

如何快速掌握新的 crates?

- 快速掌握新 crates 的方法
- Rust 值得关注的 crates
- Live coding

快速掌握新 crates 的方法

如何判断一个 `crate` 的质量？

- crates.io 下载量 / dependents 数量 / github star
- 更新频率
- 其 dependency 的质量（是否较新）
 - 比如 crate 还依赖 `futures < 0.3` / `tokio < 1` 不值得尝试
- README / docs.rs / 测试 / 示例代码 / 源代码

快速尝试 crates 的方法

- 带着问题
- 快速阅读文档 (why, what, how)
- 了解相关的知识 (包括 ecosystem)
- clone repo
- 尝试 example (并用自己的方式实现某些 example)
- 阅读源码

示例：polars



```
[{
  name: "Tyr",
  age: 18,
  lang: ["elixir", "rust"]
}, {
  name: "Lindsey",
  age: 8,
  lang: ["scratch"]
}]
```



示例：headless_chrome

```
let options = LaunchOptions::default_builder()
    .build()
    .expect("Couldn't find appropriate Chrome binary.");
let browser = Browser::new(options)?;
let tab = browser.wait_for_initial_tab()?;
let jpeg_data = tab
    .navigate_to("https://www.wikipedia.org")?
    .wait_until_navigated()?
    .capture_screenshot(JPEG(Some(75)), None, true)?;
fs::write("screenshot.jpg", &jpeg_data)?;
```



比较 polars 和 headless_chrome

- 大概代表了我们会使用的 crates 的两级
- 代码量：polars (50k vs 8k)
- 学习曲线：polars (背景知识多)
- 文档质量：都一般
- 容易上手：headless_chrome

Rust 生态中值得关注的 crates

极简版，且不包含 `are we xxx yet` 页面提及的 crates

FFI 相关

- C++: [autocxx](#)
- Elixir: [Rustler](#)
- Python: [PyO3](#)
- Nodejs: [neon](#)
- Swift: [cbingen](#)
- Java: [jni-rs](#), [flapigen-rs](#), [robusta](#)

macro 相关

- [derive_more](#)
- [syn](#)
- [quote](#)
- [cargo-expand](#)
- [proc-macro-workshop](#)
- [seq-macro](#)
- [paste](#)

并发处理

- [parking_lot](#)
- [crossbeam](#)
- [dashmap](#)
- [flume](#)
- [rayon](#)
- [rxRust](#)

parser

- [nom](#)
- [pest](#)
- [pom](#)

日志，追踪

- tracing

其它

- [bytes](#)
- [serde](#)
- [itertools](#)
- [oso](#)
- [time](#)
- [tauri](#)

Live coding: 构建一个命令行工具

- 工具: clap, rust-headless-chrome, qrcode-rust
- 思路:
 - clap 提供基本命令行 `web2image <url> --output <path:/tmp/screenshot.jpg>`
 - 使用 `headless_chrome` 抓取 url, 存储为图片
 - 使用 `qrcode-rust` 对 URL 生成 QR code, 添加到图片中

Learning

- headless_chrome 质量一般 (capture snapshot!!!)
- 你遇到的大部分问题，别人也遇到了，善用搜索
- 速度比想象慢，release mode 速度好很多
- 最后总结：
 - <https://github.com/atroche/rust-headless-chrome/issues/227>
 - <https://github.com/atroche/rust-headless-chrome/pull/233/files>

```
thread 'main' panicked at 'called `Result::unwrap()` on an `Err` value: Error("unknown variant `marked`, expected one of `ready`, `loading`, `failed`')', src/lib.rs:10:10
note: run with `RUST_BACKTRACE=1` environment variable to display a backtrace
```


参考资料

- [polars](#)
- [headless_chrome](#)
- [fantoccini](#)
- [qrcode-rust](#)
- [clap](#)