概览

Casnode is a forum developed by Casbin community. 详细架构：

名称	描述	工具	源代码
前端	Casnode Web 前端界面	JavaScript 和 React	https://github.com/casbin/casnode/tree/master/web
后端	Casnode RESTful API 后端	Golang + Beego + MySQL + Xorm	https://github.com/casbin/casnode/

Casbin社区将论坛作为官方论坛： Casnode 。

Casnode 是开源的，您可以在这里获得代码 。

Casnode 易用。 它在用户界面中有详细的描述，这样普通用户就可以轻松地从Casnode开始。 此文档是为想要充分使用Casnode的管理员准备的。

主要功能

这里有特殊的特征区分Casnode 和其他论坛。 以下是这些功能的简短介绍。 向前阅读更多细节。

📧 邮件列表

Casnode 很好地支持 Google 小组。 通过集成 Google group-crawler, 在设置谷歌群组配置后，将开始双向同步。 现在，Casnode 只能同步谷歌群组的对话，但可以将对话推送到任何其他邮件列表。

✨ 多平台

Casnode 支持计算机和移动访问。 前端用户界面适用于 PC 和移动设备。

🔍 内置搜索

Casnode 支持内置搜索当然也支持使用诸如Google等各种引擎搜索。

🖼️ 上传图片和附件

Casnode 支持拖放以上传图片和附件。 同时，每个账户都有自己的文件库和配额。 文件库中的文件也可以使用共享链接共享，从而更方便地发布图片和图片。 支持各种开放源码软件储存，如Alibaba云和腾讯云。

📢 网站广告

支持建立可以通过背景独立发送的网站广告。

🎯 服务器端渲染

Casnode 支持服务器端渲染，友好以搜索引擎SEO。

🗄 所有类型的数据库

Casnode 使用 xorm 连接到数据库。 您可以使用 MySQL 、 sqlite3 、 mymysql 和 Postgres for Casnode。

🌐 多语言

Casnode uses i18next and Crowdin to support multi-language. 现在， Casnode 支持中文、英文、法文、德文、俄文、日文和朝鲜文。 欢迎使用 PR 或文件作为一个问题来支持您的语言！ 任何翻译贡献都受到欢迎！

🚪 使用 Casdoor 管理成员

Casnode 在数据库中没有成员表。 Casnode 使用 Casdoor 来管理用户。 Casdoor是基于 OAuth2.0 的 SSO 平台。 通过Cassdoor，Casnode 支持各种第三方登录/注册方法。 注册方法多种多样，手机、电子邮件、QQ、WeChat、GitHub、Facebook、Google、LinkedIn、DingTalk、Gitee、wecom和GitLab。 它还支持人类机器核查的图形核查代码。 如果您使用Casdoor 来管理您的组织，您的成员可以直接登录到Casnode， 无需再次注册。 要获取更多详细的功能，请前往 Casdoor。

安装

这将有助于您在服务器上部署Casnode 如果您想要通过 BT 面板或 Docker 安装 Casnode ， 请在 BT Panel and Docker 查看详细信息。

请先克隆Casnode

```
git clone https://github.com/casbin/casnode
```
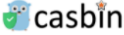
然后按照这些步骤轻松设置您自己的论坛!

生产环境

1. 设置Casdoor

Casnode 使用 Casdoor 来管理成员。 所以您需要在Casdoor 实例中创建一个组织和一个Casnode 的应用程序。

按照这些步骤设置Casdoor for Casnode:

- Casdoor配置向导(点击 此处 获取更多Casdoor详细信息)

- 登录到机构"built-in"

