

# 从C++到Rust，用cargo创建package和crate

原创 Ajonbin AJonbin的杂货铺 2023年12月31日 22:03 中国香港

一般来说，我们的应用程序都是有一个可执行程序和他多调用的库组成。

今天来看看怎么用cargo创建你的应用程序。

crate和package了解下

crate是就是rustc的一个编译单元。crate分为两种，一种是bin-二进制可执行文件，另一种就是lib-库。

package是一个或多个crates的组合。

每个package都有一个Cargo.toml文件来描述如何构建这些crates。

一个package里可以有多个二进制可执行crate，但只能有一个库crate。

创建新的package和crate都可以通过cargo命令来实现。

我们先来创建一个包含可执行crate的package

```
$ cargo new hello_bin  
Created binary (application) `hello_bin` package
```

cargo new 默认创建一个可执行的package。也可以加上 --bin 参数显式地创建一个可执行package。--bin是默认的参数。

用tree命令看一下cargo 为我们创建了哪些文件

```
$ tree hello_bin
hello_bin
├── Cargo.lock
├── Cargo.toml
└── src
    └── main.rs
```

1 directories, 3 files

 AJonbin的杂货铺

Cargo.toml定义了如何构建一个package。

cargo为我们创建的toml文件很简单。包含了一个[package]段和[dependencies]段。

[package]定义package的名字，版本。

[dependencies]定义了这个package所依赖的其他库。由于目前我们没有依赖库，所以[dependencies]是空的。

这个Cargo.toml里并没有定义这个package的类型是可执行的还是库。这是因为默认情况下，cargo会根据你的目录结构和源代码来推测package的类型。这个package中，cargo看到src/main.rs文件，就会判定这个是一个可执行的package。

## 再来创建一个lib

用cargo new 创建lib的时候，需要加上--lib参数。

再看看hello\_lib/Cargo.toml文件

这个toml文件和可执行crate的toml基本没什么两样，同理可得，cargo会根据目录结构和文件名来推测package类型。src/lib.rs就表明是个库package。

打开src/lib.rs，可以看到默认定义了一个add函数。Rust的语法细节以后再聊，现在大致猜猜也能知道这是个add函数，然后还有个测试函数。

两个package创建好了，接下来就是如何运行rust 程序了。

### 先单独跑hello\_bin

很简单，运行cargo run 就可以运行package里的可执行crate了。

默认编译运行的是debug版本，目标文件都在新建的target目录里。

接着我们让hello\_bin调用hello\_lib里的add函数

首先我们通过cargo add 命令给hello\_bin增加需要的库依赖

如果我们要添建的库已经发布到crates.io的，那么我们只需要执行 cargo add <name>，就可以添建需要的库了。

比如我们把序列化的库serde添加为依赖，可以这么做

然后来看下Cargo.toml，可以看到在[dependencies]段里新增加了一个serde-1.0.193的依赖。

## 再来添加hello\_lib

由于hello\_lib并没有发布到crates.io，是一个本地的crate，所有在运行cargo add的时候需要通过-path参数指定hello\_lib的路径。

再来看看Cargo.toml有什么变化

可以看到hello\_lib已经加到dependencies里了，和serde相比，它多了路径信息。

然后我们来修改下hello\_bin的main.rs，让它调用hello\_lib的add函数。

我们在hello\_bin/src/main.rs的main函数里增加两行，调用hello\_lib的add函数，再把结果打印出来。

老样子，运行cargo run。

可以看到，cargo首先会先下载依赖的库serde，然后编译serde，hello\_lib和hello\_bin，最后运行hello\_bin，得到结果1+2=3。

默认cargo编译运行的是debug版本，如果要运行release版本，加上--release就可以了。

今天聊了聊package和crate最基本的用法，一个package只包含一个crate。下次聊聊一个package包含多个crate的方法。

