Configurations

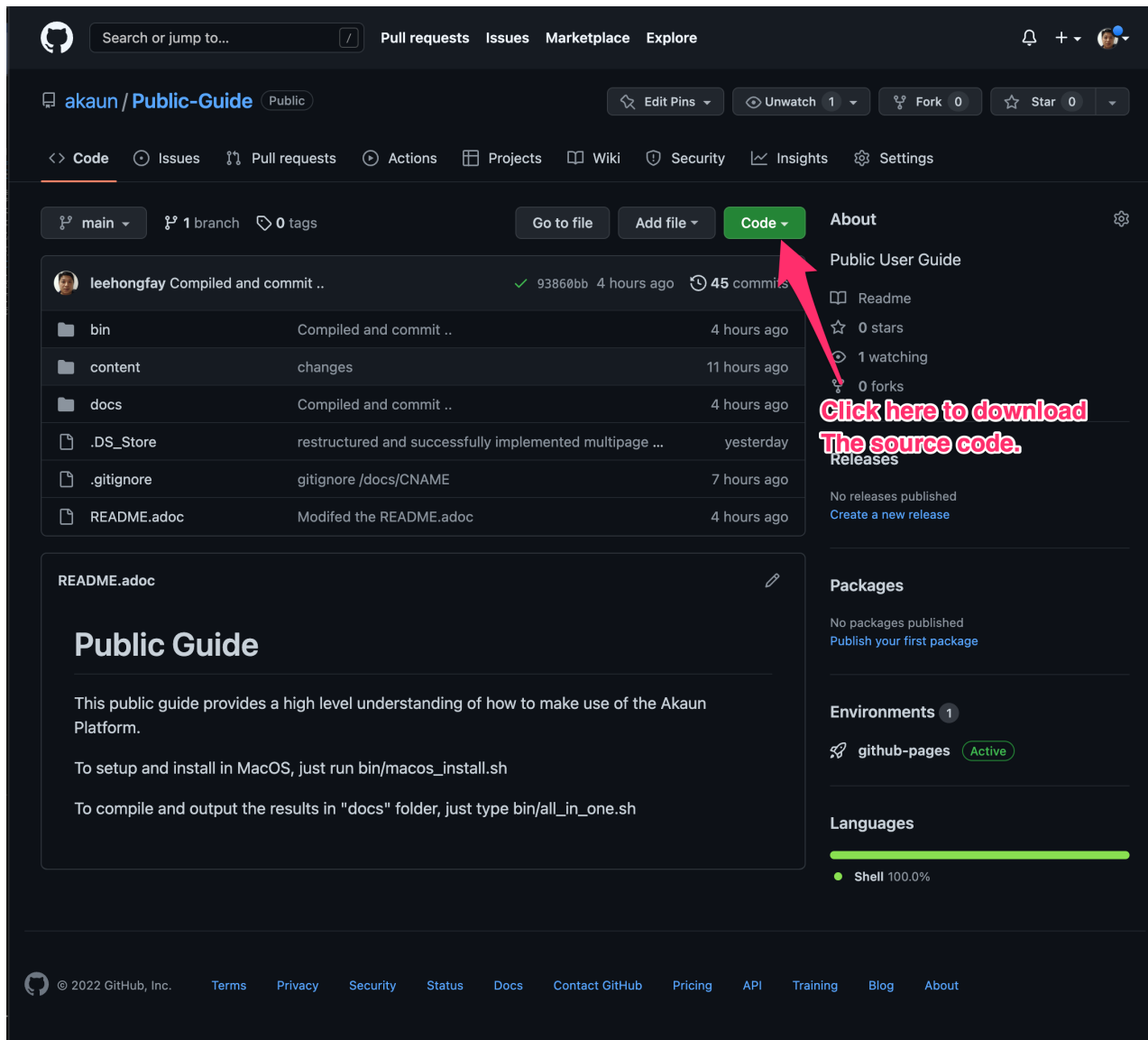
==== Github Repository

- Github User Account

In order to author this documentation, you will need to have a <https://github.com> account. As an employee of Wavelet.net or BigLedger.com , you will be using the <username>@wavelet.net to register for the github user.