- 点击顶部栏中的 Organizations
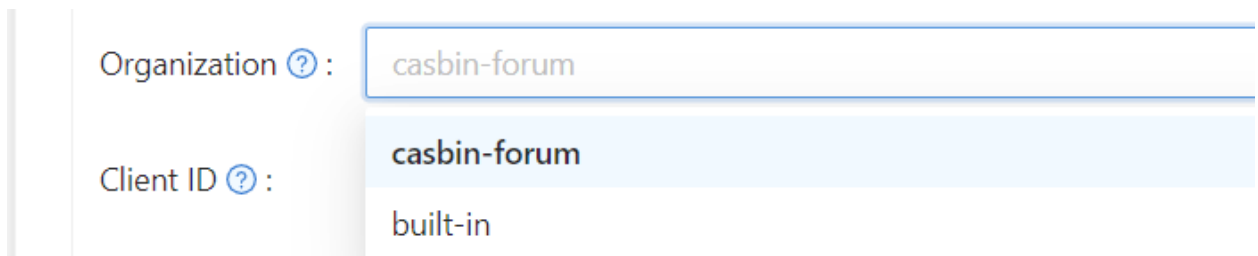
- 点击 **添加** 按钮

- 记住组织名称, 这里我使用 **casbin-forum** 作为我的组织名称:

- 点击 **顶部栏中的应用程序**

- 点击 **添加** 按钮

- 记住应用程序名称, 这里我使用论坛作为我的应用程序名称:

- 点击 **编辑**

- 选择您刚刚创建的组织作为应用程序组织

- 修改重定向URL到论坛 URL。 如果您处于正在开发的环境中，您的重定向URL是

http://localhost:3000/callback。 如果您处于生产环境，您的重定向URL是
http://yourip:7000/callback

- 点击 **保存** 并记住 `客户端 ID` 和 `客户端密钥`

2. 修改 `conf/app.conf`

以下是配置项目的解释：

Database connection:

Casnode database

```
driverName = mysql
dataSourceName = root:123@tcp(localhost:3306)/
dbName = casnode
```

Casdoor database

```
casdoorDbName = casdoor
```

> **TIP**
>
> Casdoor's `driverName` and `dataSourceName` are the same as casnode by default. If your Casdoor and Casnode are not in the same database, you can set up the casdoor database in casdoor/adapter.go
>
> Here we provide an example:
>
> 1. Add casdoor configuration in `conf/app.conf`:
>
> ```
> casdoorDriverName =
> casdoorDataSourceName =
> ```
>
> 2. Modify the `func InitCasdoorAdapter()`:
>
> ```
>     adapter =
> NewAdapter(beego.AppConfig.String("casdoorDriverName"),
> beego.AppConfig.String("casdoorDriverName"),
> beego.AppConfig.String("casdoorDbName"))
> ```

Object Storage Service (Casnode uses OSS to store resources):

```
OSSProvider = ""
accessKeyID     = ""
accessKeySecret = ""
OSSCustomDomain = ""
```

If you can not access Google in normal ways, you need to set up a http proxy here:

```
httpProxy = "127.0.0.1:10808"
```

Casdoor config:

```
# Your Casdoor endpoint in step 1
casdoorEndpoint = http://localhost:8000

# Client ID you copied in step 1
clientId = xxx

# Client Secret you copied in step 1
clientSecret = xxx

jwtSecret = CasdoorSecret

# Organization name in step 1
casdoorOrganization = "casbin-forum"
```

3. 修改 `web/src/Conf.js`

```
export const AuthConfig = {
  // Your Casdoor endpoint in step 1
  serverUrl: "http://localhost:7001",

  // Client ID you copied in step 1
  clientId: "014ae4bd048734ca2dea",

  // Application name you copied in step 1
  appName: "app-casbin-forum",
```

4. 构建前端

In folder `web`, run the following commands:

Yarn   npm

```
yarn install && yarn run build
```

```
npm install && npm run build
```

5. 重建后端

In repository root, run:

```
go build
./casnode
```

Then the Casnode app should run on port 7000. You can setup a nginx proxy pass to manage SSL or something else.

For most of site owners who want to develop a forum using Casnode, steps above is enough. But if you are a developer, want to contribute to Casnode, or modify the code to suit your own environment, then you can run Casnode in the developing mode. Please follow these steps to start developing mode:

开发环境

1. 执行上文第1-3步

4. 运行后退结束

```
go run main.go
```

5. 运行前端

In `web` folder:

Yarn npm

```
yarn install
yarn run start
```

```
npm install
npm run start
```

Now, Casnode runs its front end at port 3000 and runs it's back end at port 7000. You can modify the code and see what will happen.

