Overview

Casibase is an open-source Domain Knowledge Database & IM & Forum Software powered by ChatGPT.

You need to enable JavaScript to run this app.

Casibase features

1. Front-end and back-end separate architecture, developed by Golang, Casibase supports high concurrency, provides web-based managing UI and supports multiple languages(Chinese, English).

2. Casibase supports third-party applications login, such as GitHub, Google, QQ, WeChat, etc., and supports the extension of third-party login with plugins.

3. Based on embedding and prompt engineering for knowledge management, Casibase supports customized embedding methods and language models.

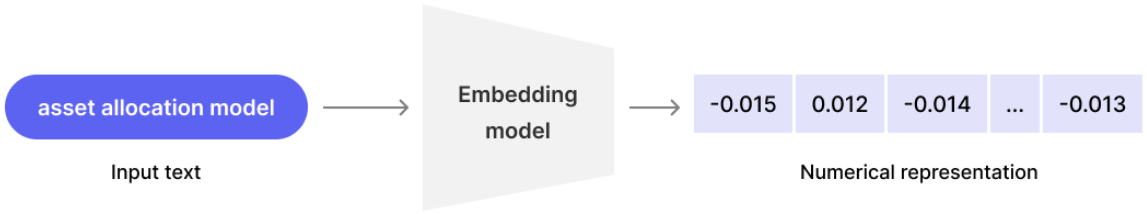
4. Casibase supports integration with existing systems by db sync, so users can transition to Casibase smoothly.

5. Casibase supports mainstream databases: MySQL, PostgreSQL, SQL Server, etc., and supports the extension of new databases with plugins.

How it works

Step 0 (Pre-knowledge)

Casibase's knowledge retrieval process is based on Embedding and Prompt engineering, so it is highly recommended that you take a brief look at how Embedding works. An introduction to Embedding.



Step 1 (Importing Knowledge)

To get started with Casibase, users need to follow these steps to import knowledge and create a domain-specific knowledge database:

1. Configure Storage: In the Casibase dashboard, users should first configure the storage settings. This involves specifying the storage system to be used for storing knowledge-related files, such as documents, images, or any other relevant data. Users can choose from a variety of storage options based on their preferences and requirements.

2. Upload Files to Storage: Once the storage is set up, users can proceed to upload files containing domain-specific knowledge into the configured storage system. These files can be in various formats, such as text documents, images, or structured data files like CSV or JSON.

3. Select Embedding Method for Knowledge Generation: After the files are uploaded, users have the option to choose the embedding method for generating knowledge and corresponding vectors. Embeddings are numerical representations of textual or visual content, which facilitate efficient similarity search and data analysis.

TIP

How knowledge is embedded?

- For textual data: Users can choose from various embedding methods, such as Word2Vec, GloVe, or BERT, to convert the textual knowledge into meaningful vectors.
- For visual data: If the uploaded files contain images or visual content, users can select image embedding techniques like CNN-based feature extraction to create representative vectors.
- More methods coming soon...

By following these steps, users can populate their domain knowledge database with relevant information and corresponding embeddings, which will be used for effective searching, clustering, and retrieval of knowledge within Casibase. The embedding process allows the system to understand the context and relationships between different pieces of knowledge, enabling more efficient and insightful knowledge management and exploration.

Step 2 (Retrieving Knowledge)

After importing your `domain knowledge`, Casibase transforms it into `vectors` and stores these vectors in a `vector database`. This vector representation enables powerful functions like `similarity search` and `efficient retrieval of related information`. You can quickly find relevant data based on context or content, enabling advanced querying and uncovering valuable insights within your domain knowledge.

Step 3 (Building the Prompt)

Casibase performs a similarity search on the stored knowledge vectors to find the closest match to the user's query. Using the search results, it creates a `prompt template` to frame a specific question for the `language model`. This ensures accurate and contextually relevant responses, delivering comprehensive answers based on the domain knowledge in Casibase.

Step 4 (Achieving the Goal)

At this stage, using Casibase, you have successfully acquired the knowledge you sought. Through the innovative combination of domain knowledge transformed into vectors and powerful language models like ChatGPT, Casibase provides you with accurate and relevant responses to your inquiries. This enables you to efficiently access and utilize the domain-specific information stored in Casibase, meeting your knowledge requirements with ease.

Step 5 (Optional Fine-tuning)

If you find that the results are not entirely satisfactory, you can try to get better results by doing the following:

- Tweaking Language Model Parameters
- Asking multiple questions
- Optimizing the original files

By utilizing these fine-tuning options, you can improve the efficiency of your knowledge management in Casibase, ensure that the system is better aligned with your goals, and provide more accurate and insightful information.

HINTS

Other ways to optimize results (may require source code changes):

- Updating `Embedding` Results: Refine the knowledge representation by adjusting the embedding results of your domain knowledge.
- Modifying `Prompt` Templates: By customizing the prompts, you can elicit more precise responses from the language model.
- Exploring Different `Language Models`: Experiment with different models to find the one that best suits your requirements for generating responses.

Online demo

Read-only site (any modification operation will fail)

- Chat bot: https://ai.casibase.com
- Admin UI: https://ai-admin.casibase.com

Writable site (original data will be restored for every 5 minutes)

- Chat bot: https://demo.casibase.com
- Admin UI: https://demo-admin.casibase.com

Global admin login:

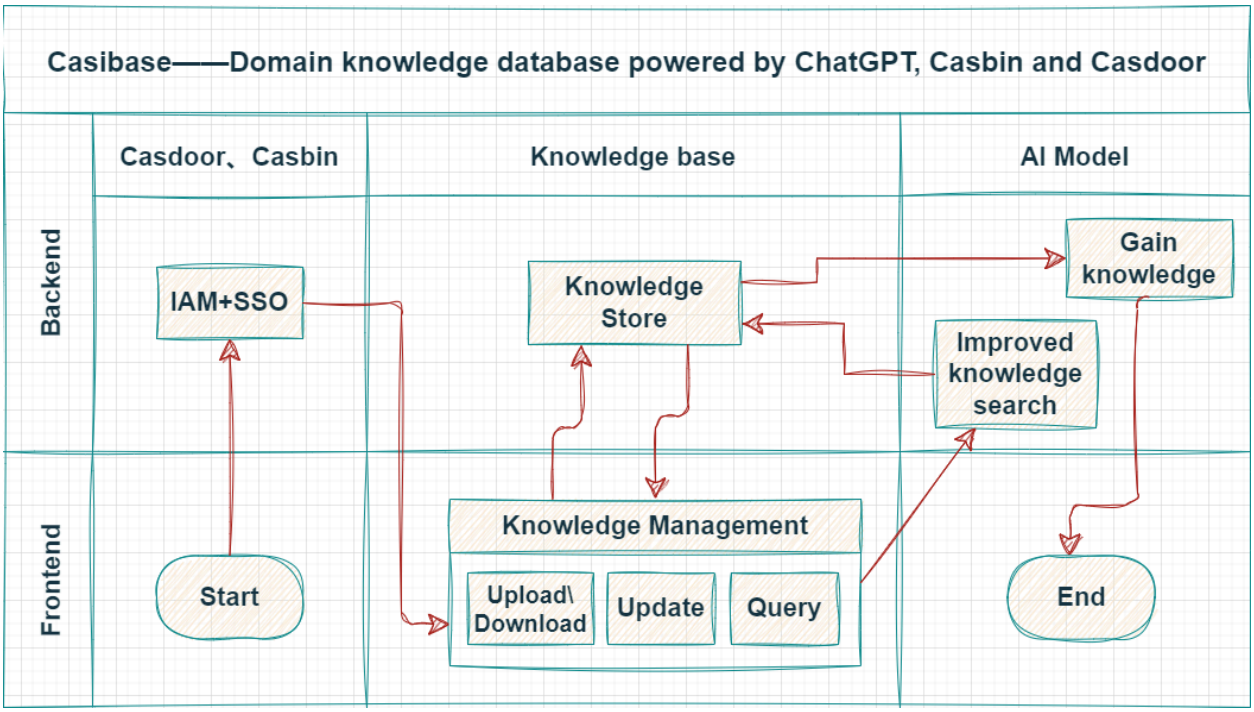
- Username: `admin`

- Password: `123`

Architecture

Casibase contains 2 parts:

Name	Description	Language	Source code
Frontend	User interface for the casibase application	JavaScript + React	https://github.com/casibase/casibase/tree/master/web
Backend	Server-side logic and API for casibase	Golang + Beego + MySQL	https://github.com/casibase/casibase



Supported Models

Language Model

Model	Sub Type	Link
OpenAI	gpt-4-32k-0613`gpt-4-32k-0314`gpt-4-32k`gpt-4-0613`gpt-4-0314`gpt-4`gpt-3.5-turbo-0613`gpt-3.5-turbo-0301`gpt-3.5-turbo-16k`gpt-3.5-turbo-16k-0613`gpt-3.5-turbo`text-davinci-003`text-davinci-002`text-curie-001`text-babbage-001`text-ada-001`text-davinci-001`davinci-instruct-beta`davinci`curie-instruct-beta`curie`ada`babbage	OpenAI
Hugging Face	meta-llama/Llama-2-7b, tiiuae/falcon-180B, bigscience/bloom, gpt2, baichuan-inc/Baichuan2-13B-Chat, THUDM/chatglm2-6b	Hugging Face
Claude	claude-2, claude-v1, claude-v1-100k, claude-instant-v1, claude-instant-v1-100k, claude-v1.3, claude-v1.3-100k, claude-v1.2, claude-v1.0, claude-instant-v1.1, claude-instant-v1.1-100k, claude-instant-v1.0	Claude
OpenRouter	google/palm-2-codechat-bison, google/palm-2-chat-bison, openai/gpt-3.5-turbo, openai/gpt-3.5-turbo-16k, openai/gpt-4,	OpenRouter

Model	Sub Type	Link
	openai/gpt-4-32k, anthropic/claude-2, anthropic/claude-instant-v1, meta-llama/llama-2-13b-chat, meta-llama/llama-2-70b-chat, palm-2-codechat-bison, palm-2-chat-bison, gpt-3.5-turbo, gpt-3.5-turbo-16k, gpt-4, gpt-4-32k, claude-2, claude-instant-v1, llama-2-13b-chat, llama-2-70b-chat	
Ernie	ERNIE-Bot, ERNIE-Bot-turbo, BLOOMZ-7B, Llama-2	Ernie
iFlytek	spark-v1.5, spark-v2.0	iFlytek
ChatGLM	chatglm2-6b	ChatGLM
MiniMax	abab5-chat	MiniMax
Local	custom-model	Local Computer

Embedding Model

Model	Sub Type	Link
OpenAI	AdaSimilarity, BabbageSimilarity, CurieSimilarity, DavinciSimilarity, AdaSearchDocument, AdaSearchQuery, BabbageSearchDocument, BabbageSearchQuery, CurieSearchDocument, CurieSearchQuery, DavinciSearchDocument,	OpenAI

Model	Sub Type	Link
	DavinciSearchQuery, AdaCodeSearchCode, AdaCodeSearchText, BabbageCodeSearchCode, BabbageCodeSearchText, AdaEmbeddingV2	
Hugging Face	sentence-transformers/all-MiniLM-L6-v2	Hugging Face
Cohere	embed-english-v2.0, embed-english-light-v2.0, embed-multilingual-v2.0	Cohere
Ernie	default	Ernie
Local	custom-embedding	Local Computer

Core Concepts

As Casibase's user, you should get familiar with at least 4 core concepts: `Provider`, `Storage`, `Chat` and `Vector`.

Provider

Providers are the backbone of Casibase, offering essential services and integration with external systems. The Provider class definition is shown as follows:

```
type Provider struct {
    Owner       string `xorm:"varchar(100) notnull pk"
json:"owner"`
    Name        string `xorm:"varchar(100) notnull pk" json:"name"`
    CreatedTime string `xorm:"varchar(100)" json:"createdTime"`

    DisplayName  string `xorm:"varchar(100)" json:"displayName"`
    Category     string `xorm:"varchar(100)" json:"category"`
    Type         string `xorm:"varchar(100)" json:"type"`
    ClientId     string `xorm:"varchar(100)" json:"clientId"`
    ClientSecret string `xorm:"varchar(2000)" json:"clientSecret"`
    ProviderUrl  string `xorm:"varchar(200)" json:"providerUrl"`
}
```

TIP

There are two primary types of providers in Casibase:

- Storage Provider:The Storage Provider facilitates the storage and

retrieval of data within Casibase. It supports various storage options, including:

  - AWS
  - Azure
  - Local File System

- **AI Provider:**The AI Provider is responsible for handling AI-related tasks and services in Casibase. It supports multiple AI models and technologies, including:

  - OpenAI
  - ChatGLM
  - InternLM

Vectors

Vectors in Casibase represent numerical representations of different types of data. These vectors enable efficient processing and analysis of information. Some of the vector types available are:

- Text Vector
- Image Vector
- ... (other vector types)

The Vector class definition is shown as follows:

```
type Vector struct {
    Owner       string    `xorm:"varchar(100) notnull pk"
```

Chats

Chats are at the core of interactive communication between users and the AI models in Casibase. They consist of three essential components:

- Question: The user's input or query, seeking information or assistance.
- Query Prompt: A formatted version of the user's question, prepared for processing by the AI models.
- Answer: The AI-generated response to the user's question, providing relevant information or solutions.

The Chat class definition is shown as follows:

```
type Chat struct {
    Owner        string   `xorm:"varchar(100) notnull pk"
json:"owner"`
    Name         string   `xorm:"varchar(100) notnull pk"
json:"name"`
    CreatedTime  string   `xorm:"varchar(100)" json:"createdTime"`
    UpdatedTime  string   `xorm:"varchar(100)" json:"updatedTime"`

    DisplayName  string   `xorm:"varchar(100)" json:"displayName"`
    Category     string   `xorm:"varchar(100)" json:"category"`
    Type         string   `xorm:"varchar(100)" json:"type"`
    User1        string   `xorm:"varchar(100)" json:"user1"`
    User2        string   `xorm:"varchar(100)" json:"user2"`
    Users        []string `xorm:"varchar(100)" json:"users"`
    MessageCount int      `json:"messageCount"`
}
```

Embedding

Embedding is the process of transforming various types of data, such as text and images, into dense vector representations. This step is crucial for facilitating efficient data processing and analysis within Casibase.

TIP

- By embedding, the questions in chat and the knowledge files in storage will be turned into vectors and used in the next step of knowledge search.
- Casibase's default embedding method is provided by OpenAI at a rate of up to three calls per minute. We propose to minimize the knowledge file coupling as much as possible to facilitate embedding and further processing.

Server Installation

Requirements

OS

All major operating systems including Windows, Linux and macOS are supported.

Environment

- Go 1.20+
- Node.js LTS (18)
- Yarn 1.x

> INFO
>
> The use of Casibase is divided into two steps:
>
> - step1: Deploy and run Casdoor
> - step2: Deploy and run Casibase (this docs)
>
> We strongly suggest you use Yarn 1.x to run & build Casdoor&Casibase frontend, using NPM might cause UI styling issues, see more details at: casdoor#294

> CAUTION
>
> For **Chinese** users, in order to download the Go dependency packages successfully, you need to use a Go proxy by Configuring the GOPROXY environment variable. We strongly recommend: https://goproxy.cn/

Database

Casibase uses XORM to talk to the database. Based on Xorm Drivers Support, Casibase currently provides support for following databases:

- `MySQL`
- `MariaDB`
- `PostgreSQL`
- `CockroachDB`
- `SQL Server`
- `Oracle`
- `SQLite 3`
- `TiDB`

Download

The source code of Casibase is hosted at GitHub: https://github.com/casibase/casibase. Both the Go backend code and React frontend code are inside the single repository.

Name	Description	Language	Source code
Frontend	Web frontend UI for Casibase	JavaScript + React	https://github.com/casibase/casibase/tree/master/web
Backend	RESTful API backend for	Golang + Beego +	https://github.com/casibase/casibase

Name	Description	Language	Source code
	Casibase	XORM	

Casibase supports `Go Modules`. To download the code, you can just simply clone the code via git:

```
cd path/to/folder
git clone https://github.com/casibase/casibase
```

Configuration

Configure Casdoor

Please refer to Casdoor-SSO section to configure Casdoor.

Remember your `clientId`、`clientSecret`、`organization`、`application` and so on in Casdoor configuration, we will use them later.

Configure Database

Casibase supports mysql, mssql, sqlite3, postgres. Casibase uses mysql by default.

MySQL

Casibase will store its users, nodes and topics information in a MySQL database named: `casibase`. If the database does not exist, it needs to be created manually. The DB connection string can be specified at: https://github.com/casibase/

casibase/blob/master/conf/app.conf

```
driverName = mysql
dataSourceName = root:123456@tcp(localhost:3306)/
dbName = casibase
```

PostgreSQL

Since we must choose a database when opening Postgres with xorm, you should prepare a database manually before running Casibase.

Let's assume that you have already prepared a database called `casibase`, then you should specify `app.conf` like this:

```
driverName = postgres
dataSourceName = "user=postgres password=postgres host=localhost
port=5432 sslmode=disable dbname=casibase"
dbName =
```

> INFO
>
> For PostgreSQL, make sure `dataSourceName` has non-empty `dbName` and leave the standalone `dbName` field empty like the above example.

CockroachDB

You can also use cockroachdb with postgres driver. It has same configuration as postgreSQL.

```
driverName = postgres
dataSourceName = "user=postgres password=postgres host=localhost
port=5432 sslmode=disable dbname=casibase
```

> INFO
>
> For CockroachDB, don't forget to add `serial_normalization=virtual_sequence` to the `dataSourceName` like the above example. otherwise you will get error regarding existed database, whenever the service started or restarted. Notice, this must be added before the database created.

Sqlite3

You should specify `app.conf` like this:

```
driverName = sqlite
dataSourceName = "file:casibase.db?cache=shared"
dbName = casibase
```

Custom configuration

Casibase supports custom configuration, you can modify the configuration file `conf/app.conf` to change the configuration.

- Backend (conf/app.conf)

```
casdoorEndpoint = <Your Casdoor endpoint>
clientId = <Your Casdoor application's client ID>
clientSecret = <Your Casdoor application's client secret>
casdoorOrganization = <Your Casdoor organization name>
casdoorApplication = <Your Casdoor application name>
```

- Frontend (web/src/Conf.js)

```
serverUrl: "<Your Casdoor endpoint>"
clientId: "<Your Casdoor application's client ID>"
appName: "<Your Casdoor application name>"
organizationName: "<Your Casdoor organization name>"
```

Run

There are currently two methods to start, you can choose one according to your own situation.

> ⚠ CAUTION
>
> **Casibase** requires **Casdoor** to provide access control and some back-end services, so you must make sure **Casdoor** is running properly before running **Casibase**.
>
> How to install and run Casdoor:
>
> - Casdoor Installation

Development mode

Backend

Casibase's Go backend runs at port 14000 by default. You can start the Go backend with the following command:

```
go run main.go
```

After the server is successfully running, we can start the frontend part.

Frontend

Casibase's frontend is a very classic Create-React-App (CRA) project. It runs at port `13001` by default. Use the following commands to run the frontend:

```
cd web
yarn install
yarn start
```

Production mode

Backend

Build Casibase Go backend code into executable and start it.

