为 Rust 工具 mdbook 实现 PDF 格式输出功能

> **为 Rust 工具 mdbook 实现 PDF 格式输出功能 · Issue #I4QM4V · openEuler/opensource-…**
>
> < 选择类别：开源实习 < 选择标签：internOSCourse 为 Rust 工具 mdbook 实现 PDF 格式输出功能【任务分值】50 分【背景描述】mdbookhttps://ru...
>
> https://gitee.com/openeuler/opensource-intern/issues/I4QM4V

版本目标 0.1.0

为 mdbook 提供 PDF 输出功能，在执行 mdbook build 的命令时，根据配置项在 book 目录下生成 PDF 文件。

项目背景

目前mdbook仅支持将书籍生成为html页面，不支持对应的pdf输出。

项目需求

1. 完整功能实现，包含一定的测试用例
2. 选择的依赖是通用的
3. 在 mac（m1）、Linux 大多数发型版和 windows 上都可以跑

开发环境要求

1. 在 X86_64 和 ARM64 架构下运行
2. Rust Edition 使用 2021

输出功能

输出标准的 PDF 文件，支持中文，支持自定义PDF纸张方向、页面缩放比例、纸张宽度和高度、页面边距、生成的PDF页面范围、是否显示页眉和页脚以及自定义其格式。

技术方案

简介

mdBook 支持使用自定义后端，当调用 mdbook build 命令时，假如书籍目录中的 `book.toml` 除了默认存在的 [output.html] 生成html网页端外，还存在 [output.pdf] 项，则会在系统PATH中调用 `mdbook-pdf`，同时该程序标准输入中传入相关书籍信息以及参数配置，具体细节请查看 mdBook 的说明文档 Alternative Backends - mdBook Documentation。

本程序使用Rust编写，根据上述原理基于headless chrome和Chrome开发工具协议，调用 Google Chrome / Chromium / Microsoft Edge 浏览器打开mdBook后端生成的网站中的 print.html 生成

PDF，由于是 headless 模式，运行时浏览器都在后台进行相应操作，无需图形界面，即使有图形界面也不会看到浏览器界面，无感知。

目前测试了一下 Firefox 对 Chrome开发工具协议 打印生成PDF部分似乎不支持，Safari 浏览器对 Webdriver 打印生成PDF部分都不支持，因而遑论对 Chrome开发工具协议 打印生成PDF部分的支持了，而对 Chrome / Chromium / Microsoft Edge 浏览器的最新版经测试都是支持的。

用户调用 mdbook build 命令时相关生成输出如下：

```
2022-01-22 03:21:43 [INFO] (mdbook::book): Book building has started
2022-01-22 03:21:43 [INFO] (mdbook::book): Running the html backend
2022-01-22 03:21:45 [INFO] (mdbook::book): Running the pdf backend
2022-01-22 03:21:45 [INFO] (mdbook::renderer): Invoking the "pdf" renderer
Generating PDF, please be patient...
PDF successfully generated at: /Users/runner/work/mdbook-pdf/mdbook-pdf/rust-by-example/book/pdf/output.pdf
```

因为headless chrome上游现在自动下载 Chromium 功能还不可用，因而现在某些情况下需要手动下载相关浏览器，即可正常使用。待后续如果上游修复了自动下载 Chromium 功能，将会同步更新，到时候用户将不用管任何东西直接运行即可。

- 在Windows 10及以上该程序无需安装任何额外软件即可正常生成 PDF，因为 Microsoft Edge 是 Windows 系统自带的浏览器。当然如果考虑到对没有自带安装 Edge 的老版本Windows的支持，在电脑上安装一个 Google Chrome 即可。
- 在 macOS 中需要下载并安装 Google Chrome。
- 在 Linux 中安装Google Chrome / Chromium / Microsoft Edge 浏览器中的任意一个即可，推荐安装 Chromium，该软件包在您的发行版中一般名称为 `chromium` 或 `chromium-browser`，可参见 https://pkgs.org/download/chromium https://pkgs.org/download/chromium-browser 查看Linux发行版的支持情况（注意，在 Ubuntu 18.04 之后需要通过 `snap` 安装 `chromium-browser`）。

具体使用方法请参见 README：

> **mdbook-pdf/README.md · Hollow Man/opensource-intern - Gitee.com**
>
> 在 openEuler 实习项目中，涉及到操作系统教学相关的任务发布和相关代码中本仓库中集中管理和保存
>
> https://gitee.com/jiangsonglin2/opensource-intern/blob/master/mdbook-pdf/README.md

技术细节

放弃了之前的WebDriver方案，虽然WebDriver方案增加了对 Firefox 的支持，但是由于一般WebDriver和浏览器版本之间是相互对应的，用户需要手动查询浏览器版本，下载对应的WebDriver，不如直接安装浏览器来的方便快捷。

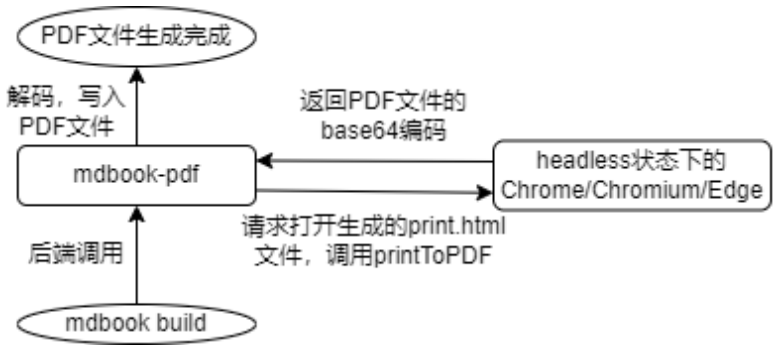
而且使用WebDriver打印生成PDF文件时存在一个重大的缺陷，遇到稍微大一点规模的文档时会由于页面崩溃导致生成失败（Chrome 和 Firefox 对应的WebDriver都同时存在这个问题）：