> ⚠️ **CAUTION**
>
> The front end uses these codes to determine whether it is a dev mode:

```
export function initServerUrl() {
  const hostname = window.location.hostname;
  if (hostname === "localhost") {
    ServerUrl = `http://${hostname}:7000`;
  }
}
```

It means if hostname is `localhost`, then you are in dev mode. If not, then you are in productive mode. Port of the back end is not same in dev mode and productive mode, so please do not use `127.0.0.1` instead of `localhost` in your browser in dev mode.

bt-panel

title: BT panel description: Install Casnode under the Linux BT panel keywords: [bt panel] authors: [oranges-eating]

> ⚠️ **CAUTION**
>
> The tutorial environment is Ubuntu 20.04

在 Linux BT 面板下安装 Casnode

准备工作

After installing the BT panel, the browser visits the BT panel, selects the software store, searches for and installs MySQL, and then searches for node, you can see that there is a PM2 manager, install the PM2 manager.

After the installation is complete, disconnect from the server or restart the server, node will be automatically written into the environment variable.

Enter `git --version` to make sure git is , if the prompts Command `git` not found, use `apt-get install git` to install git.

> 💡 **TIP**
>
> To access the Casnode successfully, you need to open the 7001 and 8000

> port.

安装 Golang

The root user executes the following commands to download and decompress the Go binary file to the **/usr/local** directory.

```
wget -c https://dl.google.com/go/go.16.5.linux-amd64.tar.gz -O
- | sudo tar -xz -C /usr/local
```

Then we need to add Golang to the environment variables, edit `/etc/profile`, add the following code in the last line of the file.

```
导出 GOROOT=/usr/local/go
导出 PATH=$PATH:$GOROOT/ bin
```

Then use command `source /etc/profile` to make the newly added environment variables work.

Now, enter `go version`, you will see the go version, and we installed it successfully. If you can't connect to GitHub, you can set up the mirror. The command is

```
go env -w GOPROXY=https://goproxy.cn,direct
```

Git clone Casnode & Casdoor

Next, execute the following commands in the folder where you want to store the project.

```
git clone https://github.com/casbin/casdoog.git
git clone https://github.com/casbin/casnode.git
```

Now, you can see there are two folders, Casnode and Casdoor.

配置 Cassdoor

运行Casdoor

We first configure Casdoor.

```
cd casdoor
go build main.go
```

Then edit `conf/app.conf`, find

```
dataSourceName = root:123@tcp(localhost:3306)/
```

Change MySQL password provided by the BT panel as 123.

```
cd web
npm install
npm building
cd ...
sudo nohup ./main &
```

配置 Casdoor 中的 Casnode

Now that Casdoor has been configured, visit http://your-ip:8000 to configure Casnode.

The default administrator login account is `admin/123`.

Click Organization, then click Add, click Edit for the added organization, and change the name to the organization name you want. Here I set it to casbin-forum, and then click Save.

Click Applications, then click Add, for the application you just added, click Edit, change the name to the application name you want, I changed it to app-casbin-forum. Click on the organization, select the organization you just added, my organization is casbin- forum. Click Redirect URLs, modify the link in the box to http://your-ip:7000/callback.Finally, remember the Client ID and Client Secret, and click Save.

Click Users, click Add, then click Edit, modify the added user, click Organization, select casbin-forum, and click is admin. Finally click Save, now your organization has an administrator account.

配置 Casnode

Next we configure in Casnode.

```
cd casnode
go build main.go
```

Edit `conf/app.conf`, find

```
dataSourceName = root:123@tcp(localhost:3306)/
```

Change MySQL password provided by the BT panel to 123, then find

casdoorEndpoint, modify it to http://your-ip:8000 (Casdoor backend address), find **ClientId** and **ClientSecret**, and modify them to the previously remembered Application client id and client secret, find casdoorOrganization, modify the organization name to your set. Save and exit.