For Linux:

```
go build
./casibase
```

For Windows:

```
go build
casibase.exe
```

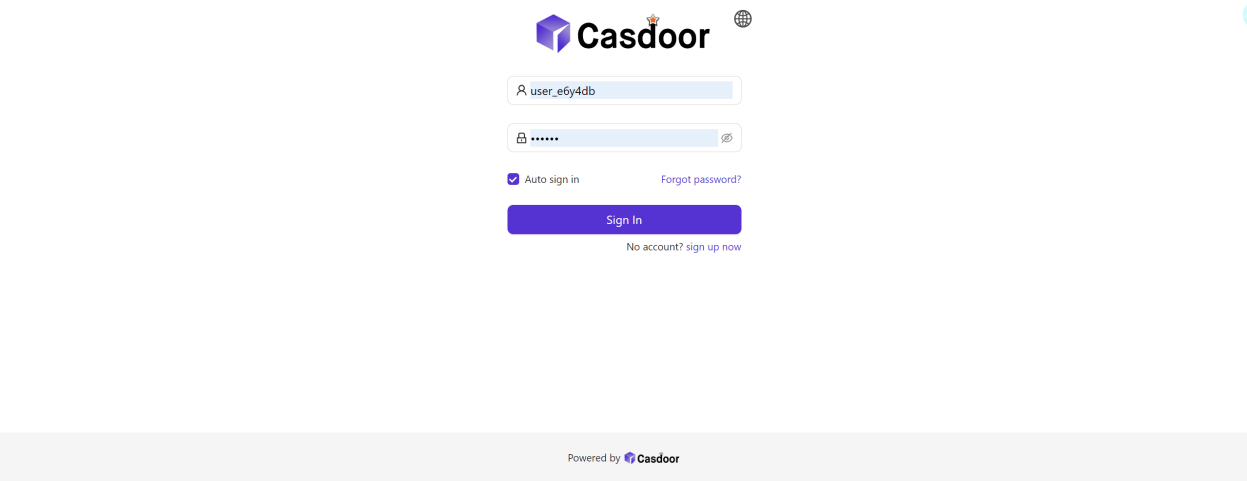
Frontend

Build Casibase frontend code into static resources (.html, .js, .css files):

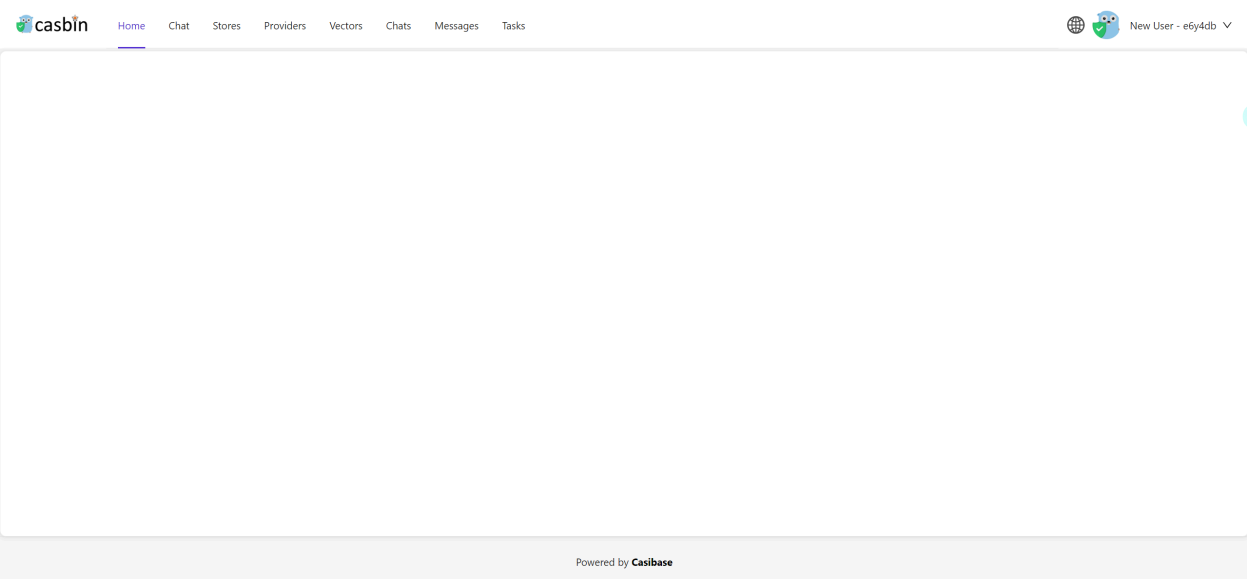
```
cd web
```

Preview

Visit: `http://localhost:13001` in your browser. Login into Casibase dashboard with the user account you have just registered in Casdoor:



Then you will go to the home page of Casibase:



TIP

To use another port, please edit `conf/app.conf` and modify `httpport`, then restart the Go backend.

(Optional) Try with Docker

Requirements

Hardware

If you want to build the Docker image yourself, please ensure that your machine has at least 2GB of memory. Casibase's frontend is an NPM project of React. Building the frontend requires at least 2GB of memory. Having less than 2GB of memory may result in a frontend build failure.

If you only need to run the pre-built image, please ensure that your machine has at least 100MB of memory.

OS

All operating systems (Linux, Windows, and macOS) are supported.

Docker

You can use **Docker (docker-engine version >= 17.05)** in Linux or **Docker Desktop** in Windows and macOS.

- Docker

Regardless of the operating system, users must ensure that they have **docker-engine version >= 17.05**. This is because we utilize the multi-stage build feature in the docker-compose.yml, which is supported in versions 17.05 and above. For more information, see https://docs.docker.com/develop/develop-images/multistage-build/.

If you are also using docker-compose, please ensure that you have **docker-compose version** >= 2.2. For Linux users, you also need to make sure that docker-compose is installed, as it is separate from docker-engine.

Get the image

We have provided two DockerHub images:

Name	Description	Suggestion
casibase-all-in-one	Both Casibase and a MySQL database are included in the image	This image already includes a toy database and is only for testing purposes
casibase	Only Casibase is included in the image	This image can be connected to your own database and used in production

1. casbin/casibase-all-in-one: This image includes the casibase binary, a MySQL database, and all the necessary configurations. It is designed for new users who want to try Casibase quickly. With this image, you can start Casibase immediately with just one or two commands, without any complex configuration. However, please note that we **do not recommend** using this image in a production environment.

Option-1: Use the toy database

Run the container with port `14000` exposed to the host. The image will be automatically pulled if it doesn't exist on the local host.

```
docker run -p 14000:14000 casbin/casibase-all-in-one
```

Visit http://localhost:14000 in your browser. Log into the Casibase dashboard with the default global admin account: `built-in/admin`

```
admin
123
```

Option-2: Try with docker-compose

Create a `conf/app.conf` directory in the same directory level as the `docker-compose.yml` file. Then, copy app.conf from Casibase. For more details about `app.conf`, you can see Via Ini file.

Create a separate database using docker-compose:

```
docker-compose up
```

That's it! 🛩️

Visit http://localhost:14000 in your browser. Log into the Casibase dashboard with the default global admin account: `built-in/admin`

```
admin
123
```

Note: If you dig deeper into the docker-compose.yml file, you may be puzzled by the environment variable we created called "RUNNING_IN_DOCKER". When the database 'db' is created via docker-compose, it is available on your PC's localhost but not the localhost of the Casibase container. To prevent you from running into

troubles caused by modifying app.conf, which can be quite difficult for a new user, we provided this environment variable and pre-assigned it in the docker-compose.yml. When this environment variable is set to true, localhost will be replaced with host.docker.internal so that Casibase can access the database.

Option-3: Try directly with the standard image

> TIP
>
> If it is not convenient to mount the configuration file to a container, using environment variables is also a possible solution.
>
> example
>
> ```
> docker run \
>   -e driverName=mysql \
>   -e dataSourceName='user:password@tcp(x.x.x.x:3306)/' \
>   -p 14000:14000 \
>   casbin/casibase:latest
> ```

Create `conf/app.conf`. You can copy it from conf/app.conf in Casibase. For more details about `app.conf`, you can see Via Ini file.

Then run

```
docker run  -p 14000:14000 -v /folder/of/app.conf:/conf casbin/casibase:latest
```

Anyway, just **mount the app.conf to /conf/app.conf** and start the container.

Visit http://localhost:14000 in your browser. Log into the Casibase dashboard with the default global admin account: `built-in/admin`

```
admin
123
```

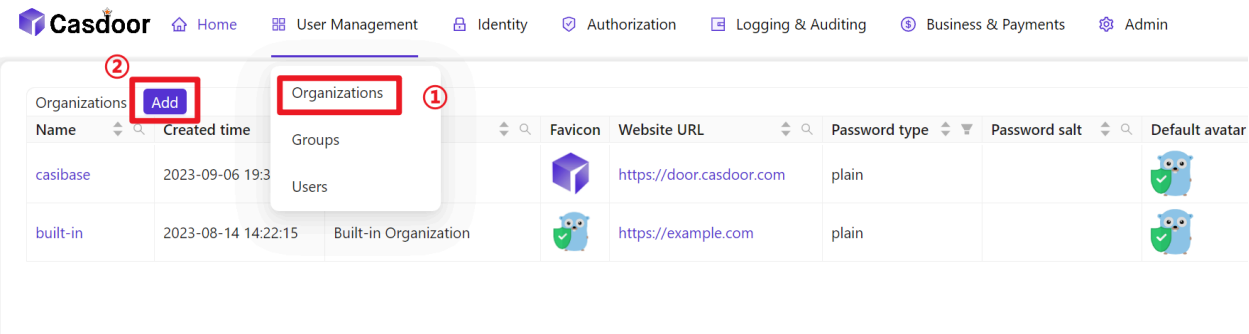
Casdoor-SSO

Casibase uses Casdoor as its identity and single-sign-on (SSO) provider. Make sure to deploy it in advance.

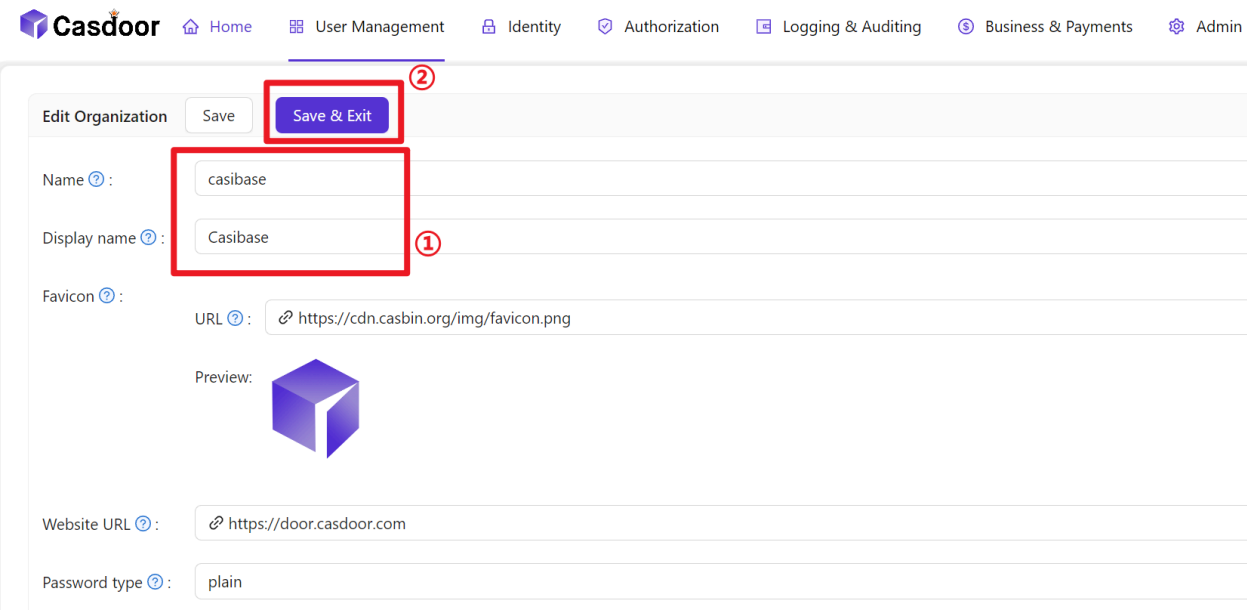
Please refer to Casdoor Server Installation to install and configure Casdoor.

Follow these steps to setup Casdoor for casibase:

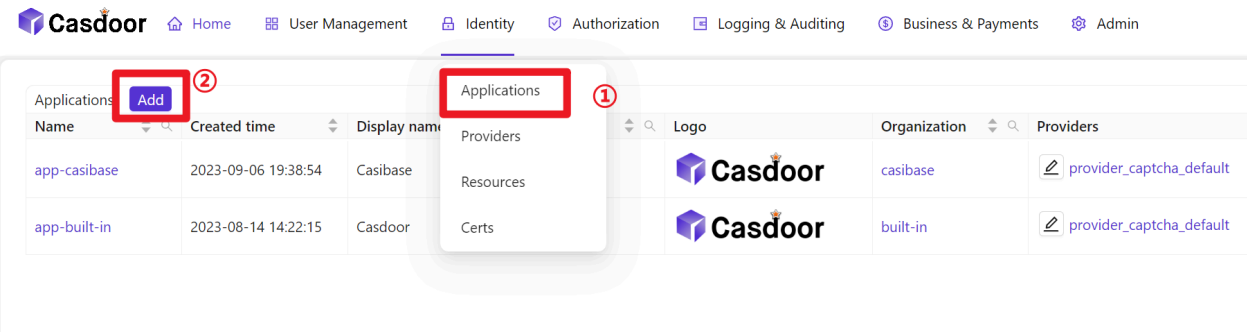
- Create an Organization



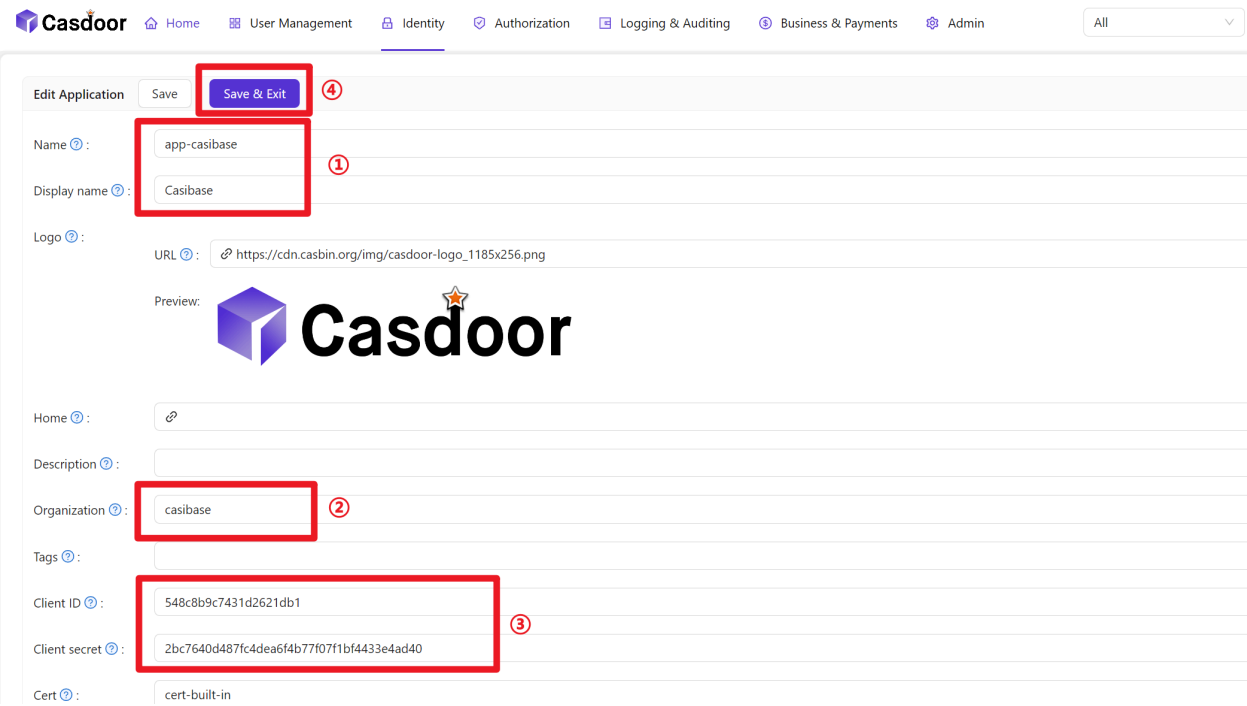
- Configure information about the Organization



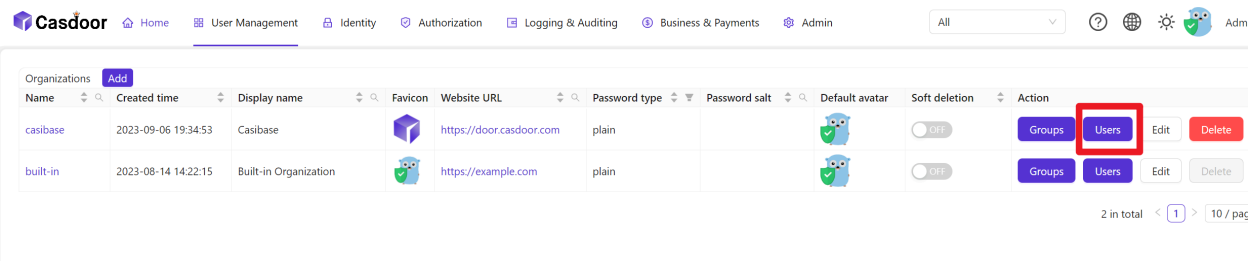
- Create a new Application

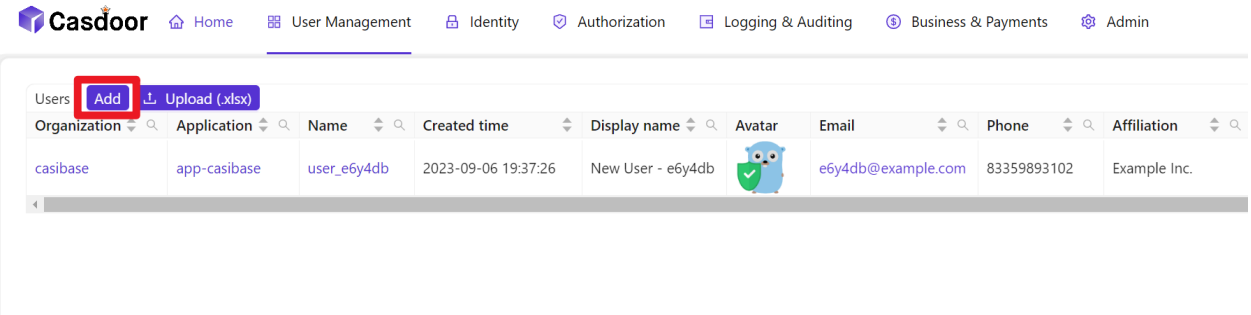


- Configuring Application Information (Remember Name, ClientID and ClientSecret)

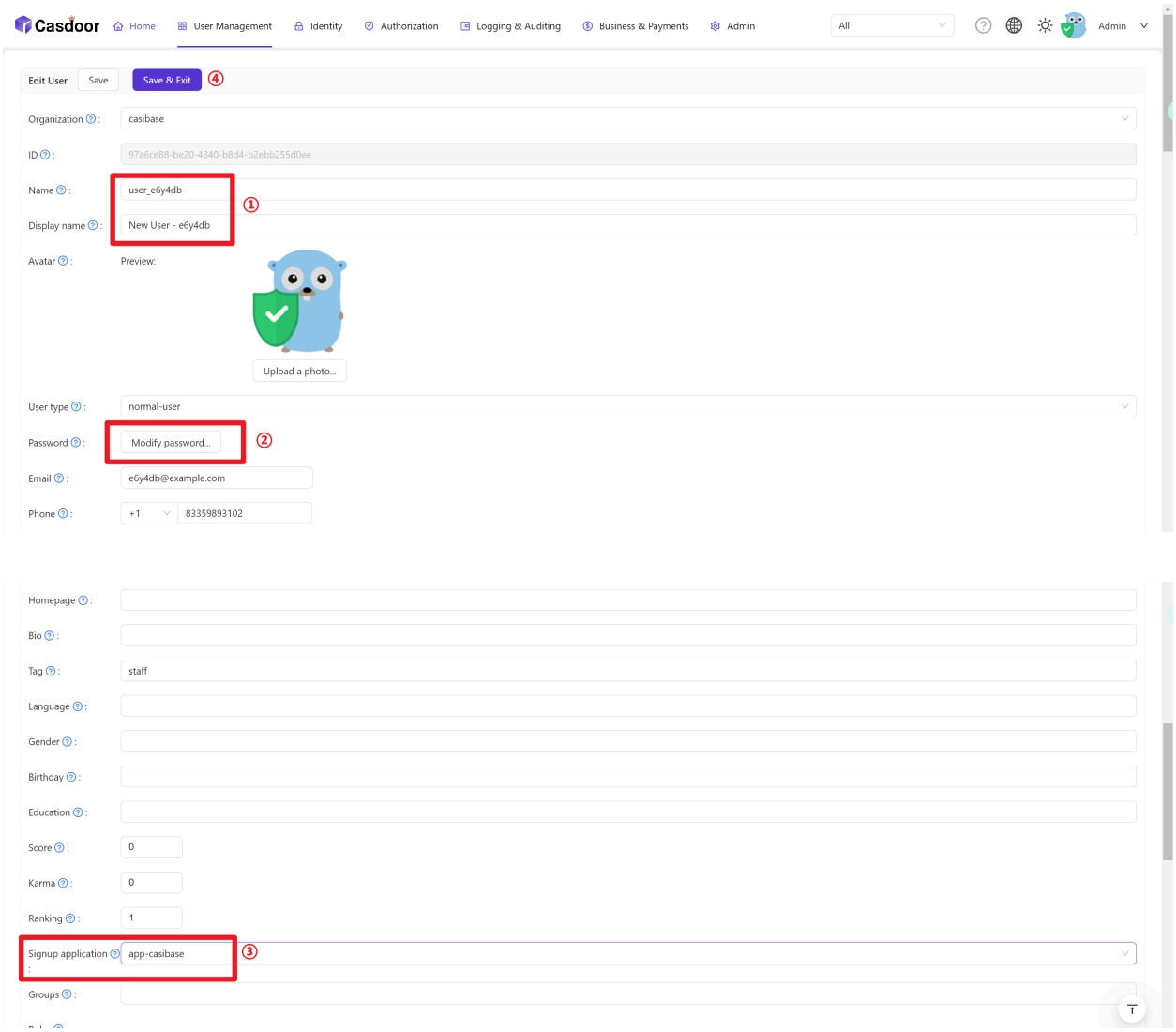


- Add a member to the newly created organization





- Configure member information (remember its Name as well as Password)


Casdoor
Home
User Management
Identity
Authorization
Logging & Auditing
Business & Payments
Admin
All
Admin
Edit User
Save
Save & Exit
④
Organization :
casibase
ID :
97a6ce88-be20-4840-b8d4-b2ebb255d0ee
Name :
user_e6y4db
①
Display name :
New User - e6y4db
Avatar :
Preview:
Upload a photo...
User type :
normal-user
Password :
Modify password...
②
Email :
e6y4db@example.com
Phone :
+1
83359893102
Homepage :
Bio :
Tag :
staff
Language :
Gender :
Birthday :
Education :
Score :
0
Karma :
0
Ranking :
1
Signup application :
app-casibase
③
Groups :

Walkthrough Guides

📄 Deploy Casdoor and Casibase

Discover how to deploy Casdoor and Casibase.

