

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. User Guide	5
1.1. General Users	5
1.1.1. Your AkaunID	5
1.1.2. The Production, Staging and Development Environment	5
1.1.3. Understanding Root vs Tenant	5
1.1.4. The Applet Store	6
1.2. Project Implementors	7
1.2.1. Subscription setup in Akaun Platform	7
1.2.2. Tenant setup in Akaun Platform	7
1.2.3. Applet Catalogs	7
1.2.4. Configuration of Core Modules	7
1.3. Tenant Owners	8
1.3.1. Section 1	8
1.3.2. Section 2	8
1.4. Applet Developers	9
1.4.1. Signing up your bigledger account	9
1.5. Reseller Guide	10
1.5.1. Section 1	10
1.5.2. Section 2	10
1.6. System Integrators	11
1.6.1. Section 1	11
1.6.2. Section 2	11
2. Guide by Industry	12
2.1. Consumer Electronics Industry	12
2.1.1. Problems	12
2.1.2. Solutions	12
2.1.3. Features	12
2.2. Pharmacy Industry	12
2.2.1. Problems	12
2.2.2. Solutions	12
2.2.3. Features	12
2.3. Car Workshops	12
2.3.1. Problems	12
2.3.2. Solutions	12
2.3.3. Features	12
2.4. Retail Chainstores	12
2.4.1. Problems	12
2.4.2. Solutions	12

2.4.3. Features	12
2.5. Logistics Industry	12
2.5.1. Problems	13
2.5.2. Solutions	13
2.5.3. Features	13
2.6. Manufacturing Industry	13
2.6.1. Problems	13
2.6.2. Solutions	13
2.6.3. Features	13
2.7. Pet Industry	13
2.7.1. Problems	13
2.7.2. Solutions	13
2.7.3. Features	13
3. Guide by Modules	14
3.1. Core Module	14
3.1.1. Overview	14
3.1.2. Benefits	14
3.1.3. Related Applets	14
3.1.4. Pricing	14
3.2. Accounting Module	14
3.2.1. Overview	14
3.2.2. Features	14
3.3. Point of Sales Module	14
3.4. Manufacturing Module	14
3.5. Membership Module	14
3.6. Multilevel Marketing Module	14
3.7. Unified Contact Center Module	14
3.8. Inventory Module	14
3.9. EcomSync Module	14
3.10. Delivery Module	14
3.11. Cross Platform Commerce	14
3.12. Procure to Pay Module	15
3.13. Order To Cash Module	15
4. Applet Directory	16
4.1. Applet Listing	16
5. Cloud Native Architecture	17
5.1. Background	17
5.2. Motivation	17
5.2.1. Integrations	17
5.2.2. Infrastructure and Toolchain	17
5.2.3. Network Effect	17

5.2.4. Cloud Cost Savings	18
5.2.5. Artificial Intelligence	18
5.2.6. Human Resource	18
5.3. Journey	18
5.4. Challenges and Lessons	18
5.5. Technical Considerations	18
6. Technical Guide	19
6.1. Akaun Shell Development Guide	19
6.2. Cloud Infrastructure	19
6.3. BigLedger Database Guide	19
6.3.1. Overview	19
6.3.2. Database Schema	19
6.3.3. Liquibase	19
6.3.4. The Root Database Schema	19
6.3.5. The Tenant Database Schema	19
7. Documentation Authoring	20
7.1. Authoring this Documentation	20
7.2. Setup and Configurations	20
7.2.1. Github Repository	20
7.2.2. AsciiDoctor Toolchain	21
Installing AsciiDoctor	21
Installing Epub3 Converter	22
Installing PDF Converter	22
Installing Multipage HTML Converter	22
7.2.3. Visual Studio Code	22
7.3. Understanding the Directory Structure	22
7.4. Committing Changes To The Repository	23
Glossary	25
Appendix A: First Appendix	26
A.1. First Subsection	26
A.2. Second Subsection	26
Appendix B: Second Appendix	27
B.1. First Chapter	27
B.2. Second Chapter	27

Chapter 1. User Guide

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"

1.1.4. The Applet Store

By default, a newly signed up user would see the "Applet Store" (applet) upon successful login. From the **Applet Store** , a user would be able to install applets in the public or private applet-catalogs.

It is very important for users to take note that they are able to see "RANK" tag and "TENANT" tag before installing the applet.