Edit `web/src/Conf.js`, modify `serverUrl` to http://your-ip:8000 (Casdoor front-end access address), modify `ClientId` to the ClientId of the application just added, modify `appName` to the set application name, and modify `organizationName` to the set organization name.

```
npm install
npm run build
cd ..
nohup ./main &
```

Next visit **http://your-ip:7000**, click login, enter the account you added before, user_1/123, you have now successfully logged in to Casnode.

For more settings please see Casnode.

docker

title: Docker description: Install casnode through docker keywords: [docker] authors: [oranges-eating]

> ⚠️ **CAUTION**
>
> The tutorial environment is Ubuntu 20.0.4 .

通过 docker 安装 casnode

准备工作

安装 docker 和 docker-compose

Install Docker and Docker-compose, you see [docker](#) and [docker-compose](#)

Clone casnode & casdoor

Next, clone Casdoor and Casndoe from GitHub.

```
git clone https://github.com/casbin/casdoog.git
git clone https://github.com/casbin/casnode.git
```

Now, you can see two folders, `casnode` and `casdoor`.

配置casdoor

运行casdoor

We first configure casdoor.

Edit `conf/app.conf`, modify dataSourceName = root:123@tcp(localhost:3306)/ to dataSourceName = root:123@tcp(db:3306)/

```
docker-compose up
```

> 💡 **TIP**
>
> mysql and casdoor are in different docker containers.

在腰果中配置 casnode

Now that Casdoor has been configured, visit http://your-ip:8000 to configure Casnode.

The default administrator login account is `admin/123`.

Click Organization, then click Add, click Edit for the added organization, and change the name to the organization name you want. Here I set it to casbin-forum, and then click Save.

Click Applications, then click Add, for the application you just added, click Edit, change the name to the application name you want, I changed it to app-casbin-forum.Click on the organization, select the organization you just added, my organization is casbin- forum. Click Redirect URLs, modify the link in the box to `http://your-ip:7000/callback`.Finally, remember the Client ID and Client

Secret, and click Save.

Click Users, click Add, then click Edit, modify the added user, click Organization, select casbin-forum, and click is admin. Finally click Save, now your organization has an administrator account.

配置种子节点

Next we configure in Casnode.

Edit `conf/app.conf`, modify dataSourceName = root:123@tcp(localhost:3306)/ to **dataSourceName = root:123@tcp(db:3306)/** so that the data come from your database.

Then find `casdoorEndpoint`, modify it to `http://your-ip:8000` (Casdoor backend address), find `clientId` and `clientSecret`, and modify them to the previously remembered Application client id and client secret, find casdoorOrganization, modify the organization name to you set.

Edit `web/src/Conf.js` , modify `serverUrl` to http://your-ip:8000 (Casdoor front-end access address), modify `clientId` to the clientId of the application just added, modify `appName` to the set application name, and modify `organizationName` to the set organization name.

Next, run casnode with docker

停靠人员组成

Next visit http://your-ip:7000, click login, enter the account you added before, user_1/123, you have now successfully logged in to Casnode.

More settings reference [casnode.](#)

从 DiscuzX 迁移

Casnode 提供了很多Go 脚本来帮助用户将他们的论坛从 DiscuzX 3.x 迁移到 Casnode。 脚本位于: https://github.com/casbin/casnode/tree/master/discuzx

Xorm支持的数据库(例如MySQL) 由 Cansode 用于存储主题和回复等论坛数据。 Casnode 用于存储图像和附件文件的对象存储(作为Casdoor 存储提供商的形式)。
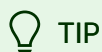
准备工作

您需要在进行迁移之前准备好以下环境:

1. 云端虚拟机（含4个核心和8GB内存，8个核心和16GB更高）， 更快地使用内联网与数据库的连接。 此 VM 用于运行迁移脚本。
2. 上面虚拟机中带有迁移脚本的 Casnode git 仓库(Casnode 实例可以运行或停止)。
3. 正在运行的 Casdoor 实例 (在同一个云端虚拟机中更优越，速度更快)， 至少有一个对象存储被配置为Casdoor 存储提供商。 这用于将DiscuzX的图像和附件文件上传到对象存储。
4. 您的 DiscuzX 实例已上线。

配置

首先根据他们的安装指南正确配置Casdoor 和 Casnode ， 确保他们在迁移前正常工作。

> 💡 **TIP**

> Let Casdoor and Casnode connect to the DB's Intranet URL. Let Casdoor's storage provider's endpoint be the Intranet URL of the cloud object storage. It will be much faster.