📄 Add a Storage Provider

Discover how to integrate a storage provider with Casibase.

📄 Add a Model Provider

Learn how to add a model provider to enhance Casibase functionality.

📄 Add an Embedding Provider

Explore how to integrate an embedding provider with Casibase.

📄 Add a Store

Learn how to add a store to your Casibase knowledge base system.

📄 Chat with AI

Implement AI chat functionality in your Casibase knowledge base system.

Deploy Casdoor and Casibase

Introduction

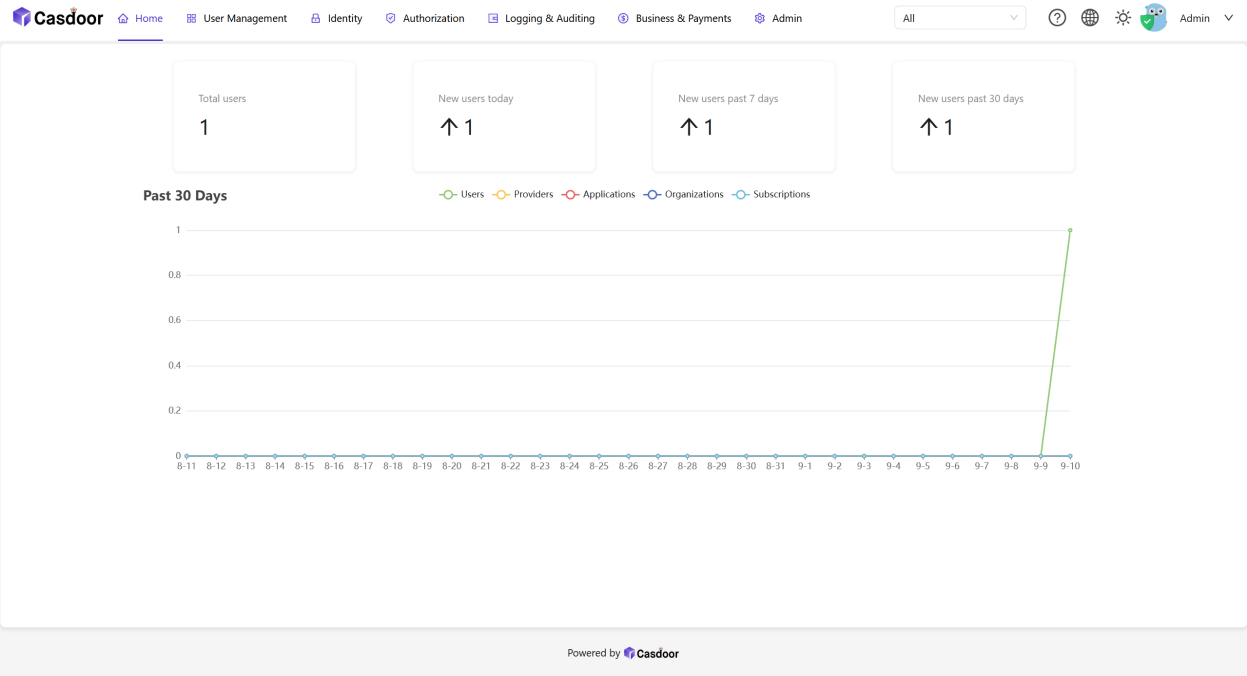
This document is a step-by-step tutorial designed for beginners. It will guide you through the process of deploying Casdoor and Casibase, our powerful knowledge base system.

> TIP
>
> What is Casdoor?
>
> Casdoor is a powerful authentication system that provides a secure and reliable login experience. It's a prerequisite for Casibase, so be sure to deploy it first.
>
> Refer to the Casdoor website for more information.

Step 1: Deploy Casdoor

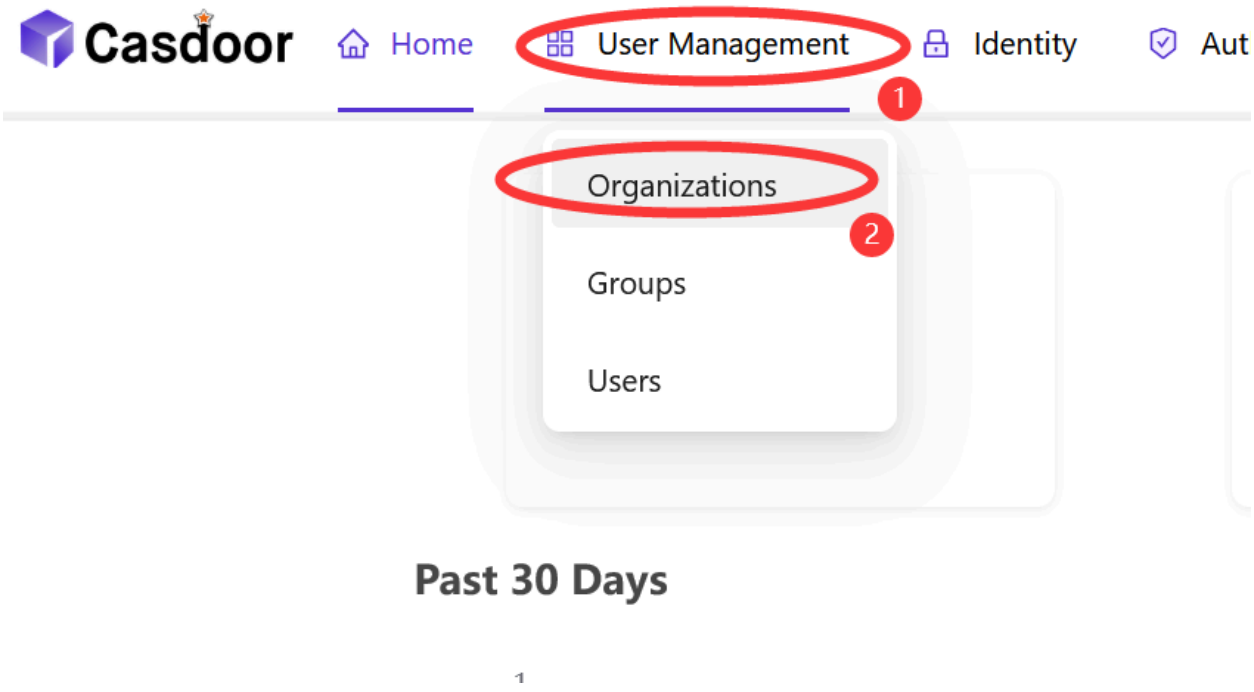
In Casdoor Deployment Guide, you can find the detailed steps to deploy Casdoor.

Once you've deployed Casdoor, you'll look like this:



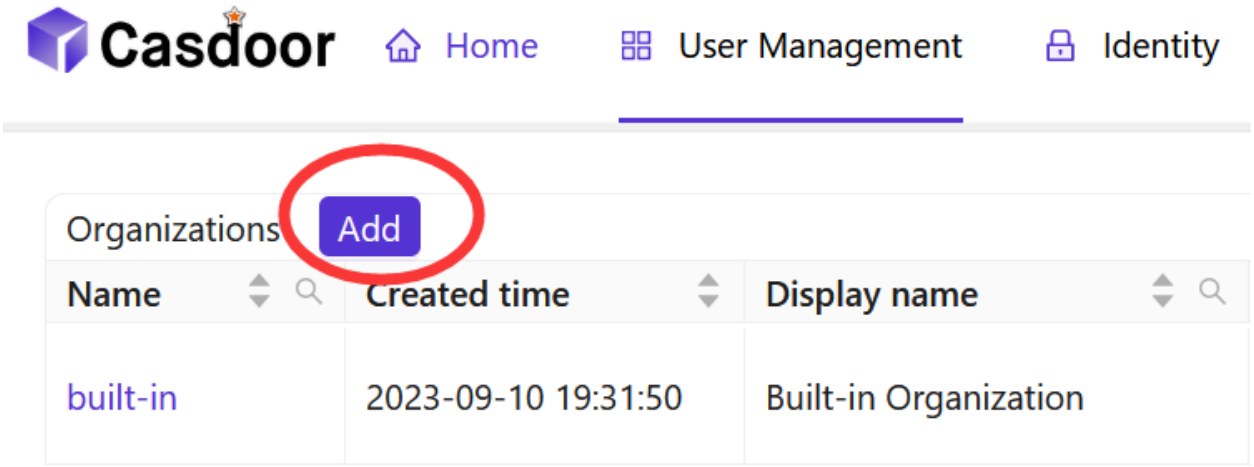
Step 2: Create an organization in Casdoor

In Casdoor, you can create an organization to manage your users and applications. You can create an organization by clicking the `User Management - Organizations` button on the home page.



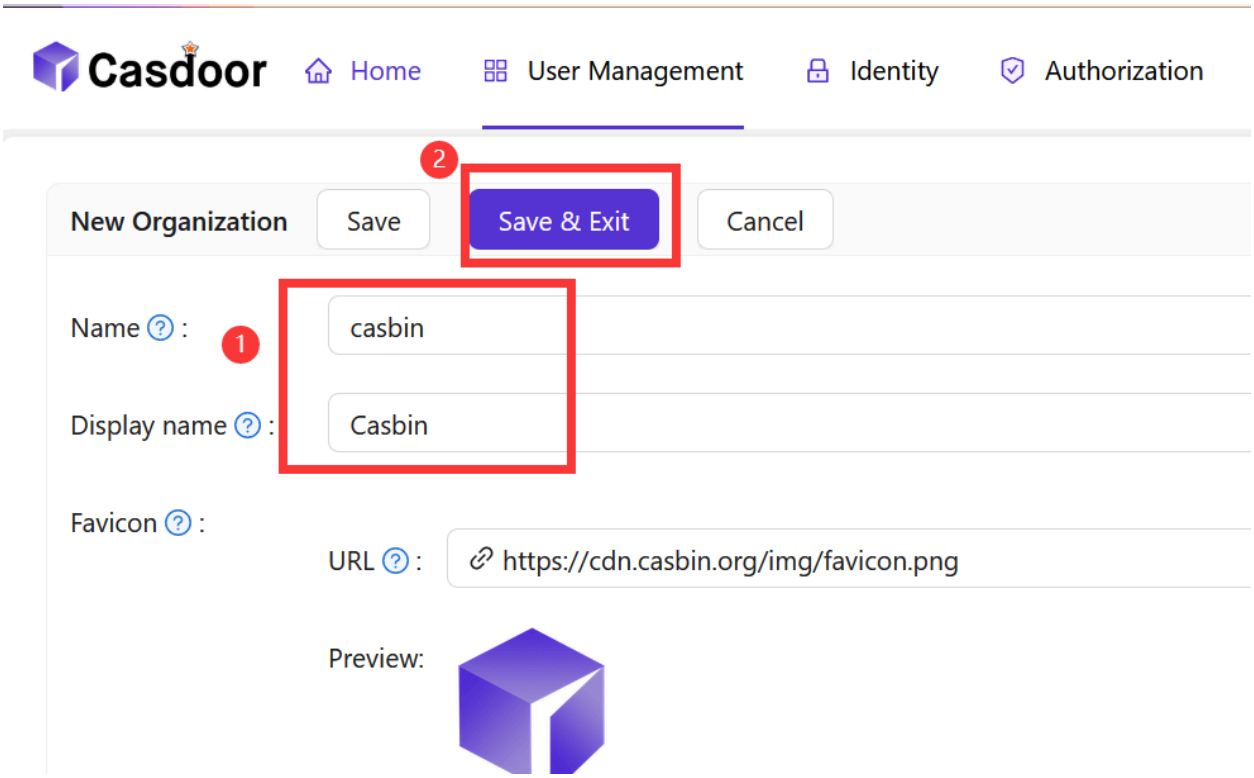
Step 2.1: Add an organization

Click the `Add` button to add an organization.



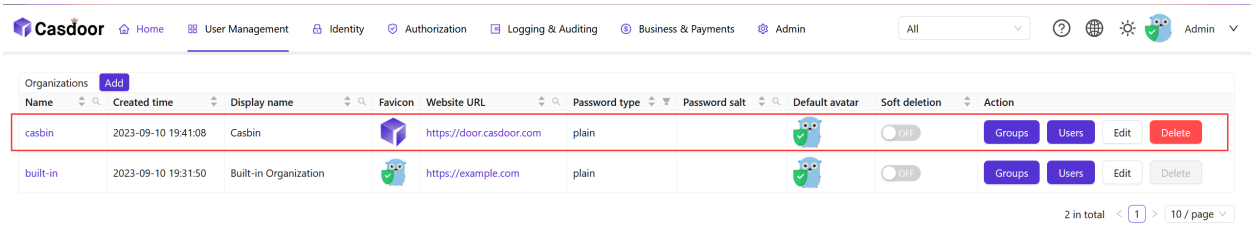
Step 2.2: Fill in the organization information

Fill in the organization information and click the `Save & Exit` button.



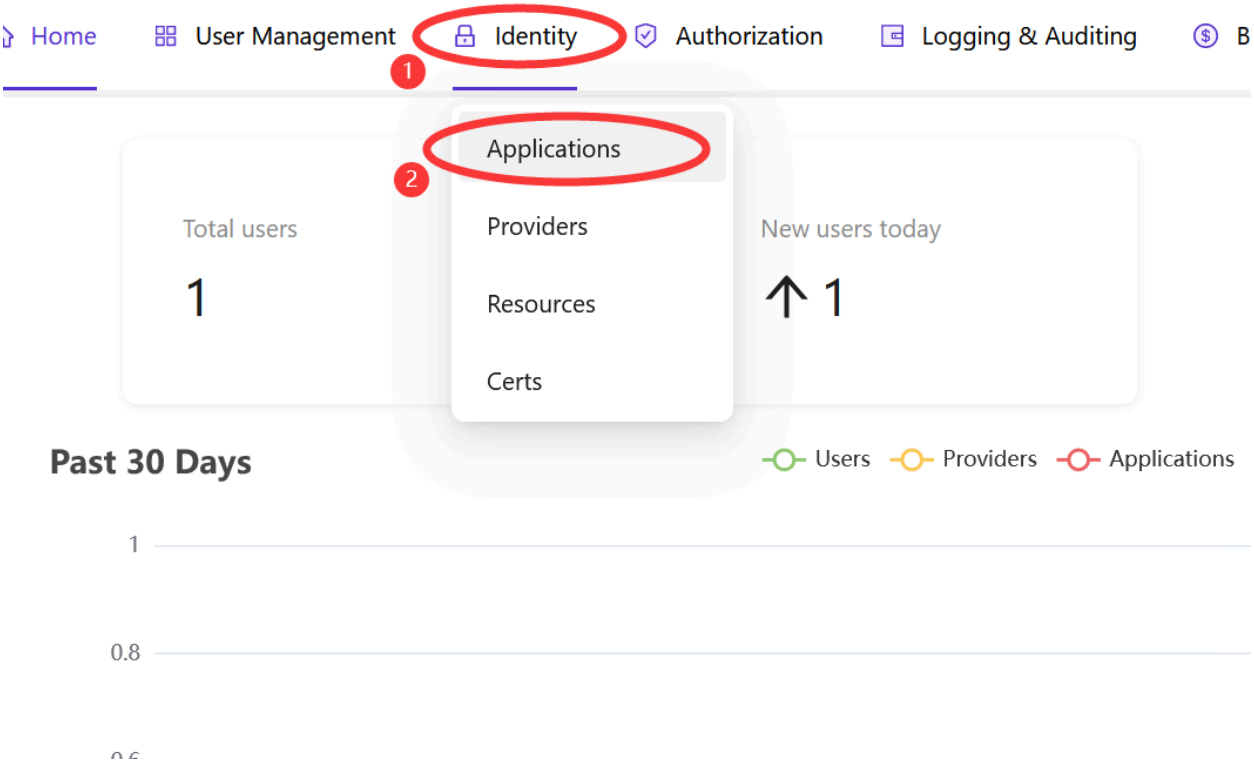
Step 2.3: View the organization

After adding the organization, you can view the organization information.



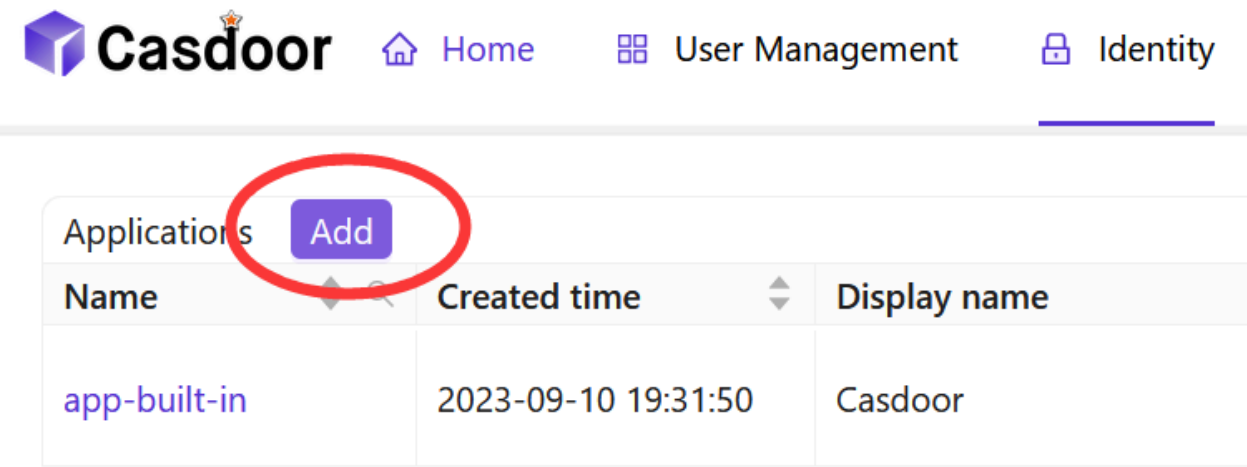
Step 3: Create an application in Casdoor

In Casdoor, you can create an application to manage your users and organizations. You can create an application by clicking the `Identity - Applications` button on the home page.



