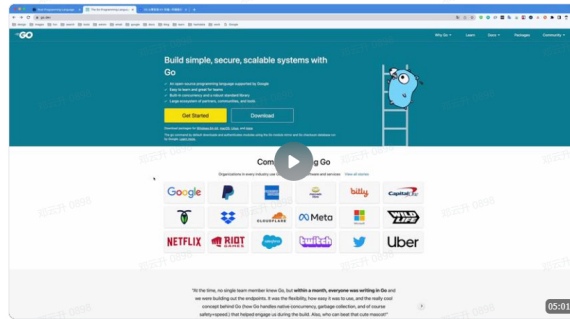


03 从零实现 KV 存储—环境搭建



本课程的 KV 存储项目，将会使用 Go 和 Rust 两种语言实现。你可以选择自己熟悉的语言进行环境的搭建，下面是环境搭建的简要教程。

Go 环境搭建

首先我们需要安装 Go 的运行环境，如果你常用的开发机上已经有了的话，则可以忽略。
可以在 Go 官网下载对应操作系统的安装包: <https://go.dev/dl/>

不同的操作系统当中，Go 语言的安装步骤在官网中已经有详细的描述：

安装文档: <https://go.dev/doc/install>

Documentation > Download and install

Download and install

Download and install Go quickly with the steps described here.

For other content on installing, you might be interested in:

- [Managing Go installations](#) -- How to install multiple versions and uninstall.
- [Installing Go from source](#) -- How to check out the sources, build them on your own machine, and run them.

Download (1.20.1)

Don't see your operating system here? Try one of the [other downloads](#).

Go installation

Select the tab for your computer's operating system below, then follow its installation instructions.

Linux

Mac

Windows

1. Remove any previous Go installation by deleting the `/usr/local/go` folder (if it exists), then extract the archive you just downloaded into `/usr/local`, creating a fresh Go tree in `/usr/local/go`:

```
$ rm -rf /usr/local/go && tar -C /usr/local -xzf go1.20.1.linux-amd64.tar.gz
```

(You may need to run the command as root or through sudo).

Do not untar the archive into an existing `/usr/local/go` tree. This is known to produce broken Go installations.

2. Add `/usr/local/go/bin` to the `PATH` environment variable.

You can do this by adding the following line to your `$HOME/.profile` or `/etc/profile` (for a system-wide installation):

```
export PATH=$PATH:/usr/local/go/bin
```

Note: Changes made to a profile file may not apply until the next time you log into your computer. To apply the changes immediately, just run the shell commands directly or execute them from the profile using a command such as `source $HOME/.profile`.

3. Verify that you've installed Go by opening a command prompt and typing the following command:

```
$ go version
```

4. Confirm that the command prints the installed version of Go.

安装完成后，使用 `go version` 命令，如果能够查看到 Go 的版本信息，则说明安装成功。

安装以后默认的 GOPROXY 是国外的镜像地址，在拉取一些依赖的时候，会比较慢，可以修改为国内的镜像，比如七牛云的，使用如下命令修改：

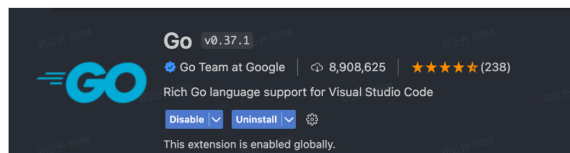
```
1 go env -w GOPROXY="https://goproxy.cn"
```

并且可以将 GOSUMDB 这个变量关闭掉：

```
1 go env -w GOSUMDB="off"
```

对于编辑器的选择，可以按照的你喜欢来，我这里推荐常用的 Goland 或者 VSCode，如果你使用 IDEA 的旗舰版，可以安装 Go 插件，安装后的效果和 Goland 基本上是一样的。

VSCode 用户可以安装 Go 的插件：



Rust 环境搭建

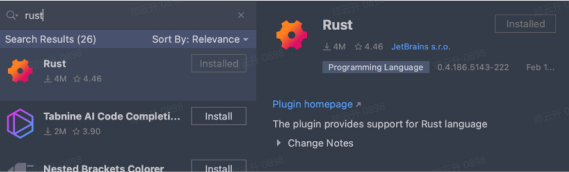
首先需要安装 Rust 的运行环境，可以在官网中查看具体的安装方法：
<https://www.rust-lang.org/tools/install>

这里推荐使用 rustup 的方法，只需要通过以下命令：

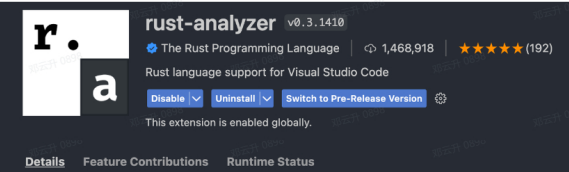
```
1 curl --proto 'https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

然后设置好 cargo 的环境变量，如果能够输出 cargo --version 和 rustc --version，则安装成功。

针对 Rust 的编辑器，我这里推荐 IDEA 或者 VSCode，在 IDEA 中需要安装 rust 的插件：



VSCode 用户可以安装 rust-analyzer 插件：



初始化项目

环境搭建好了之后，可以开始初始化项目了，我们可以在适当的位置，新建一个叫做 kv-projects 的目录，里面会存放我们的 kv 项目，包含 rust 和 go 的。

然后在这个目录当中，先初始化一个 Go 的项目，新建一个 bitcoask-go 的目录，然后进入这个目录，执行

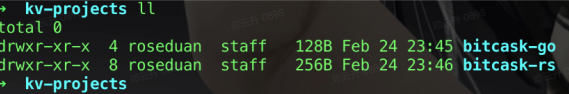
```
1 go mod init bitcoask-go
```

在这个目录中初始化一个 go 项目。

如果是 Rust 语言，则使用以下命令初始化项目：

```
1 cargo new bitcoask-rs
```

最后我们的 kv-projects 目录中的结构就是这样的，包含一个 Go 项目和一个 Rust 项目：



你和其他 4 人已赞