Bash

```
selenium.common.exceptions.WebDriverException: Message: unknown error: session deleted because of page crash
from unknown error: cannot determine loading status
from tab crashed
```

此外 WebDriver提供的可自定义打印 PDF 参数选项也相较而言不够多，所以最终决定抛弃 WebDriver，使用 Chrome开发工具协议 直接和 headless 状态下基于 Chromium 的浏览器直接交互。

架构简图如下：



Rust Crate headless chrome 是Chrome开发工具团队维护的Node JS 库 Puppeteer 的 纯 Rust 实现，但是对于生成大的 PDF 文件的稳定性还不够，容易报超时错误，因而我通过 fork 该 Crate 仓库打补丁的方式，取消了超时限制：https://github.com/HollowMan6/rust-headless-chrome/commit/0c5d707b28bdc577b5d386bc2e90d91b10143116， 同时增加了对调用Edge生成的支持： https://github.com/HollowMan6/rust-headless-chrome/commit/c4517ef721f7f61e88a49f0005a729e792428fa0

在补丁之后使用该 Crate 都能十分顺利地生成 PDF，且在同样的测试环境下也没遇到 WebDriver 下类似的崩溃错误。

目前该 mdbook-pdf 程序还未上架 crates.io，因为这里使用了fork仓库中的依赖，而并非源于 crates.io，所以离上架还遥遥无期，我已经提了相关的issue和pr，希望能够在解决之后上游能release出新版本同步到crates.io。

Request to add an option to disable all the wait timeout · Issue #287 · atroche/rust-…

I suggest adding an option to disable all the timeout for places that use Wait::with_timeout(x).until so that they will wait forever to increase the robustness of the program. In my case, errors re...

https://github.com/atroche/rust-headless-chrome/issues/287

github.com

https://github.com/atroche/rust-headless-chrome/pull/288

因而目前安装该程序的方式只能是通过仓库编译 cargo build --release 进行生成二进制文件使用，而不能直接使用 cargo install 的形式安装。对于x86_64 的 macOS、Linux 和 Windows，以及arm64 版本的 Linux 可以通过 https://github.com/HollowMan6/mdbook-pdf/actions/runs/1737778711 该工作流页面下方的 Artifacts 部分下载二进制文件直接使用。

测试方案

mdbook-pdf/test.yml at main · HollowMan6/mdbook-pdf

A backend for mdbook in Python for generating PDF based on Chrome DevTools Protocol. - mdbook-pdf/test.yml at main · HollowMan6/mdbook-pdf

https://github.com/HollowMan6/mdbook-pdf/blob/main/.github/workflows/test.yml

使用GitHub Actions进行CI/CD，以rust相关书籍来进行测试，以能否成功生成PDF文档为判断标准。

cargo/src/doc at master · rust-lang/cargo

The Rust package manager. Contribute to rust-lang/cargo development by creating an account on GitHub.

https://github.com/rust-lang/cargo/tree/master/src/doc

github.com

https://github.com/rust-lang-nursery/edition-guide

github.com

https://github.com/rust-embedded/book

github.com

https://github.com/rust-lang/mdBook/tree/master/guide

github.com

https://github.com/rust-lang/mdBook/tree/master/test_book

github.com

https://github.com/rust-lang/reference

github.com

https://github.com/rust-lang/rust-by-example

GitHub - rust-lang/book: The Rust Programming Language

The Rust Programming Language. Contribute to rust-lang/book development by creating an account on GitHub.

https://github.com/rust-lang/book

github.com

https://github.com/rust-lang/rustc-dev-guide

github.com

https://github.com/rust-lang/rust/tree/master/src/doc/rustdoc

GitHub - rust-lang/nomicon: The Dark Arts of Advanced and Unsafe Rust Programming

The Dark Arts of Advanced and Unsafe Rust Programming - GitHub - rust-lang/nomicon: The Dark Arts of Advanced and Unsafe Rust Programming

https://github.com/rust-lang/nomicon

目前所有上述书籍的PDF版本文档在x86_64 的 Linux，Windows，macOS 即使是无桌面环境下都可以正常生成，可点击此链接中的 Artifacts 部分查看测试Workflow生成的PDF文档。

开发资料

Rust 2021 - The Edition Guide

Introduction 1. What are editions? 1.1. Creating a new project 1.2. Transitioning an existing project to a new edition 1.3. Advanced migrations 2. Rust 2015 3. Rust 2018 3.1. Path and module system ch

https://doc.rust-lang.org/edition-guide/rust-2021/index.html

Alternative Backends - mdBook Documentation

Create book from markdown files. Like Gitbook but implemented in Rust

https://rust-lang.github.io/mdBook/for_developers/backends.html

github.com

https://github.com/atroche/rust-headless-chrome

chromedevtools.github.io

https://chromedevtools.github.io/devtools-protocol/tot/Page/#method-printToPDF