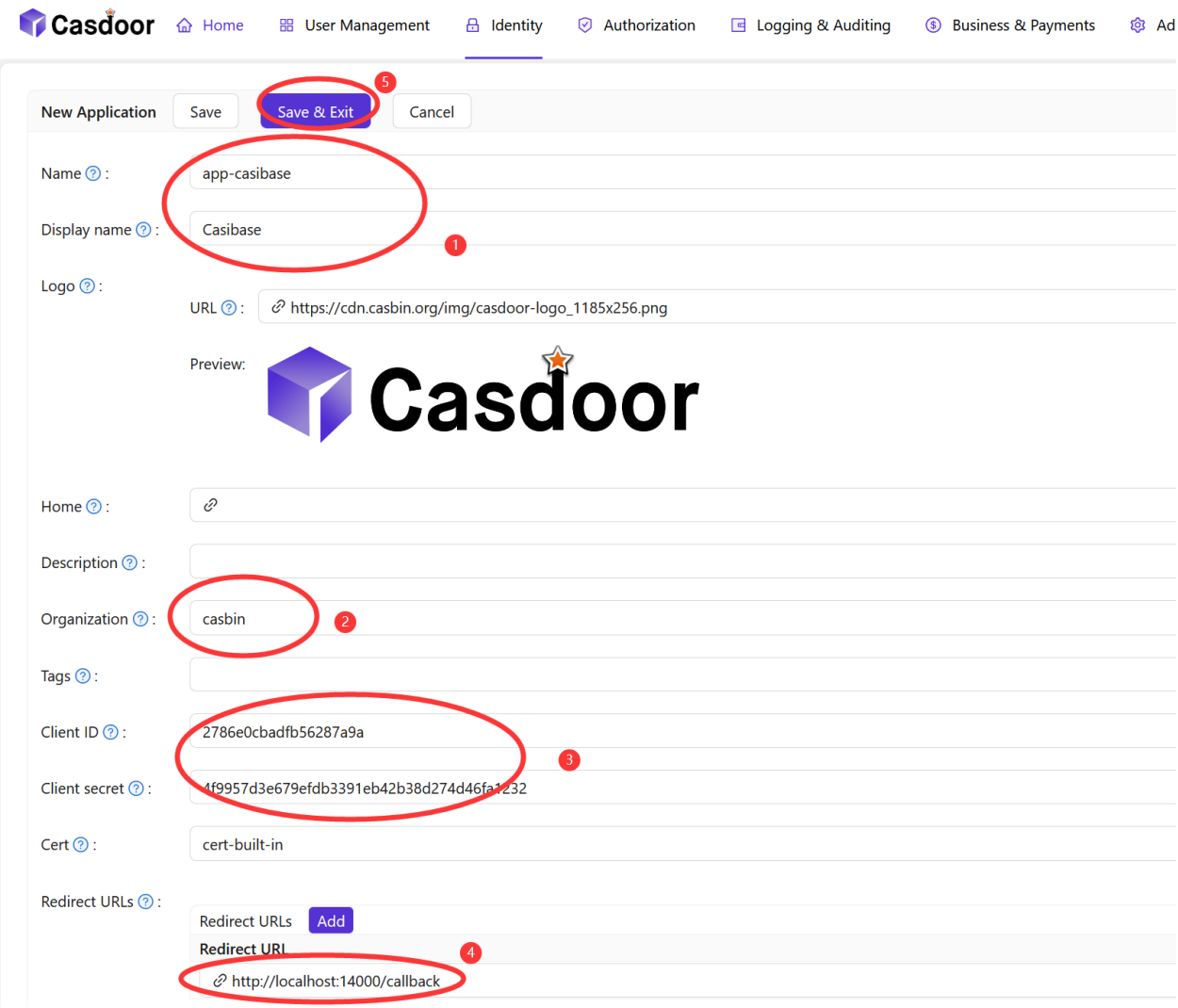
Step 3.1: Add an application

Click the `Add` button to add an application.



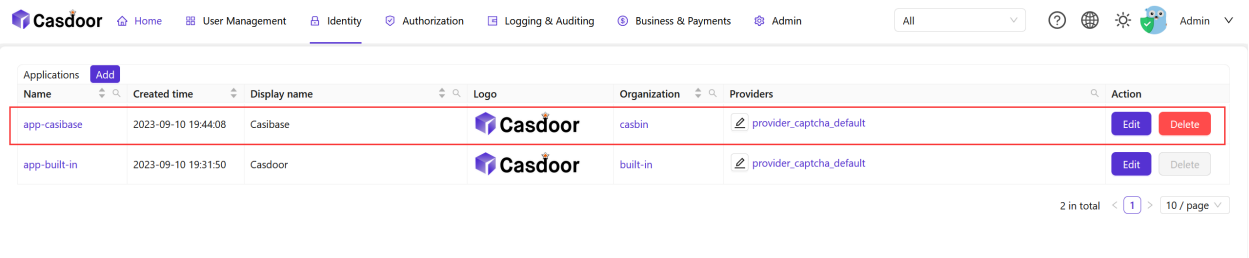
Step 3.2: Fill in the application information

Fill in the application information and click the `Save & Exit` button.



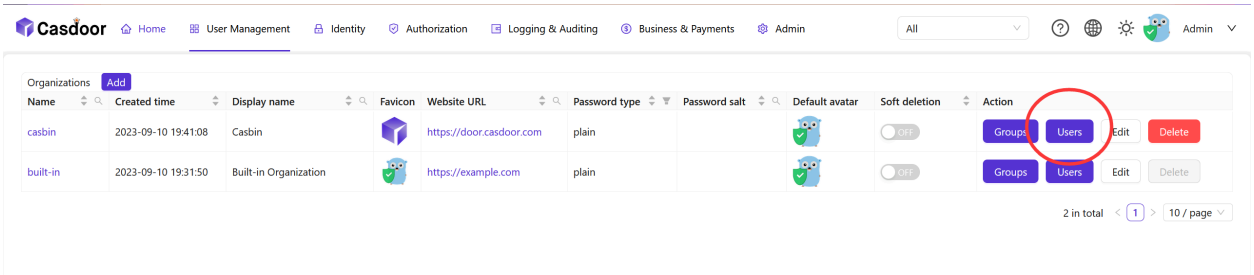
Step 3.3: View the application

After adding the application, you can view the application information.



Step 4: Create a user in Casdoor for Casibase

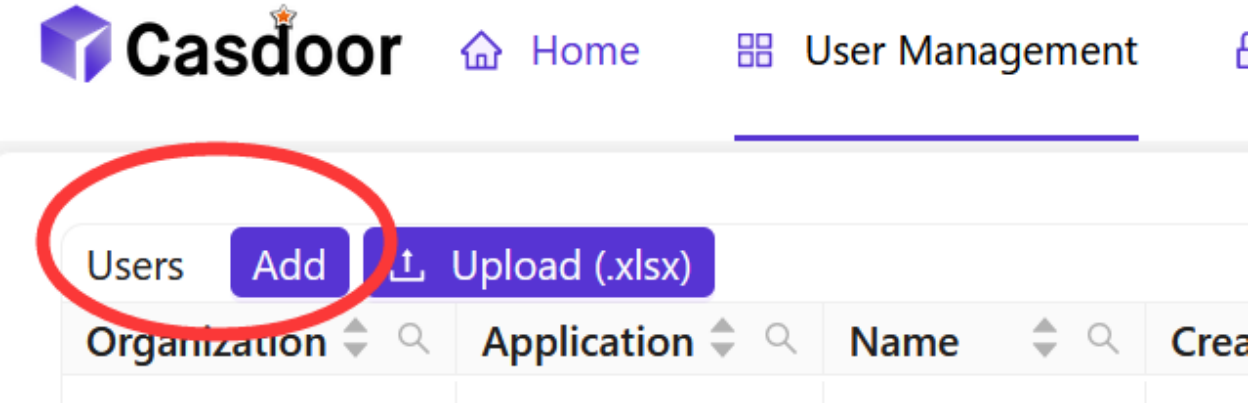
In Casdoor, you can create a user to login Casibase. You can create a user by clicking the `User Management - Organizations - Users` button from the home page.



> **TIP**
>
> A user is a member of an organization who can login to applications in the organization.
>
> Refer to the Casdoor website for more information.

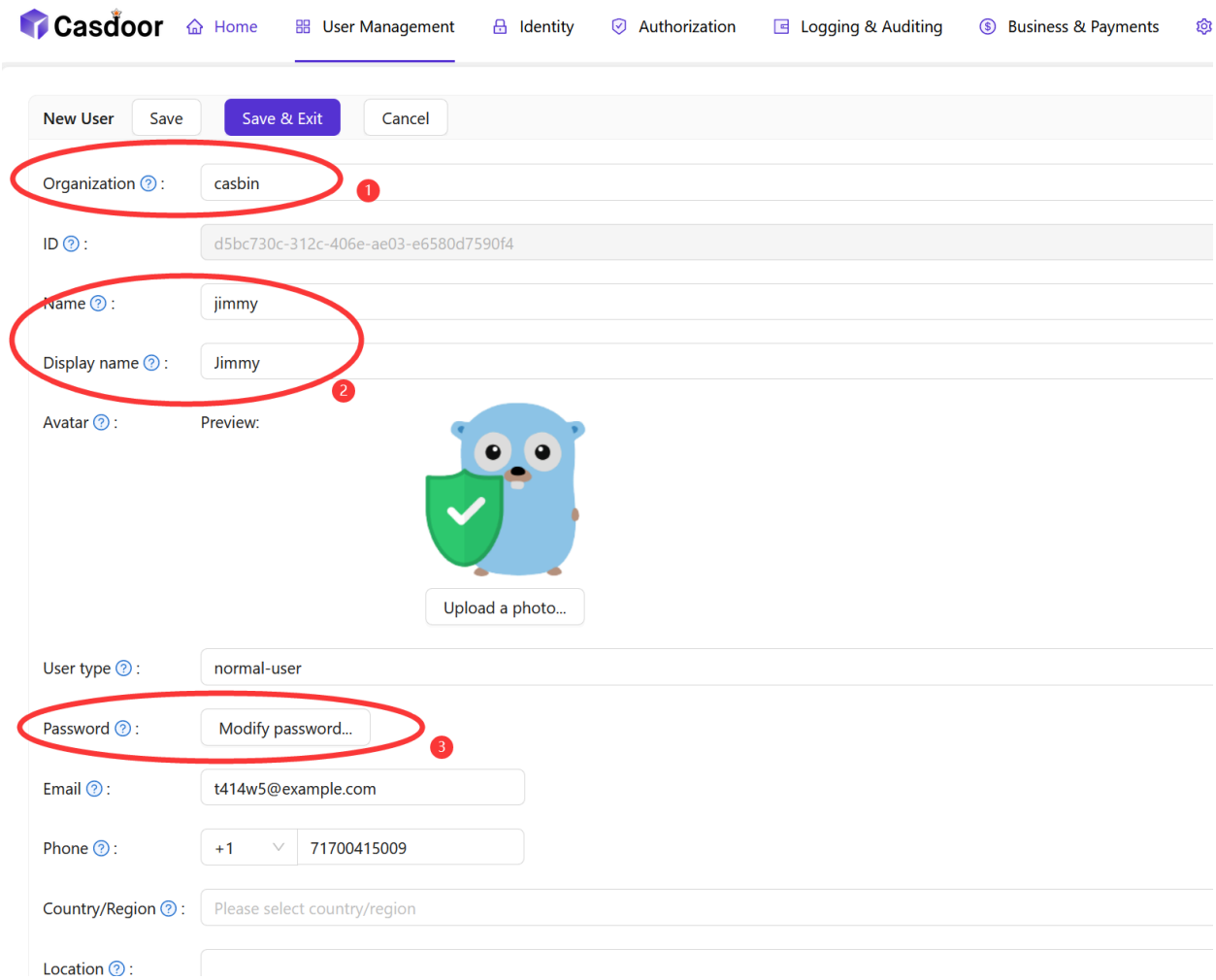
Step 4.1: Add a user

Click the `Add` button to add a user.



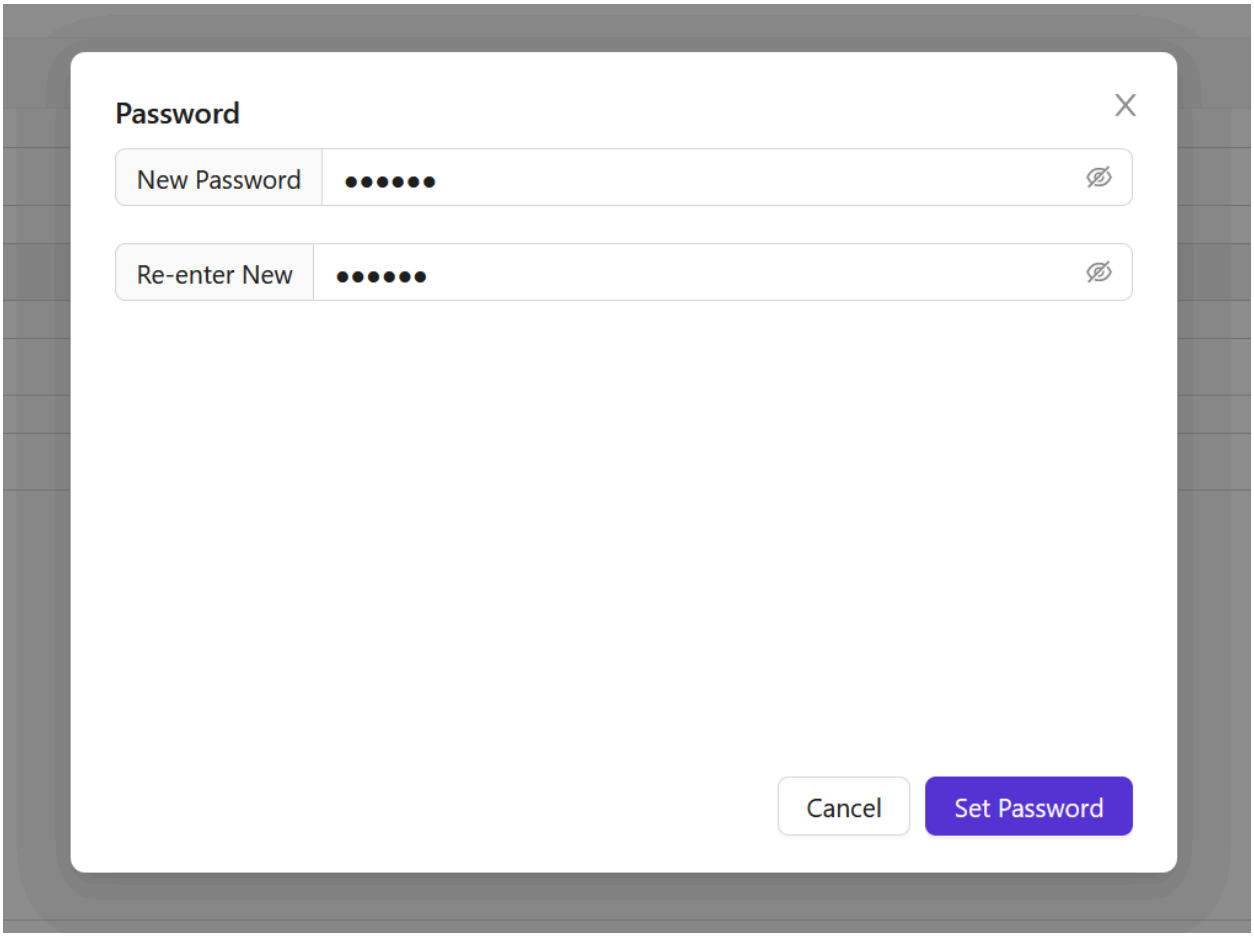
Step 4.2: Fill in the user information

Fill in the user information and click the `Save & Exit` button.



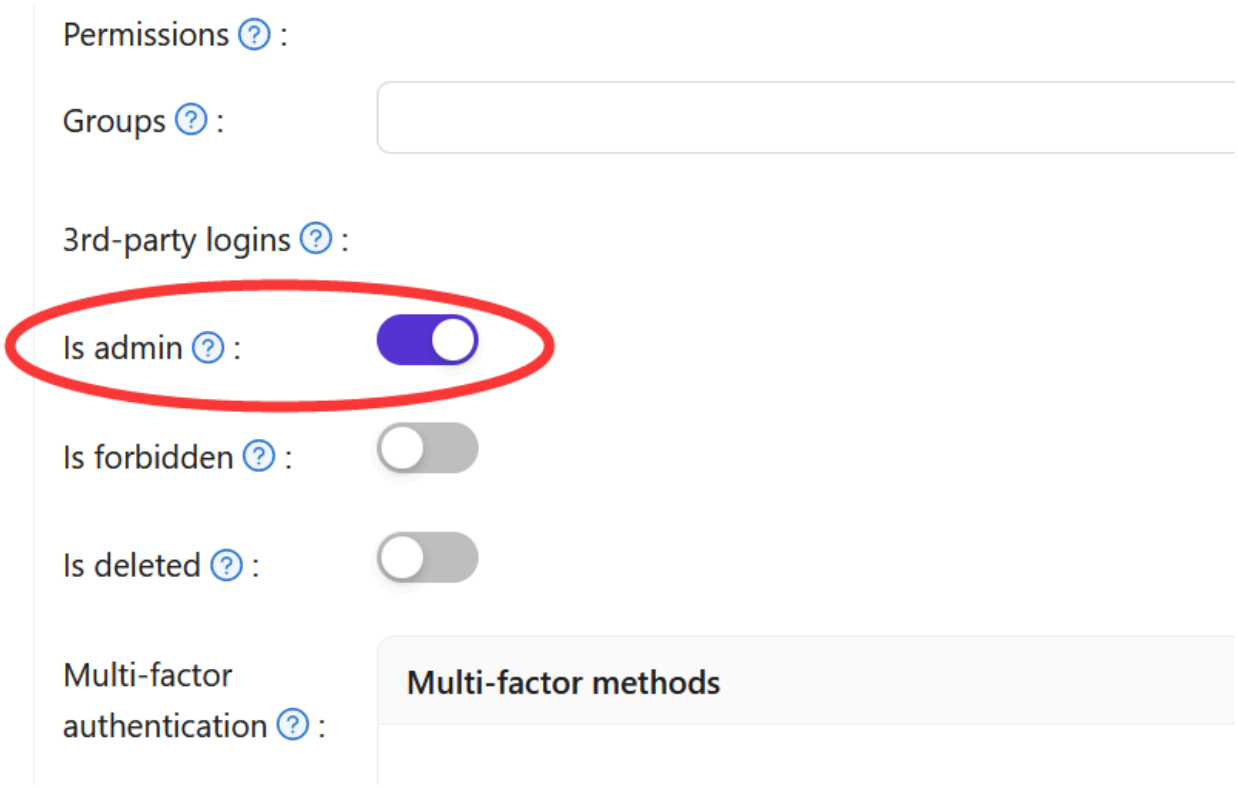
- Password

You can set the user's password by clicking the `Modify password` button.



