Rust 编程语言入门

杨旭，微软 MVP
Rust、Go 开发者

14.4 发布 crate 到 crates.io（2）

创建并设置 Crates.io 账号

- 发布 crate 前，需要在 crates.io 创建账号并获得 API token
- 运行命令：cargo login [你的API token]
  - 通知 cargo，你的 API token 存储在本地 ~/.cargo/credentials
- API token 可以在 https://crates.io/ 进行撤销

为新的 crate 添加元数据

- 在发布 crate 之前，需要在 Cargo.toml 的 [package] 区域为 crate 添加一些元数据：
  - crate 需要唯一的名称：name
  - description：一两句话即可，会出现在 crate 搜索的结果里
  - license：需提供许可证标识值（可到 http://spdx.org/licenses/ 查找）
    - 可指定多个 license：用 OR
  - version
  - author
- 发布：cargo publish 命令
- （例子）

发布到 Crates.io

- crate 一旦发布，就是永久性的：该版本无法覆盖，代码无法删除
  - 目的：依赖于该版本的项目可继续正常工作
- （例子）

发布已存在 crate 的新版本

- 修改 crate 后，需要先修改 Cargo.toml 里面的 version 值，再进行重新发布
- 参照 http://semver.org/ 来使用你的语义版本
- 再执行 cargo publish 进行发布

使用 cargo yank 从 Crates.io 撤回版本

- 不可以删除 crate 之前的版本
- 但可以防止其它项目把它作为新的依赖：yank（撤回）一个 crate 版本
  - 防止新项目依赖于该版本
  - 已经存在项目可继续将其作为依赖（并可下载）
- yank 意味着：
  - 所有已经产生 Cargo.lock 的项目都不会中断
  - 任何将来生成的 Cargo.lock 文件都不会使用被 yank 的版本。
- 命令：
  - yank 一个版本（不会删除任何代码）：cargo yank --vers 1.0.1
  - 取消 yank：cargo yank --vers 1.0.1 --undo

再见