1.2. Project Implementors

Understanding of the Akaun Platform is a prerequisite for successful implementation of any projects using the BigLedger platform.

In this section, we are going to

1.2.1. Subscription setup in Akaun Platform

This step can only be performed by the

1.2.2. Tenant setup in Akaun Platform

As of the time of writing, only the Akaun Platform system administrators (Wavelet and BigLedger employees) are able power up the tenant for you. Each of the tenant

1.2.3. Applet Catalogs

1.2.4. Configuration of Core Modules

1.3. Tenant Owners

Platform Guide for Tenant Owners

1.3.1. Section 1

1.3.2. Section 2

1.4. Applet Developers

Guid for applet Developers

1.4.1. Signing up your bigledger account

1.5. Reseller Guide

Guide for resellers

1.5.1. Section 1

1.5.2. Section 2

1.6. System Integrators

Guide for system integrators

1.6.1. Section 1

some sections here

1.6.2. Section 2

Chapter 2. Guide by Industry

This guide by industry provide :

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

2.1. Consumer Electronics Industry

2.1.1. Problems

2.1.2. Solutions

2.1.3. Features

2.2. Pharmacy Industry

2.2.1. Problems

2.2.2. Solutions

2.2.3. Features

2.3. Car Workshops

2.3.1. Problems

2.3.2. Solutions

2.3.3. Features

2.4. Retail Chainstores

2.4.1. Problems

2.4.2. Solutions

2.4.3. Features

2.5. Logistics Industry

2.5.1. Problems

2.5.2. Solutions

2.5.3. Features

2.6. Manufacturing Industry

2.6.1. Problems

2.6.2. Solutions

2.6.3. Features

2.7. Pet Industry

2.7.1. Problems

2.7.2. Solutions

2.7.3. Features

Chapter 3. 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.

3.1. Core Module

3.1.1. Overview

3.1.2. Benefits

3.1.3. Related Applets

3.1.4. Pricing

3.2. Accounting Module

3.2.1. Overview

3.2.2. Features

3.3. Point of Sales Module

3.4. Manufacturing Module

3.5. Membership Module

3.6. Multilevel Marketing Module

3.7. Unified Contact Center Module

3.8. Inventory Module

3.9. EcomSync Module

3.10. Delivery Module

3.11. Cross Platform Commerce

3.12. Procure to Pay Module

3.13. Order To Cash Module

Chapter 4. Applet Directory

4.1. Applet Listing

Chapter 5. 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

5.1. 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

5.2. 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

5.2.1. Integrations

spikes in loads

5.2.2. Infrastructure and Toolchain

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

5.2.3. Network Effect

5.2.4. Cloud Cost Savings

5.2.5. Artificial Intelligence

5.2.6. Human Resource

5.3. Journey

5.4. Challenges and Lessons

5.5. 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

Chapter 6. Technical Guide

This technical guide is used by both internal and external employees of

Unresolved directive in chapters/Technical_Guide/README.adoc -
include::Akaun_Applet_Development_Guid/README.adoc[]

6.1. Akaun Shell Development Guide

6.2. Cloud Infrastructure

6.3. BigLedger Database Guide

6.3.1. Overview

6.3.2. Database Schema

6.3.3. Liquibase

6.3.4. The Root Database Schema

6.3.5. The Tenant Database Schema

Unresolved directive in chapters/Technical_Guide/README.adoc -
include::BigLedger_Java_API_Guide[]

Unresolved directive in chapters/Technical_Guide/README.adoc -
include::Domain_Driven_Design_Guide[]