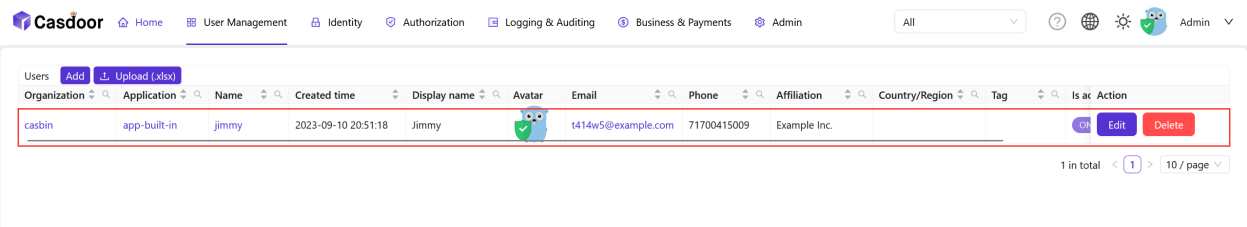
- Admin

You can set the user's admin permission by clicking the `Is admin` button.



Step 4.3: View the user

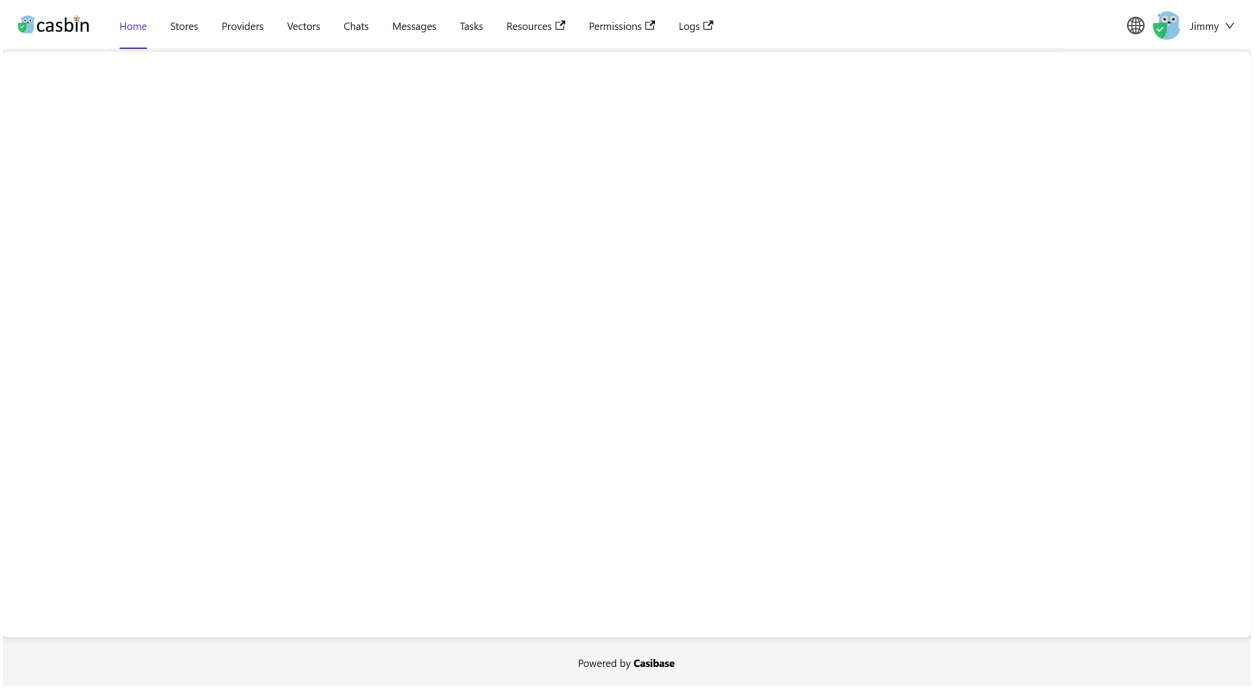
After adding the user, you can view the user information.



Step 5: Deploy Casibase

Like Casdoor, you can deploy Casibase by following the Casibase Deployment Guide.

Once you've deployed Casibase, you'll look like this:



Don't worry if you see a blank page in the beginning. In the next chapter, we will learn how to initialize data in Casdoor and Casibase.

Add a Storage Provider

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of integrating a storage provider with Casibase, our powerful knowledge base system.

Introduction

Adding a storage provider to Casibase enables you to efficiently manage and store data, making it an essential component for your knowledge base system.

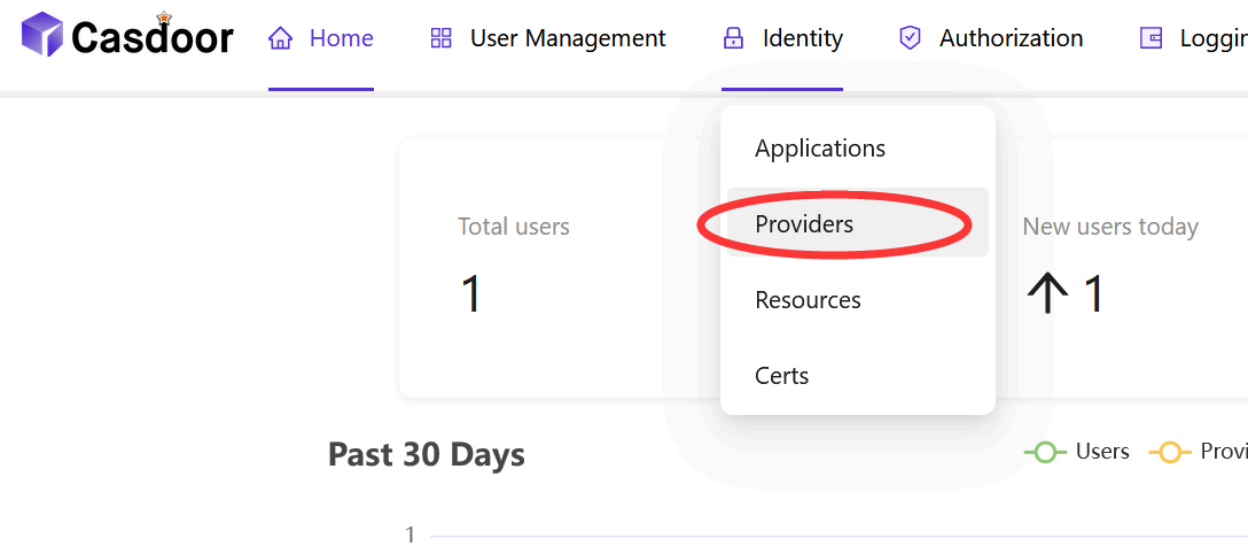
If you're new to integrating storage providers, don't worry. We've broken down the process into simple steps that anyone can follow.

Step 1: Deploy Casdoor and Casibase

In the last chapter, we learned how to deploy Casdoor and Casibase. If you haven't done so already, please refer to the Deploy Casdoor and Casibase tutorial.

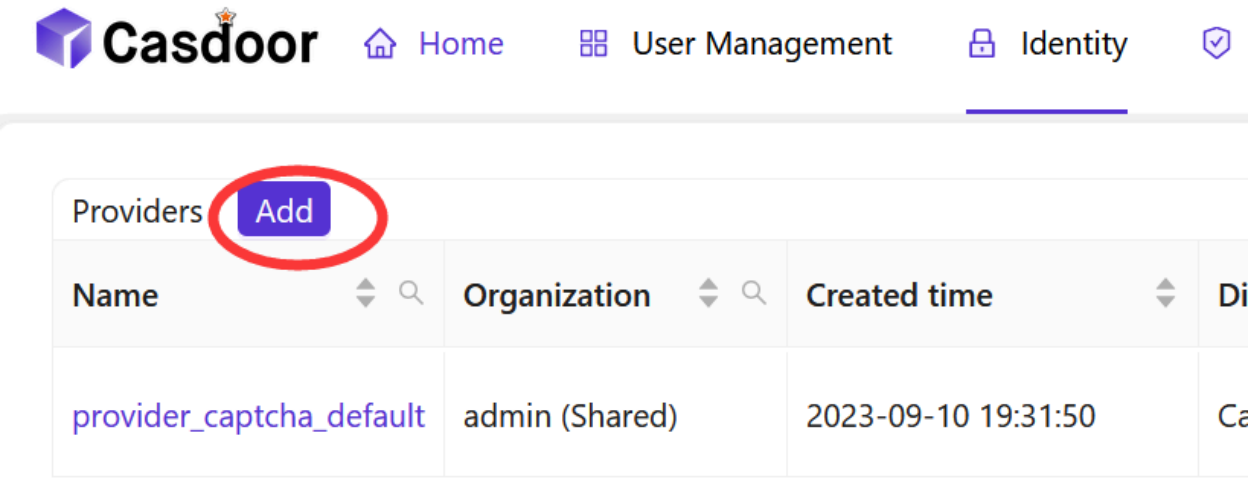
Step 2: Add a New Storage Provider

Storage providers are used to store data. They can be added in Casdoor by clicking the `Identity - Providers` button on the home page.



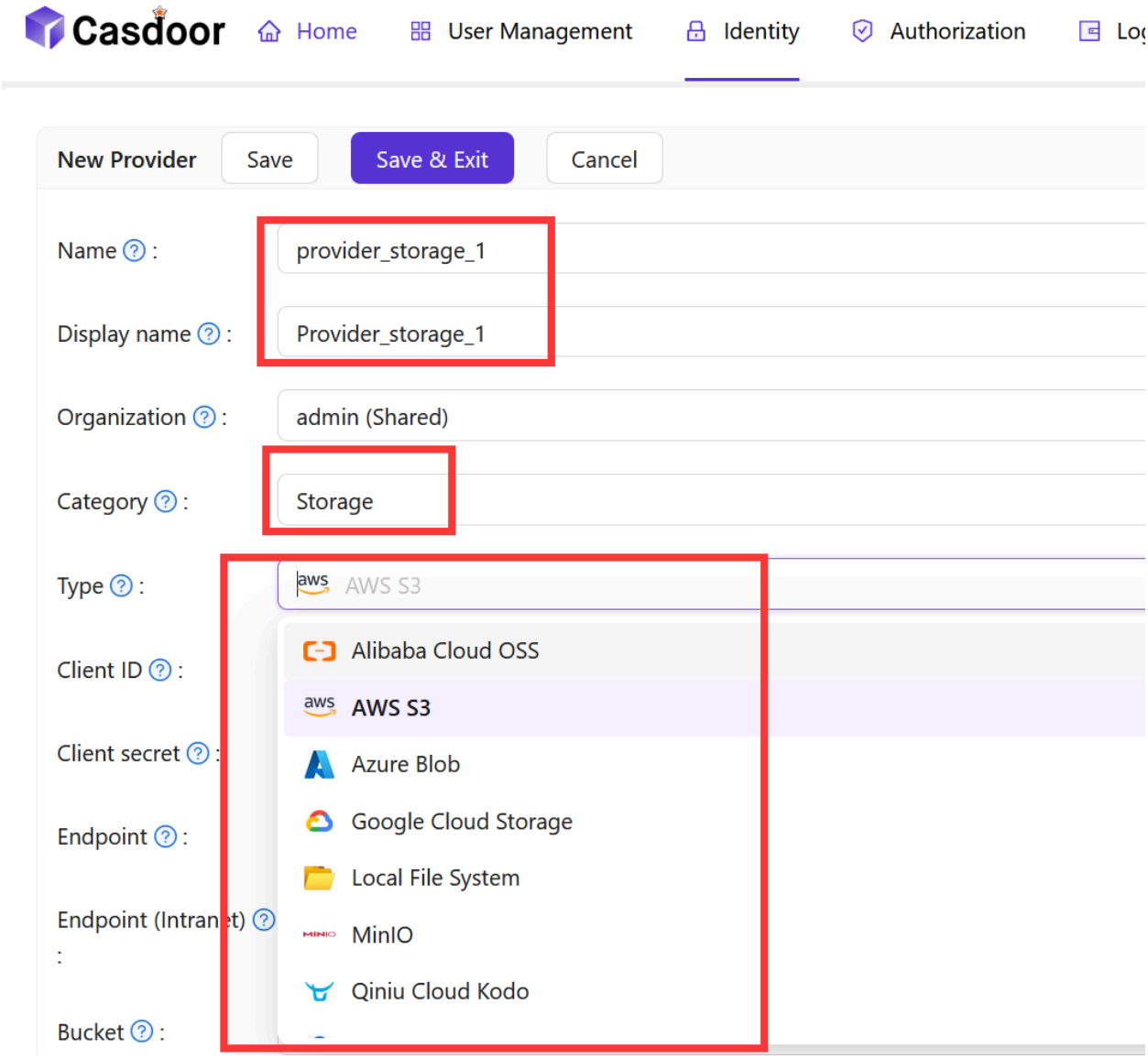
Step 2.1: Add a storage provider

Click the `Add` button to add a storage provider.



Step 2.2: Fill in the storage provider information

Fill in the storage provider information and click the `Save & Exit` button.



TIP

Casdoor supports many storage providers, including:

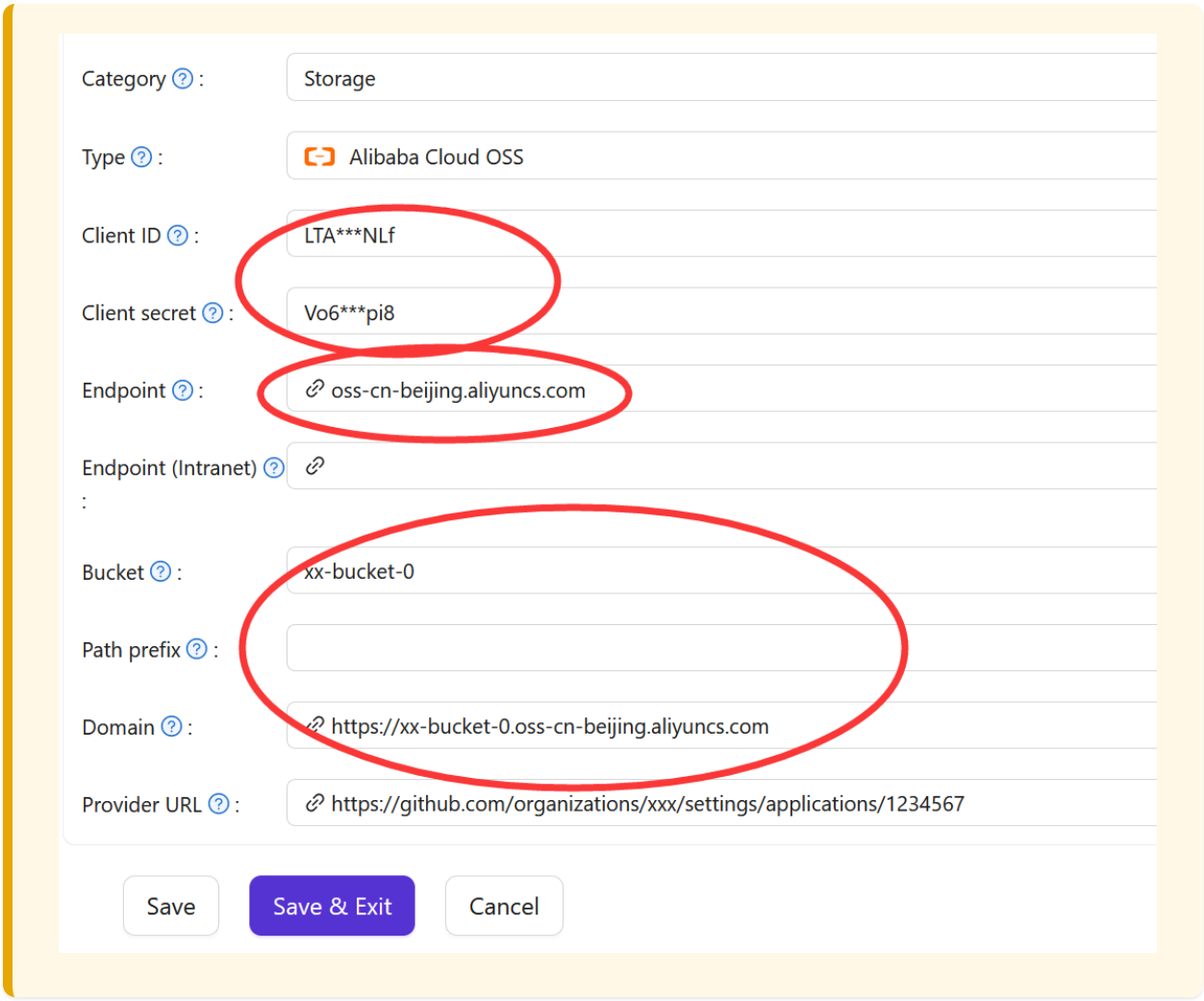
- AWS S3
- Azure Blob
- Google Cloud Storage
- MinIO

- Qiniu Cloud Kodo
- Alibaba Cloud OSS .....

Example

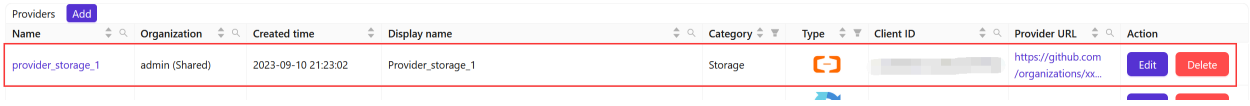
Add an Aliyun OSS storage provider

> ⚠ CAUTION
>
> - Client ID: The AccessKey ID of your Aliyun OSS account.
> - Client Secret: The AccessKey Secret of your Aliyun OSS account.
>
> `****` is the placeholder for your Aliyun OSS account information.



Step 2.3: View the storage provider

After adding the storage provider, you can view the storage provider information.

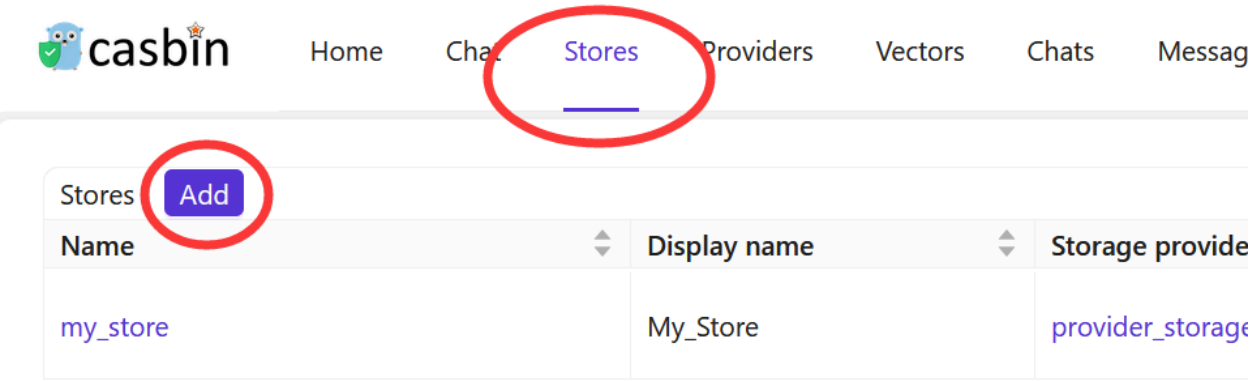


Step 3: Add a New Storage Provider to Casibase

In Casibase, you can add a storage provider to manage your data. You can add a storage provider by clicking the `Providers - Add` button on the home page.

Step 3.1: Add a storage provider

Click the `Add` button to add a storage provider.