You can request the HR department to grant you access to the hr@wavelet.net to grant you the access to Github Akaun organization (<https://github.com/akaun>).

- Downloading the Github Desktop Client by clicking on the button below:



Please visit <https://desktop.github.com/> to download the Github Desktop.

- Accessing the repository

After you have downloaded the github, please clone the <https://github.com/akaun/Public-Guide> to your local computer.

==== AsciiDoctor Toolchain

Depends on the operating systems you are using, please follow the instructions below accordingly:

===== Installing AsciiDoctor

- Linux
 - <https://docs.asciidoctor.org/asciidoctor/latest/install/linux-packaging/>
- Windows
 - <https://docs.asciidoctor.org/asciidoctor/latest/install/windows/>

- Mac

- <https://asciidoctor.org/docs/install-asciidoctor-macos/>
- <https://docs.asciidoctor.org/asciidoctor/latest/install/macos/>

===== Installing Epub3 Converter

Please click on the link below for detailed instructions:

<https://docs.asciidoctor.org/epub3-converter/latest/>

===== Installing PDF Converter

<https://asciidoctor.org/docs/asciidoctor-pdf/#install-the-published-gem>

===== Installing Multipage HTML Converter

<https://github.com/owenh000/asciidoctor-multipage>

===== Visual Studio Code

- Downloading Visual Studio Code

Please visit the following page to download. <https://code.visualstudio.com/download>

After you have installed the Visual Studio Code, please install the following extensions:

- AsciiDoc
- Github Pull Requests and Issues
- vscode-pdf

The next step is to open the folder where you cloned the <https://github.com/akaun/Public-Guide> repository using the Visual Studio Code

=== Understanding the Directory Structure

Folder Name	Description
bin	This folder contains shell scripts that you need to install and setup. You will also see a file called <code>publish.sh</code> , and this is the file that you would execute to compile the asciidoc into various file formats, and save them in the "docs" folder. You shouldn't be editing any files in this folder.
build	This is a temporary folder that gets deleted and re-created every time we "compile" the documentation. You shouldn't need to mess with files in this folder.

content	This is where you will be contributing by editing the files in this folder.
docs	You shouldn't need to edit any files in this folder. But you are welcome to open the files in this folder to check the "compiled output" of your changes , to ensure they are correct before pushing your code to the github, and request for your branch to be merged with the "main" branch in github.

=== Committing Changes To The Repository

Before you make any changes, it is absolutely important that you **checkout** the local repository into a different branch, and the naming convention that we use would be:

YYYY-<email_address>

You may find the following resource useful:

- <https://code.visualstudio.com/docs/editor/versioncontrol#:~:text=You%20can%20create%20and%20checkout,tags%20in%20the%20current%20repository.>
- <https://www.youtube.com/watch?v=X9-iaXfKY5g>
- <https://www.youtube.com/watch?v=H5BLEPhqxe8>
- <https://www.jasongaylord.com/blog/2020/07/08/create-delete-branches-using-visual-studio-code>

Once you have made the changes you want, remember to run the command from the "Public-Guide" directory:

```
bin/publish.sh
```

The above shell script will clean up the "docs" and "build" folders, and then compile all the adoc files into the intermediate "build" folder and subsequently publish the final output into the "docs" folder.

To view the final output, you can open the "docs" folder and view the content in the folder by opening one of the following files:

- index.html (This is the multipage html output)
- blg_handbook.pub
- blg_handbook.pdf
- blg_handbook.html

Once you have completed your editing, the next step is to create a **pull request**. Refer to the following for more information:

- <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/getting-started/about-collaborative-development-models>
- <https://docs.github.com/en/pull-requests>
- <https://www.youtube.com/watch?v=oa1wXWeH1IQ>

Congratulations !!