Configure the Casdoor database in Casnode's app.conf, so Casnode can directly connect to Casdoor's DB and create users. This will be faster than calling Casdoor's RESTful API to create users.

```
casdoorDbName = casdoor
```

Configure the migration script at: https://github.com/casbin/casnode/blob/master/discuzx/conf.go .

1. dbname: 你的 DiscuzX's 数据库名称
2. discuzxDomain: 你的 DiscuzX's 公共域名, 带有尾随斜杠的
3. discuzxAttachmentBaseUrl: 你的 DiscuzX的附件基URL, 带有尾随斜杠(你可以从你DiscuzX的附件文件 URL 获取)
4. 头像PoolBaseUrl: 不要更改

```
package discuzx

var dbname = "ultrax"
var discuzxDomain = "https://www.discuz.net/"
var discuzxAttachmentBaseUrl = "https://attachment.discuz.net/forum/"
```

> 💡 TIP
>
> In Casnode, we assume you use the same DB username and password for all 3 DBs: Casnode's DB, Casdoor's DB and DiscuzX's DB. So make sure

> this DB user can access all 3 DBs.

Migration

You may see the `XXXConcurrency` is defined at the top of the scripts, this is the number of the concurrent threads to do the migration. You can tune this value based on your environment. If it is too small, the migration will be very slow. If it is too high, the DB connections will be more likely to report "Connections too many" errors.

```
var SyncAvatarsConcurus = 20
```

> 💡 **TIP**
>
> The estimation time we provide in the following sections is measured in a machine with remote Internet connection with the DB and object storage. If your VM is in the same subnet with the DB and object storage, you will be much faster.

1. 用户迁移

To migrate all your DiscuzX's UCenter members to Casdoor's users:

Run `TestAddUsers` in: https://github.com/casbin/casnode/blob/master/discuzx/user_test.go

This step will roughly take 2 minutes for 60,000 users.

2. 用户头像迁移

To migrate all your DiscuzX's UCenter members' avatars to Casdoor's resources (via Casdoor's storage provider, backed by the cloud object storage):

Run `TestSyncAvatars` in: https://github.com/casbin/casnode/blob/master/discuzx/avatar_test.go

This step will roughly take 10 minutes for 60,000 users.

3. 论坛迁移

To migrate all your DiscuzX's forums to Casdoor's tabs and nodes:

Run `TestAddForums` in: https://github.com/casbin/casnode/blob/master/discuzx/forum_test.go

This step will usually finish in 2 seconds.

4. 线程和帖子迁移

To migrate all your DiscuzX's threads & posts to Casdoor's topics & replies:

Run `TestAddThreads` in: https://github.com/casbin/casnode/blob/master/discuzx/thread_test.go

This step will roughly take 7 minutes for 10,000 users.

最后完成

During the migration, you can keep the Casnode instance running at the same

time, so you can see the effect immediately by pressing `F5` in Casnode's public homepage.

If you encounter panic when running the migration scripts, contact the Casnode authors.

API 参考

📄 API 概述

Casnode API

API 概述

我们使用 swagger 来记录我们所有的 API，您可以在这里看到api docs 。

结构

📄 **概览**

Casnode's architecture

📄 **主包**

Casnode main packages

📄 **路由**

Filters in Casnode

📄 **控制器**

Use controllers to handle requests

📄 服务端渲染

Server Side Rendering

概览

Casnode 是一个开源项目。 一个活跃的社区就是它的活力。

本章针对的是想要为Casnode做出贡献的开发人员。 您将在这里学习种子节点的结构以及它如何运作。

结构

Casnode 有两个部分：前端和后端。

名称	描述	实用小工具	源代码
前端	Casnode Web 前端界面	JavaScript + React + Ant-Design	https://github.com/casbin/casnode/tree/master/web
后端	Casnode RESTful API 后端	Golang + Beego + MySQL + Xorm	https://github.com/casbin/casnode

正如我们前面提到的，在产品环境中，卡斯诺节点的前端是由后端构建和服务的。 在发展环境中，诺代日人为前端服务。

主包

Casnode后端有几个软件包。 The main function and the Beego framework call these packages when the program starts.