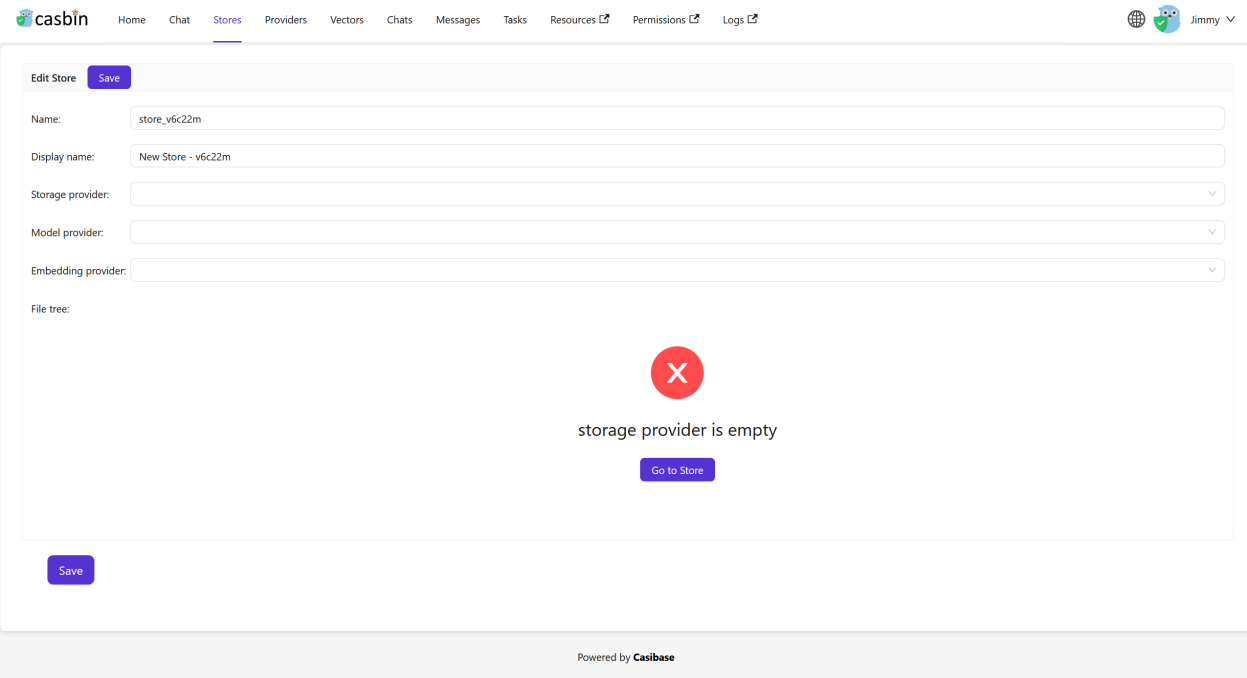


Step 3.2: Fill in the storage provider information

Fill in the storage provider information and click the `Save & Exit` button.

If Edit page is not displayed, please click the `Edit` button.

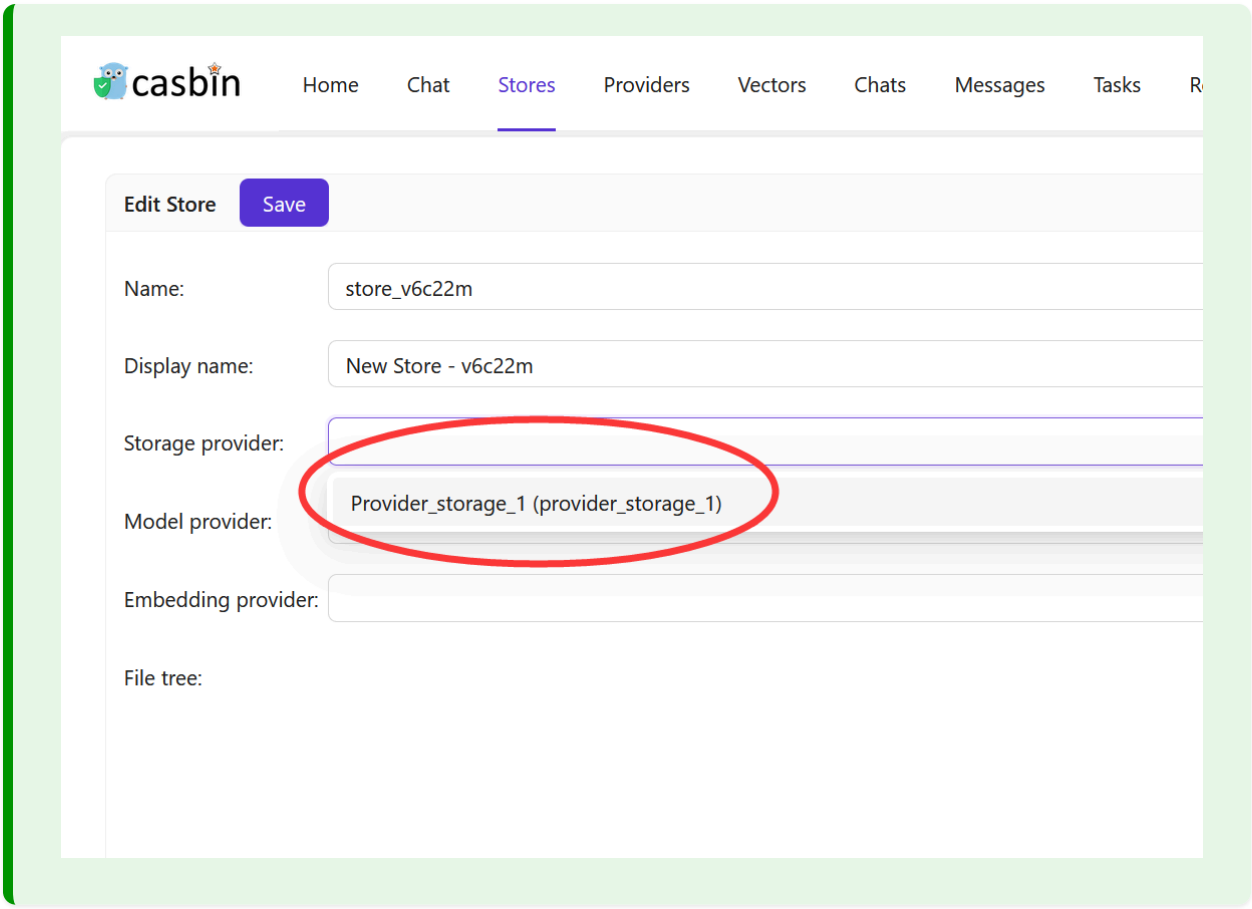
You'll see the following page:



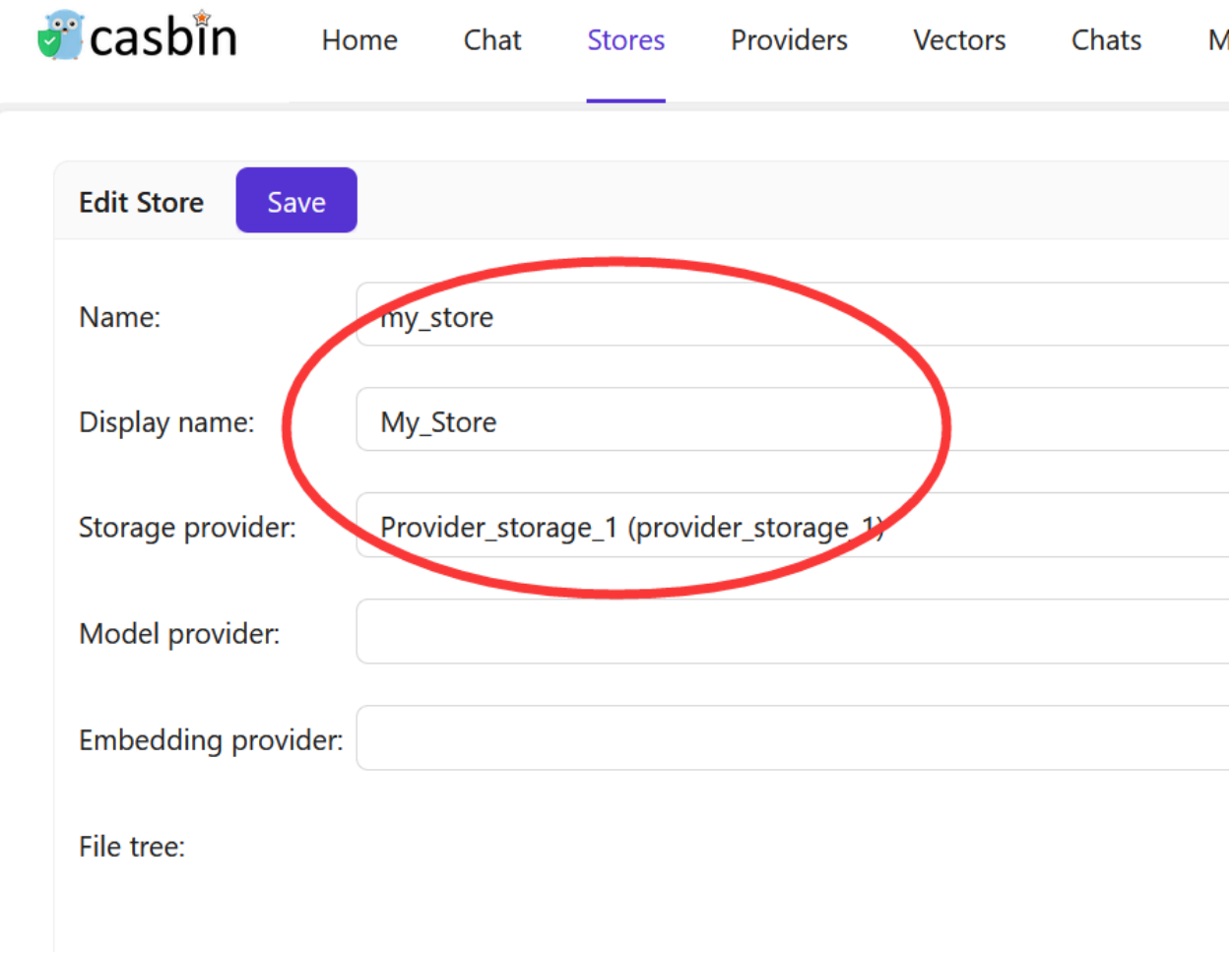
TIP

Storage providers comes from Casdoor. You can add a storage provider in Casdoor and then add it to Casibase.

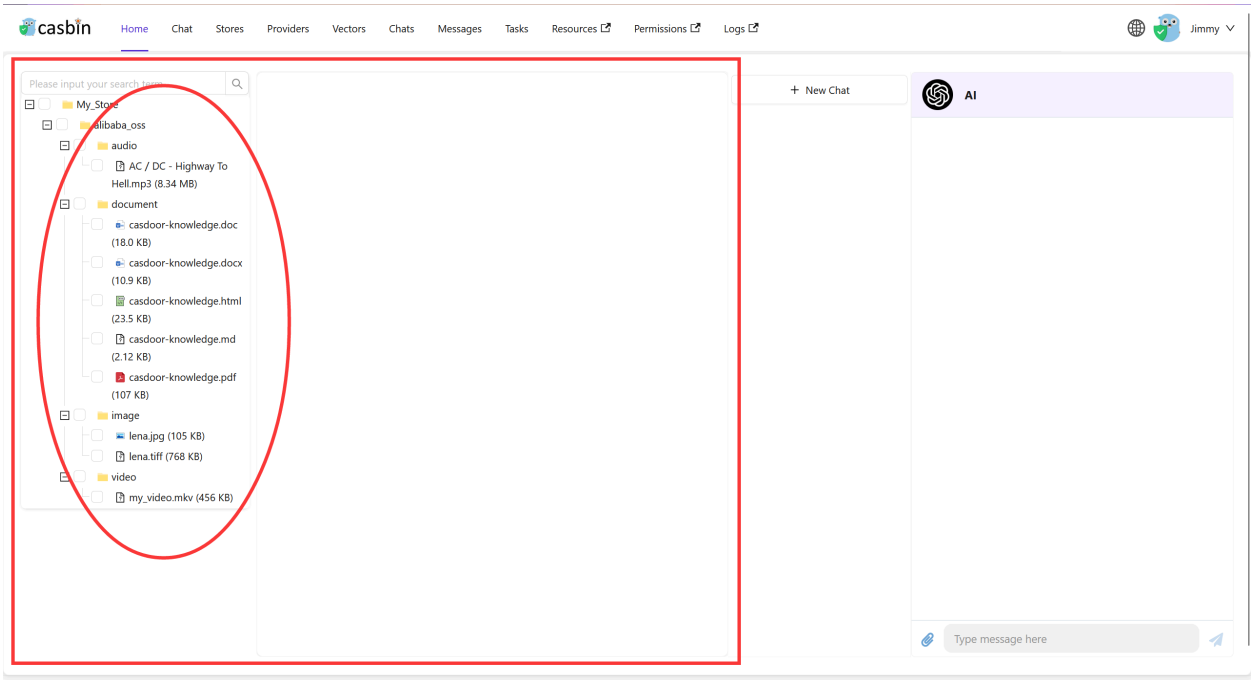
Refer to Step 2: Add a New Storage Provider for more information.



Store Example



Save the configuration, return to the home page, and you'll see the file-tree of the storage provider.



Now you can manage your data in Casibase.

In the next chapter, we'll learn how to add a `AI model provider` to Casibase.

Add a Model Provider

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of integrating a model provider with Casibase, our powerful knowledge base system.

Introduction

Adding a model provider to Casibase enables you to enhance its functionality by incorporating machine learning models and AI capabilities. Model providers allow you to analyze and process data within your knowledge base system, making it more intelligent and efficient.

If you're new to integrating model providers, don't worry. We've broken down the process into simple steps that anyone can follow.

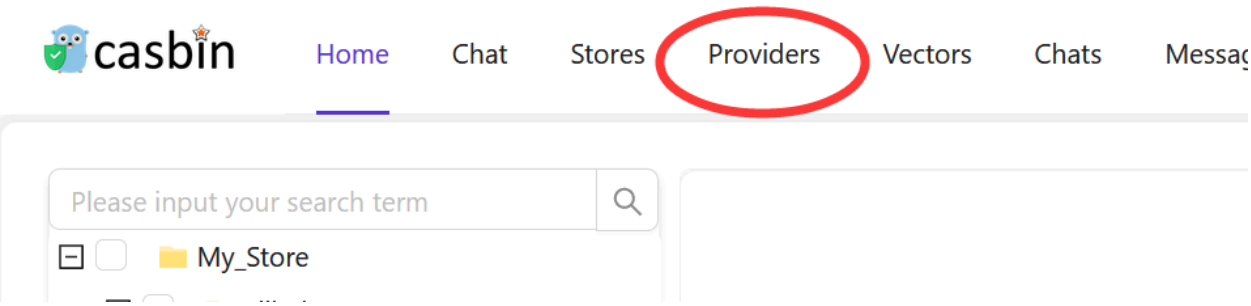
Step 1: Deploy Casdoor and Casibase

Before you can add a model provider, make sure you have Casdoor and Casibase deployed. If you haven't done this yet, please refer to the Deploy Casdoor and Casibase tutorial in our previous documentation.

Step 2: Add a New Model Provider

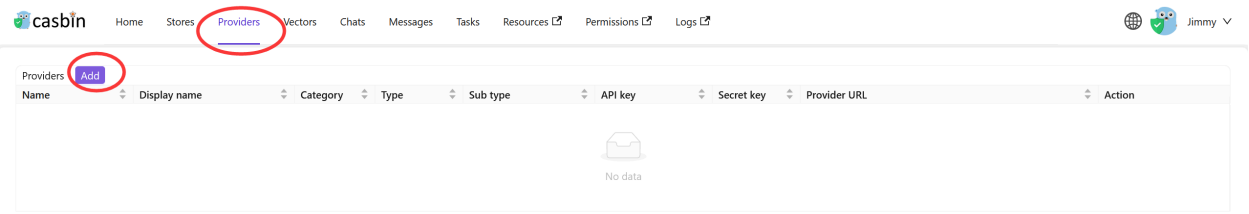
Model providers are used to integrate LLM into Casibase. You can add them by following these steps:

Click the `Providers` button on the home page.



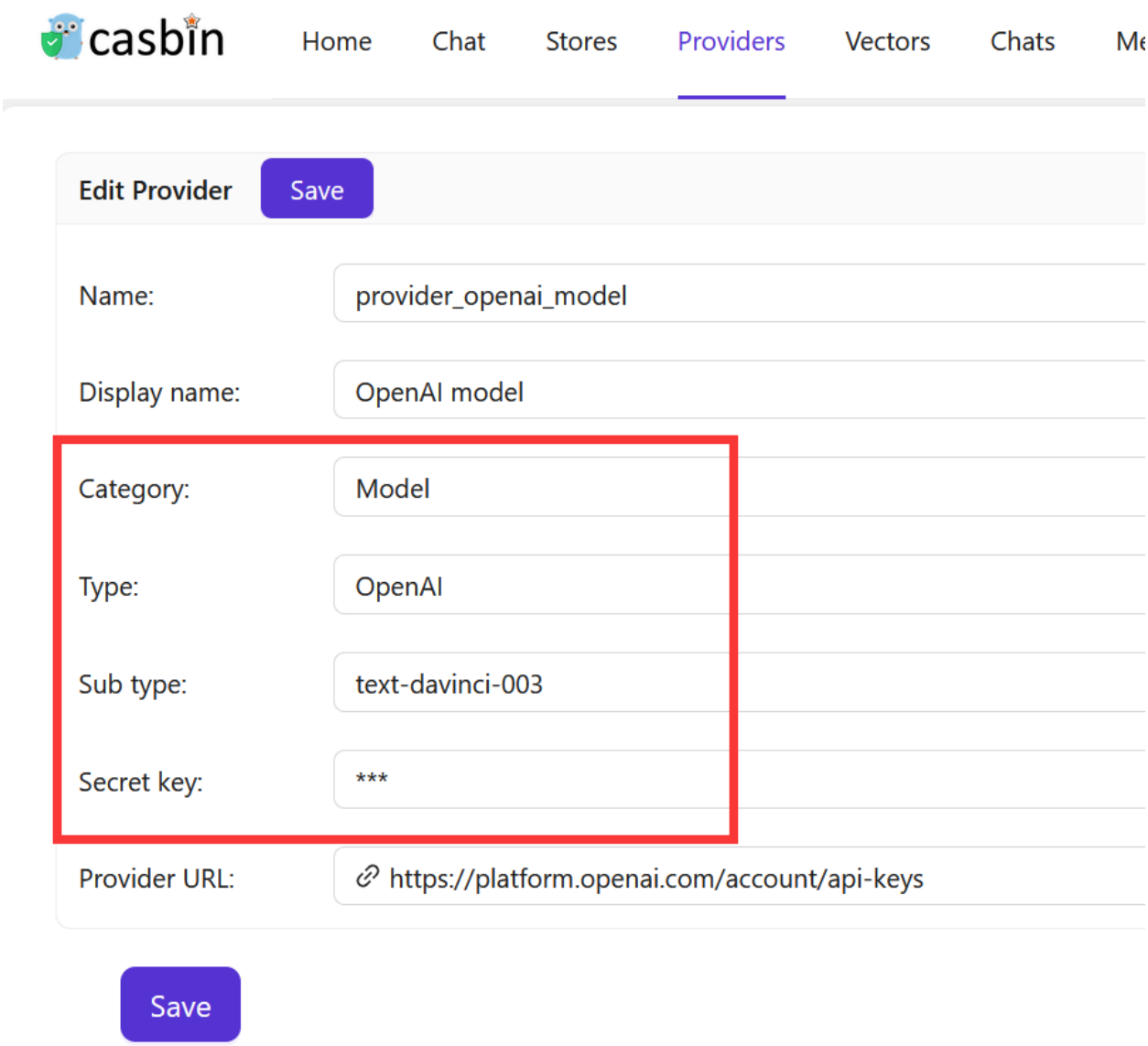
Step 2.1: Add a Model Provider

Click the `Add` button to add a model provider.



Step 2.2: Fill in Model Provider Details

Fill in the model provider details and click the `Save & Exit` button.