Chapter 7. Documentation Authoring

7.1. 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

7.2. Setup and Configurations

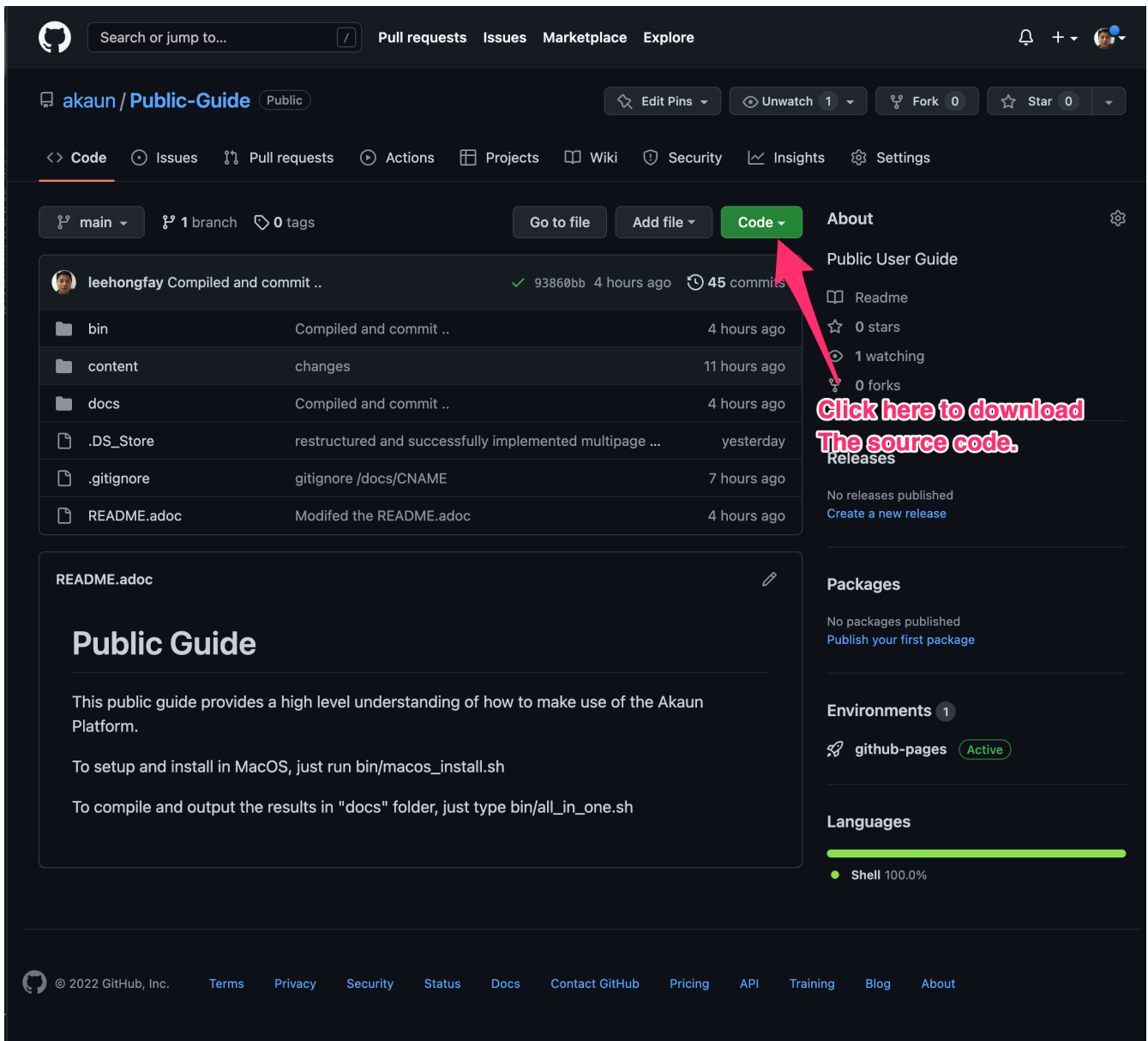
7.2.1. Github Repository

- 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.

7.2.2. 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>

7.2.3. 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

7.3. 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.

7.4. 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 !!

Glossary

Akaun Shell

The user interface that loads into a web browser, and showing users a list of applets.

Applet

An application that runs on the Akaun Shell.

Applet Store

The applet where users can discover other applets and install these other applets.

Environment

There are 3 environments, namely production, staging and development. Each of these environments lives in a different AWS account.

Platform Administrators

This refers to the BigLedger MIS Department.

System Administrator

This refers to the BigLedger MIS Department.

Root Database

This refers to the Platform main and global database.

Tenant

Every tenant is isolated from each other , it has its own independent database and other resources like files storage. Within a tenant, it is possible to create multiple companies, and they share the same database.

Tenant Admin

This refers to the OWNER or ADMIN of the Tenant.

Appendix A: First Appendix

A.1. First Subsection

A.2. Second Subsection

Appendix B: Second Appendix

B.1. First Chapter

B.2. Second Chapter

sdfsfa