主要的

主要的包裹是卡斯诺德的入口。 我们在主包中做一些基本设置步骤:

- **设置一些全局变量**: 数据库适配器、http 客户端、开放源码软件适配器、Segmenter 和论坛基本信息 (论坛版本、在线号码、谷歌组 Crawlers)

- **设置过滤器：** API请求过滤器，搜索引擎机器人过滤器

- **会话数据：** Casnode 使用 Beego 会话来存储用户信息。 在主要函数中，写下这些行使用基于文件的会话:

```
Beego.BConfig.WebConfig.SessionProvider = "file"
beego.BConfig.WebConfig.Session.SessionProviderConfig = "./tmp"
beego.BConfig.WebConfig.SessionGCMaxLifetime = 360* 24 * 365
```

如果您想要在Beego使用另一个会话适配器，请参阅 Beego会话。

路由

实际上，`路由器` 是一个内置的Beego包件。`routers.init()` 程序启动时由框架运行。然而，我们在包裹中添加了一些过滤功能。

我们不会在这里说 `路由器/路由器`。Beego已经有一个很好的文档。我们将在这里介绍我们的过滤器。

路由器/过滤器.go

透明静态： 此过滤器将服务于静态文件。如果请求路径不是以 `/api/` 开头的，那么过滤器将为浏览器提供静态文件。过滤器将在 `web/build/` 中找到请求的文件，如果存在则服务于该文件。

新旧账户激活状态： 更新用户发送请求时的在线状态。

路由器/filter_ssr.go

我们使用 Chromedp 渲染搜索引擎机器人的页面。如果安装了 Chrome ，此文件中的函数将返回渲染页面到机器人。

我们使用正则表达式 `bot|slurp|bing|crawler` 来匹配请求的用户代理人。如果匹配，我们认为请求是由机器人发送的。

控制器

在 路由器/路由器 中，您可以找到很多行像是

```
Beego.Router("/api/get-topics", &controllers.ApiController{},
"GET:GetTopics")
```

请注意第三个参数 `"GET:GetTopics"`。 此字符串是HTTP 请求方法和请求处理器的函数名称的组合。 您可以在软件包 控制器 中找到一个名为 `GetTopics` 的函数。 并且它是请求的处理程序 `/api/get-topics`

您可以通过这种方式找到请求的相应函数。

> 💡 TIP
>
> If you are using **Goland**, you can press `Ctrl+Shift+F` (vscode use `Ctrl+F` instead), and search `func (c *ApiController) FunctionName` to locate to the function quickly.

服务端渲染

Casnode 支持 SSR。 SSR(Server-side rendering) 是一种在服务器上呈现客户端单页应用程序(SPA)，然后向客户端发送完全呈现的页面的常用技术。

这将允许动态组件作为静态HTML标记。 当索引无法正确处理 JavaScript 时，他的方法可以用于搜索引擎优化(SEO)。 在下载一个大型JavaScript包因网络缓慢而受到损害的情况下，它也可能是有益的。

Implementation

您可以在 `路由器/filter_ssr.go` 中看到服务器端渲染。我们使用 Chromedp 来渲染搜索引擎机器人的页面。 如果安装了 Chrome ，此文件中的函数将返回渲染页面到 bots。

我们使用正则表达式 `bot|slurp|bing|crawler` 来匹配请求的用户代理人。 如果匹配，我们认为请求是由机器人发送的。

Internationalization

Casnode 支持多种语言，将翻译部署到Crowdin，我们支持中文、法文、德文、俄文、日文和朝鲜文。

Casnode uses the official Crowdin cli to sync translations from Crowdin, if you want to add more languages supports, please propose in our community, and if you want to help us speed up the translating work, please help us translate on Crowdin.