TIP

Casibase supports many model providers, including:

- Hugging Face
  - meta-llama/Llama-2-7b

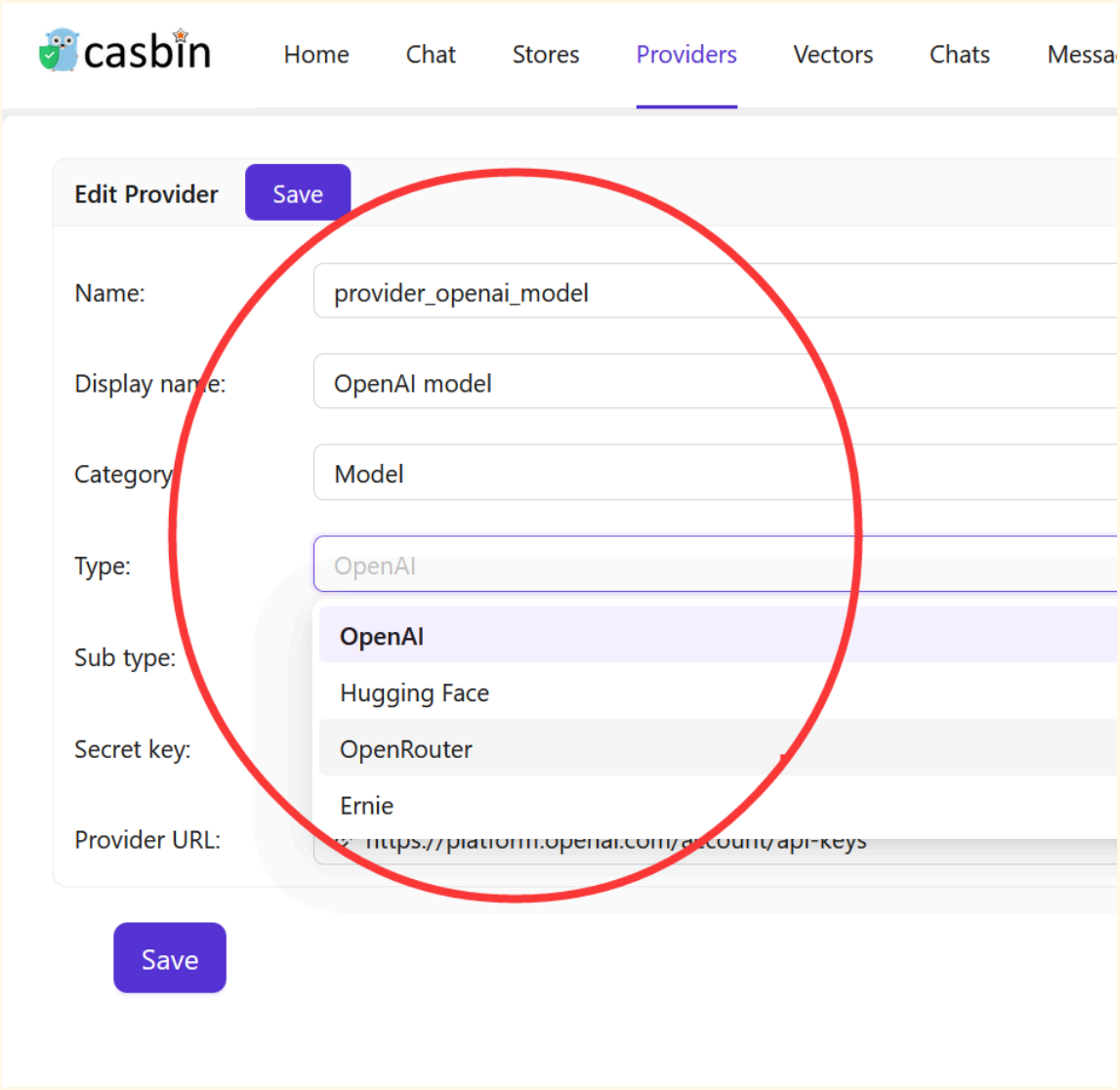
    - THUDM/chatglm2-6b
    - baichuan-inc/Baichuan2-13B-chat
    - gpt2
    - ……
- OpenRouter
    - anthropic/claude-2
    - palm-2-chat-bison
    - palm-2-codechat-bison
    - openai/gpt-4
    - ……
- OpenAI
    - text-davinci-003
    - gpt-3.5-turbo
    - gpt-4
    - ……

⚠ CAUTION

- Category: The first-level category of the model provider. For example, `Model` and `Embedding`.
- Type: The second-level category of the model provider. For example, `OpenAI` and `Hugging Face`.
- SecretKey: The secret key of your OpenAI account.

Example

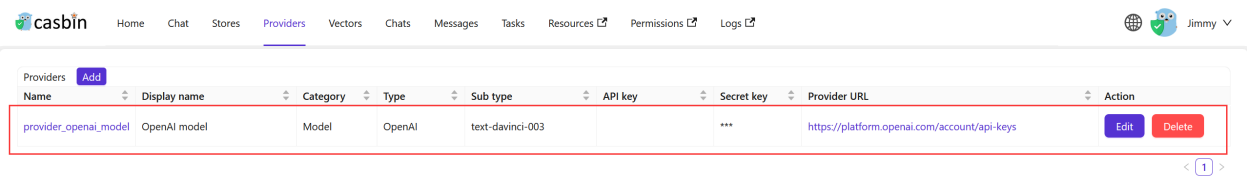
Add a OpenAI model provider



> **CAUTION**
>
> Some models don't support **streaming-output**. Known models that support streaming-output include:
>
> - `gpt-3.5-turbo-0613`

After adding a model provider, you can use it to analyze and process data in Casibase using chatbots, question answering, and other AI capabilities.

Return model provider list page:



Now that you've added a model provider, you can use it to analyze and process data in Casibase using chatbots, question answering, and other AI capabilities.

In the next chapter, we'll learn how to add an embedding provider to Casibase.

Add an Embedding Provider

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of integrating an embedding provider with Casibase, our powerful knowledge base system.

Introduction

Embedding is a technique used to represent words and documents as vectors. Embedding providers allow you to analyze and process data within your knowledge base system, making it more intelligent and efficient.

Refer to the Core Concepts section of our previous documentation for more information about embedding.

In Casibase, you can add an embedding provider by following these steps:

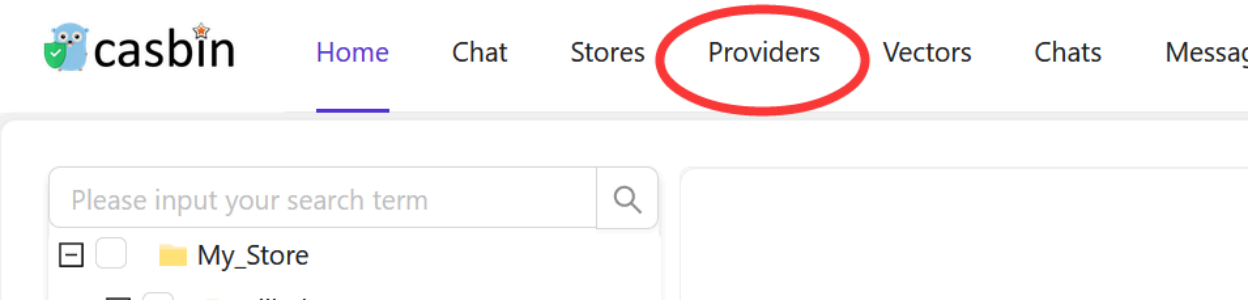
Step 1: Deploy Casdoor and Casibase

Before you can add an embedding provider, make sure you have Casdoor and Casibase deployed. If you haven't done this yet, please refer to the Deploy Casdoor and Casibase tutorial in our previous documentation.

Step 2: Add a New Embedding Provider

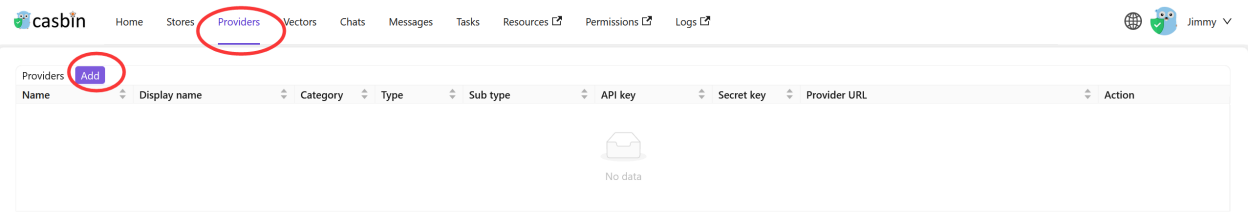
Embedding providers are used to integrate embedding into Casibase. You can add them by following these steps:

Click the `Providers` button on the home page.



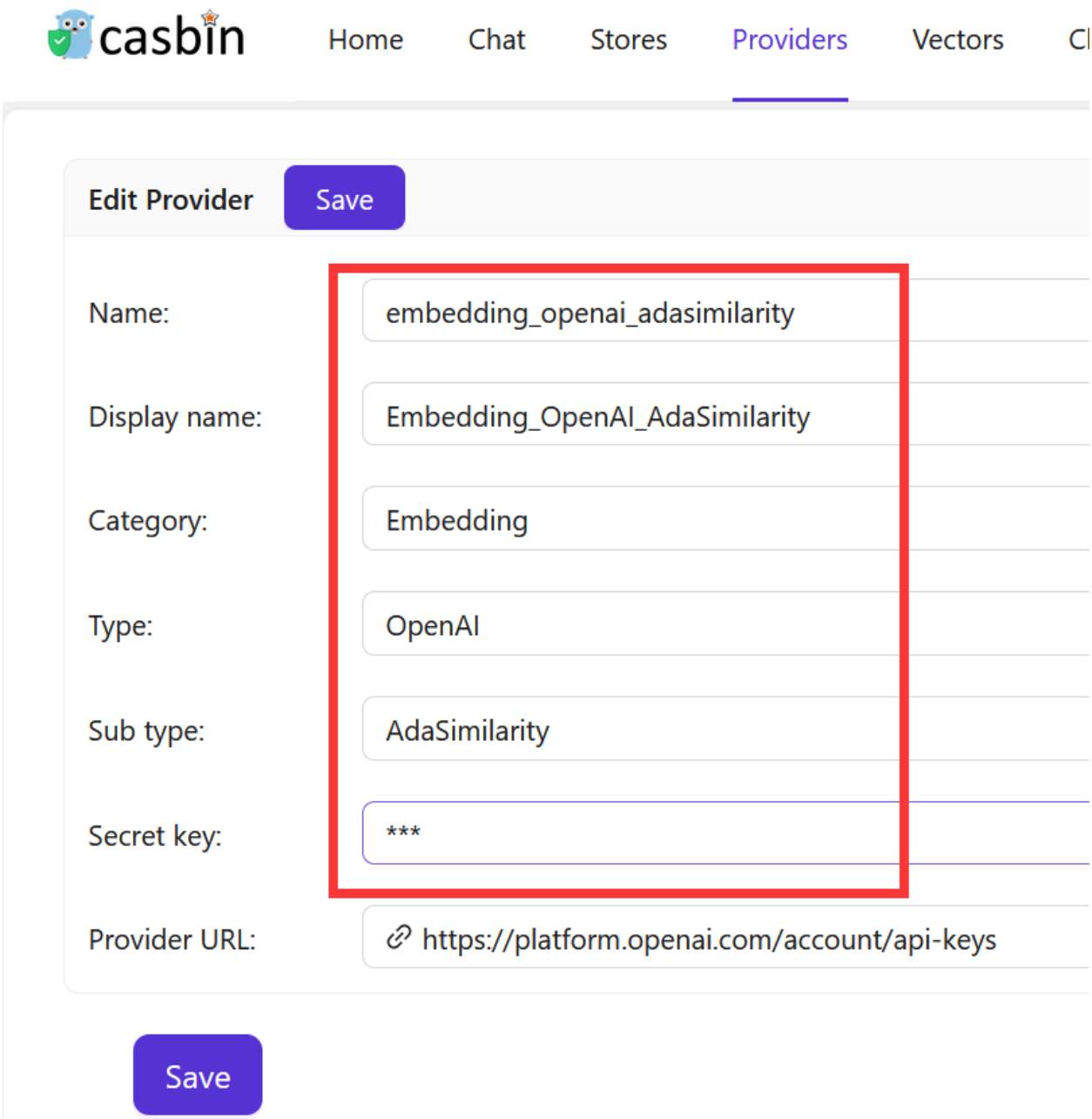
Step 2.1: Add an Embedding Provider

Click the `Add` button to add an embedding provider.



Step 2.2: Fill in Embedding Provider Details

Fill in the embedding provider details and click the `Save & Exit` button.

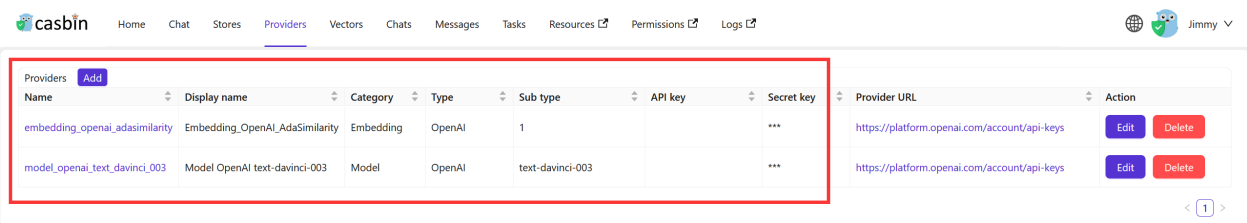


TIP

Same as the Model Provider section, Casibase supports many embedding providers, including:

- OpenAI
  - AdaSimilarity
  - DavinciSimilarity
  - AdaEmbedding2
  - ......
- Hugging Face
  - sentence-transformers/paraphrase-MiniLM-L6-v2
  - ......

Return providers list page:



Now, you can use the embedding provider to convert text to vectors.

After adding an embedding provider, you can use it to retrieve similar documents in Casibase. For more information, please refer to the Core Concepts section of our previous documentation.

In the next chapter, we will learn how to integrate storage providers, model providers, and embedding providers with Casibase.

Add a Store

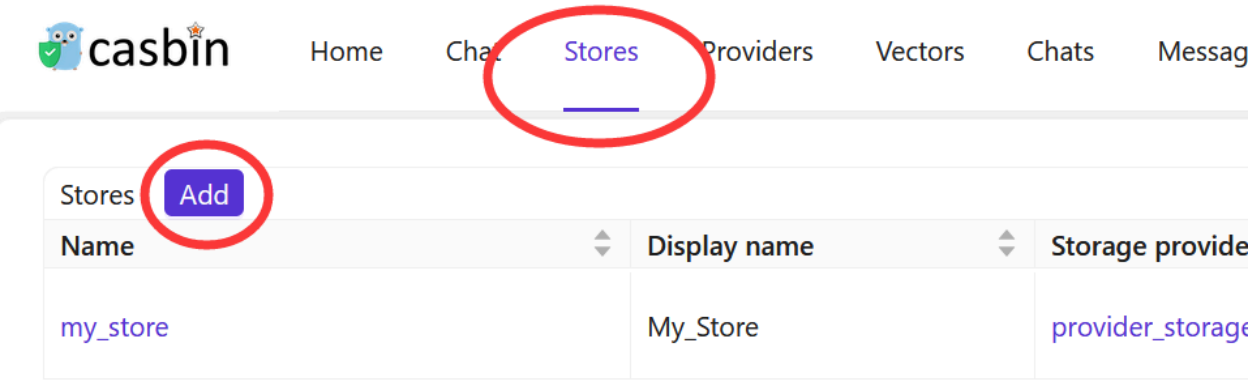
We have add a storage provider, a model provider, and a embedding provider. Now we need to configure a store to use these providers.

> ⚠ CAUTION
>
> This guide assumes that you have already deployed a Casibase knowledge base system. If you have not, please follow the Deploy Casdoor and Casibase guide.
>
> Besides, this guide assumes that you have already added a storage provider, a model provider, and a embedding provider. If you have not, please follow the Add a Storage Provider, Add a Model Provider, and Add a Embedding Provider guides.

Step 1: Add a New Store

Stores are used to integrate storage, model, and embedding providers into Casibase. You can add them by following these steps:

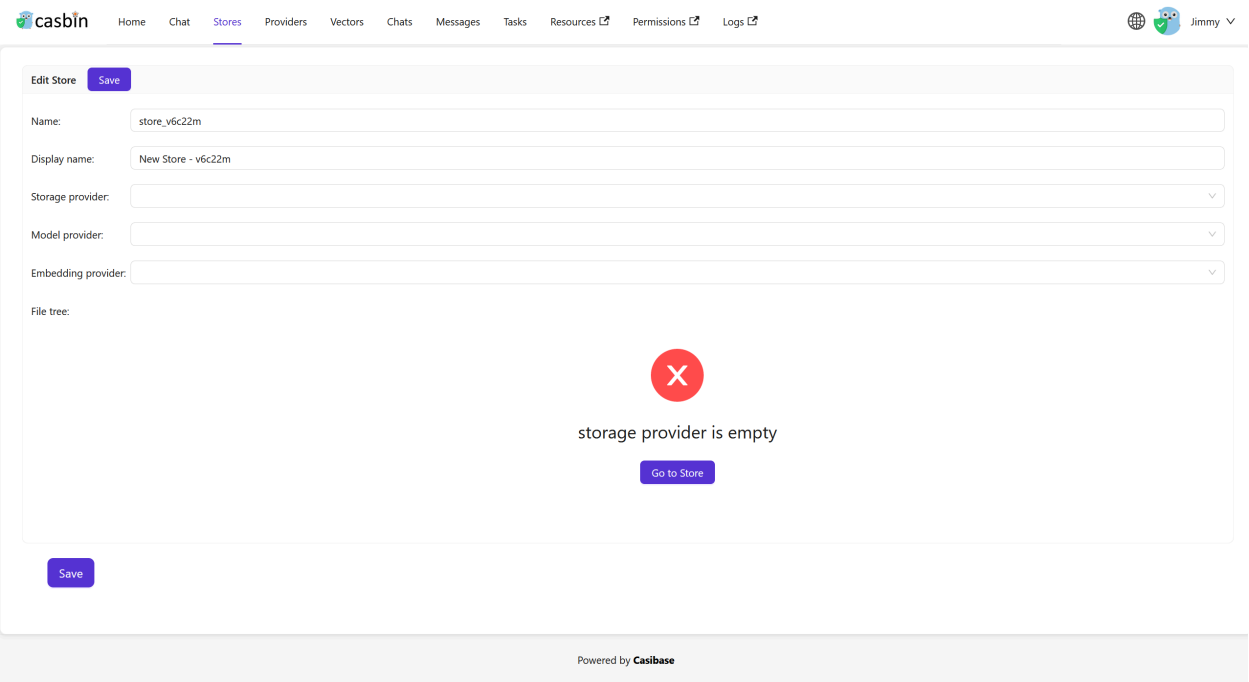
Click the `Stores` button on the home page and then click the `Add` button to add a store.



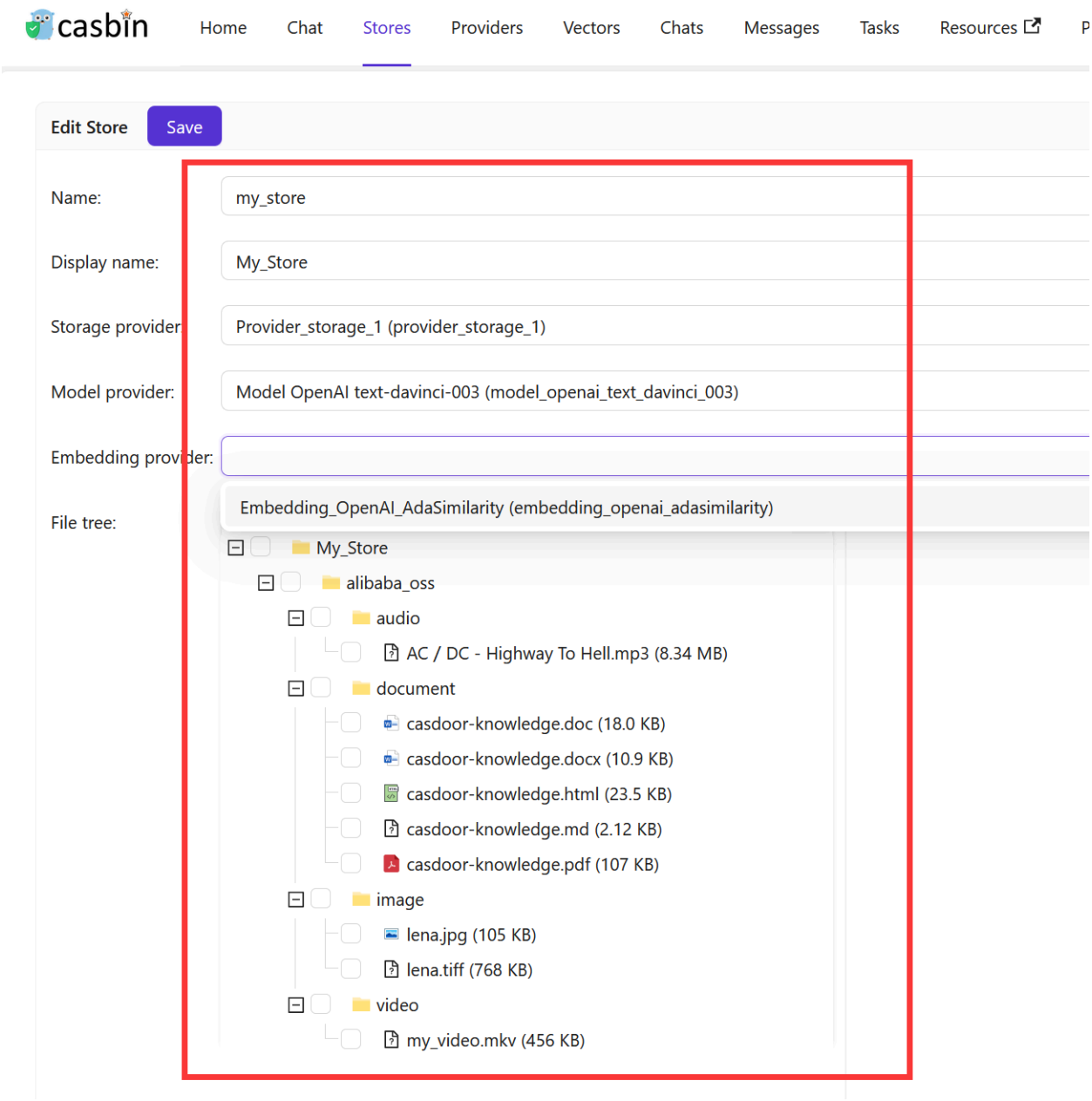
Step 2: Fill in Store Details

Fill in the store details and click the `Save & Exit` button.

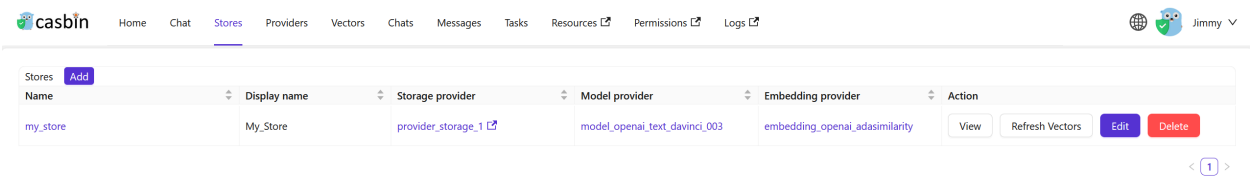
This part we have done if you follow the Add a Storage Provider guides.



Select the storage provider, model provider, and embedding provider you added before.



Click the `Save & Exit` button and return stores list page:



Now, you can use the store to store knowledge base data, convert text to vectors, and chat with the chatbot.

In the next section, we will learn how to chat with the chatbot in Casibase.

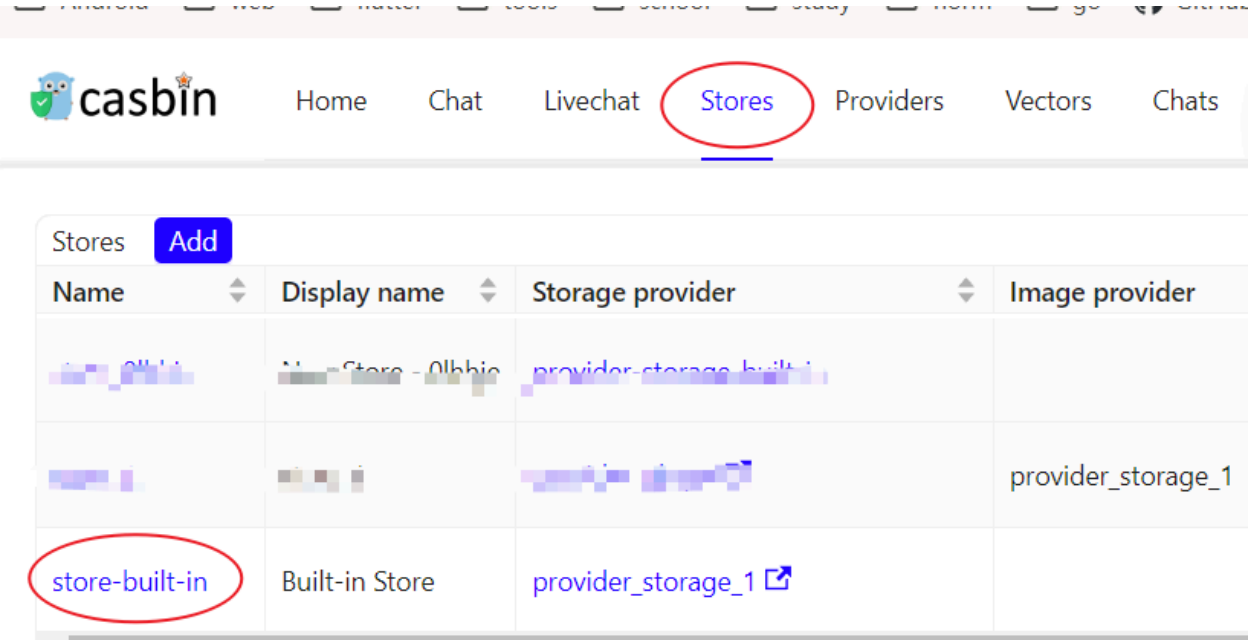
Support Multi-Store

The multi-store mode provides users with different models, suggestions, and more within each distinct store.

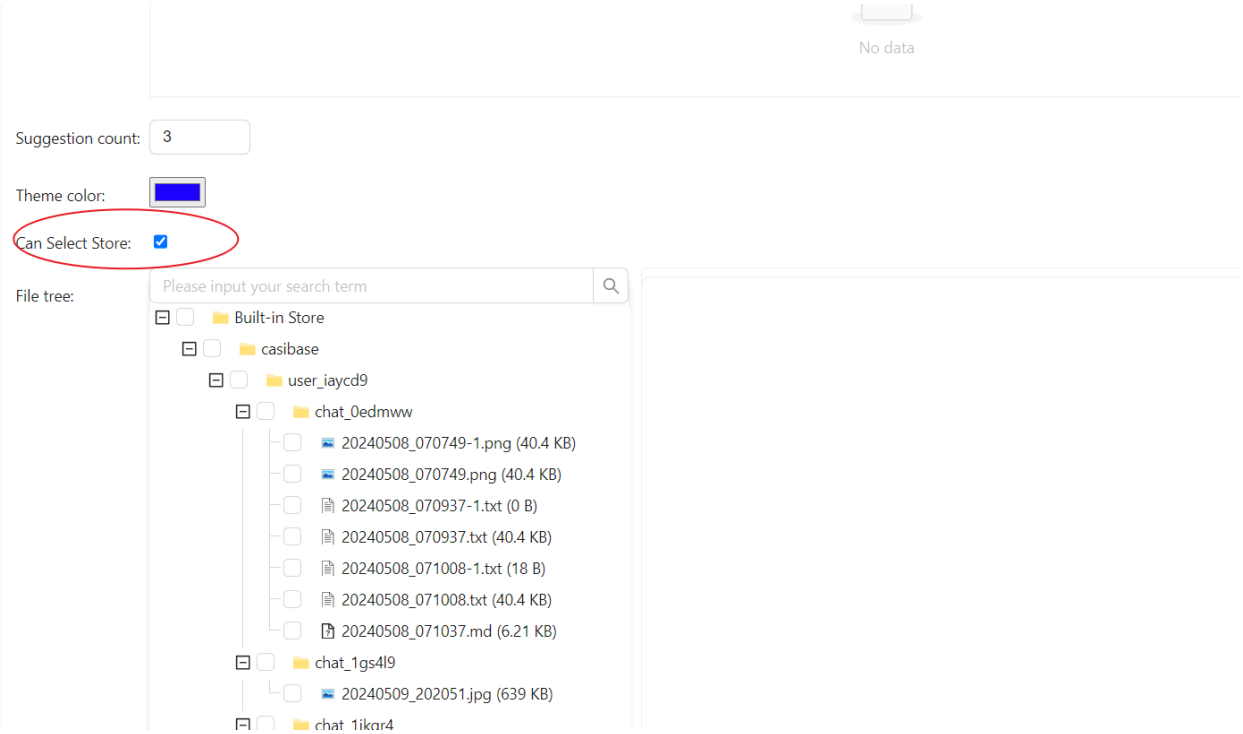
Step 1: Enable Multi-Store

First you should enable multi-repository mode in the built-in store.

Click the `Stores` button on the home page and then click the `store-built-in` button to enter the store-built-in store.



Scroll down and find the `Can Select Store` field, tick it.

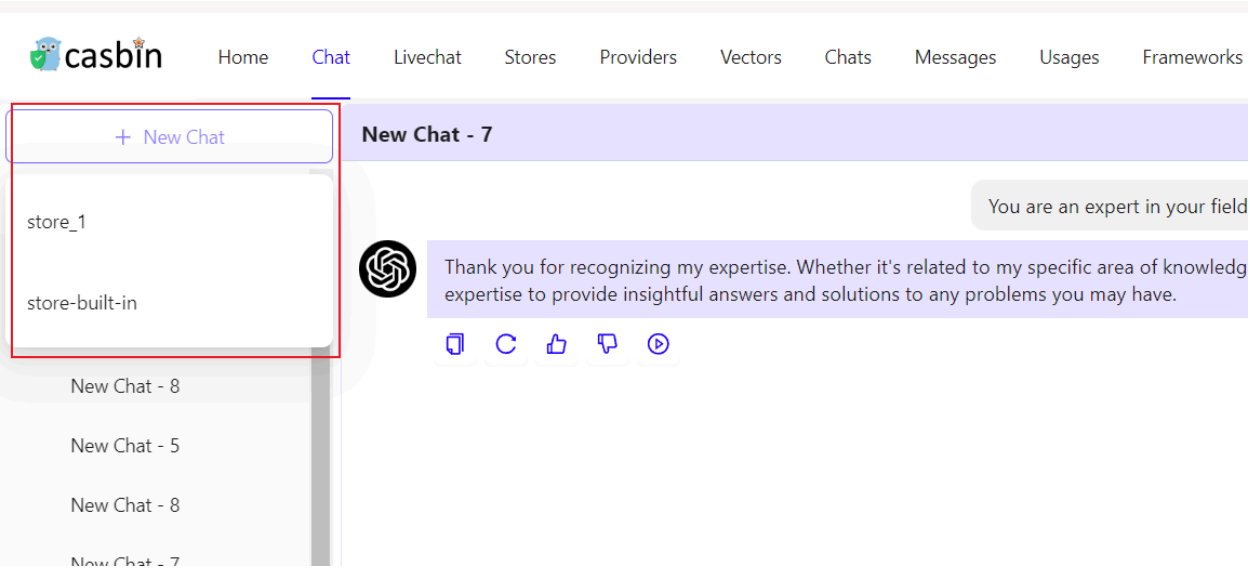


Step 2: Add Usable Store

The multi-store mode only provides usable stores. To make a store usable, you need to configure its storage provider, model provider, and embedding provider.

Step 3: Select For Conversation

Casibase provides a very convenient method for selecting a store.



Just hover your mouse over "New Chat" and then you can select the Store you wish to use from the list that appears below.

If you click the "New Chat" button, the system will assign you a default Store.

Chat with AI

This document is a step-by-step tutorial designed for beginners. It will guide you through the process of implementing AI chat functionality in your Casibase knowledge base system.

Introduction

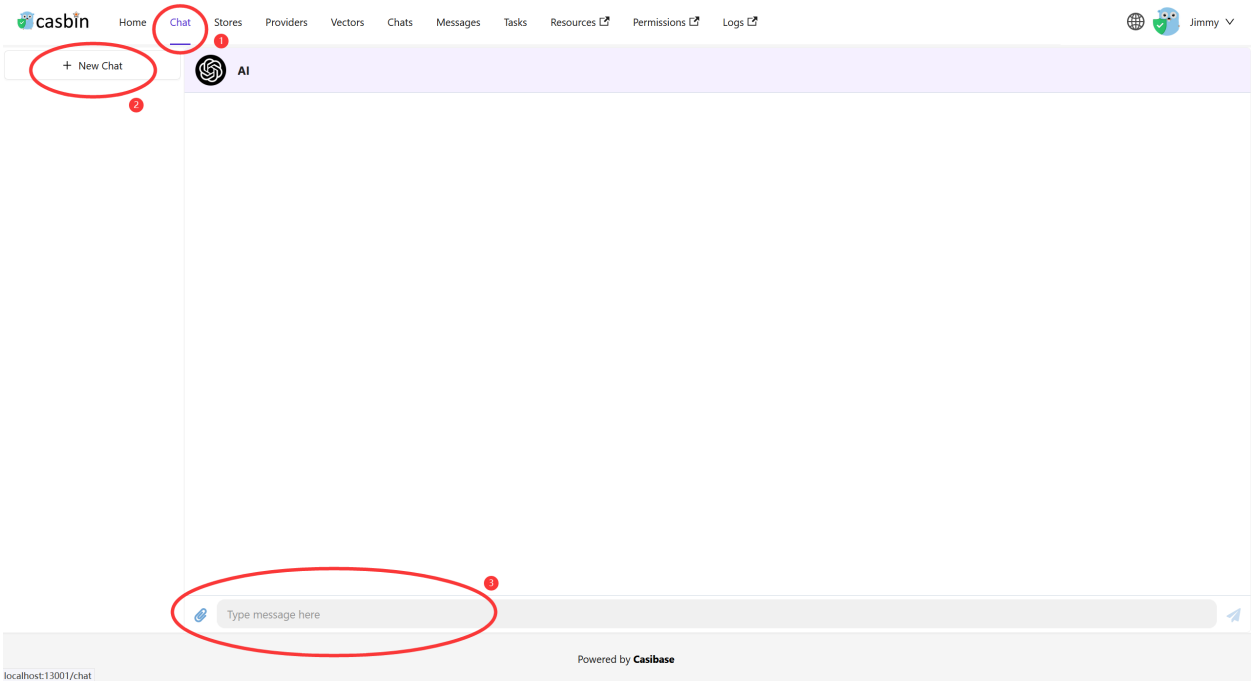
In previous sections, we have deployed Casdoor and Casibase, integrated a storage provider, a model provider, and a embedding provider with Casibase, and added a store to use these providers.

Refer to the Add a Store section of our previous documentation for more information about stores.

Now, let's implement AI chat functionality in Casibase.

Step 1: Add a New Chat

Click the `Chats` button on the home page and then click the `New Chat` button to add a chat.



Step 2: Send a Message

Write a message and click the `Send` button to send it.

Step 3: Knowledge Base Chat

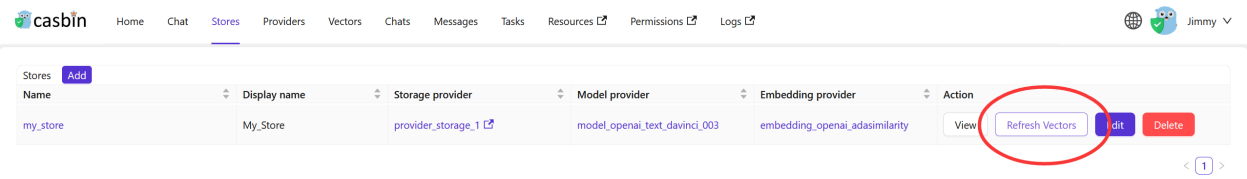
Addtionally, you can chat with the chatbot in the knowledge base.

There are some requirements for the knowledge base chat:

- The knowledge base must have a store.
- The store must have a model provider.
- The store must have a embedding provider.

- The store must have a storage provider.
- The storage provider must have a readable document (e.g. a markdown file, docx file and pdf file).

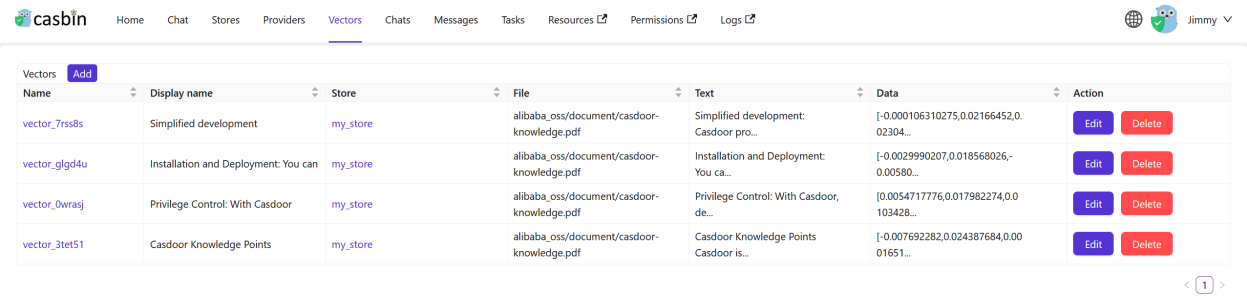
Once you have met these requirements, you can return to the `Stores` page and click the `Refresh Vectors` button to embedding the knowledge base data.



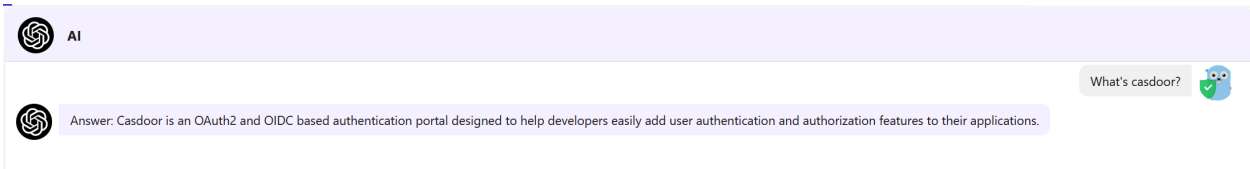
The button will be disabled when the embedding is in progress.

After the embedding is complete, you can click the `Vectors` button in the navigation bar to view the vectors.

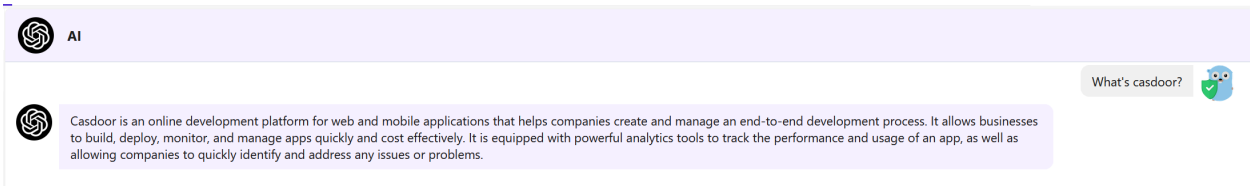
Result:



Let's **chat with the chatbot in the knowledge base.**



Compare the results with **non-knowledge base chat**:



> ⚠ CAUTION
>
> The embedding rate is related two factors:
>
> - The documents in the knowledge base.
>   - Documents number: The more documents, the longer the embedding time.
>   - Documents size: The larger the document size, the longer the embedding time.
> - The embedding provider.
>   - Api rate limit: The more api rate limit, the faster the embedding speed.
>   - Api concurrency: The more api concurrency, the faster the embedding speed.
>
> For example, if you use the OpenAI API as the embedding provider, the embedding rate is related to the OpenAI API rate limit and concurrency.

Conclusion

In this guide, we have learned how to implement AI chat functionality in Casibase.

Now, you can chat with the chatbot in Casibase. Enjoy it!

More information about Casibase can be found in the Core Concepts section of our documentation.