

ExtendedSearch



一个功能强大的文件搜索和管理工具

rust 1.70+

slint 1.12.0

license MIT

platform Windows | Linux | macOS

✨ 特性

🔍 高级搜索功能

- **多目录搜索:** 同时在多个目录中搜索文件
- **正则表达式支持:** 支持文件名和完整路径的正则表达式匹配
- **文件类型过滤:** 根据文件属性进行精确筛选
- **大小范围过滤:** 按文件大小范围搜索
- **日期范围过滤:** 支持多种日期筛选模式（天、周、月、年、特定日期范围）
- **隐藏文件搜索:** 可选择是否搜索隐藏文件和文件夹
- **只读文件过滤:** 可选择是否包含只读文件

📁 文件管理功能

- **批量选择:** 支持单选、多选和全选文件
- **文件删除:** 安全删除选中的文件
- **文件映射:** 将搜索结果映射到指定目录结构
- **去重功能:** 基于文件哈希值自动去除重复文件
- **哈希计算:** 可选择计算并记录文件 SHA-256 哈希值

数据管理

- **结果导出:** 将搜索结果导出为 JSON 格式
- **结果导入:** 从 JSON 文件导入之前的搜索结果
- **持久化设置:** 自动保存和恢复过滤器设置

用户界面

- **现代化界面:** 基于 Slint 的响应式用户界面
- **直观操作:** 简洁的侧边栏和主要内容区域
- **实时反馈:** 悬停效果和状态提示
- **文件预览:** 显示文件详细信息（路径、大小、修改时间、哈希值）

快速开始

系统要求

- **操作系统:** Windows 10+, Linux (任何现代发行版), macOS 10.15+
- **Rust:** 1.70 或更高版本

安装方法

从源码构建

1. 克隆仓库

```
git clone https://github.com/your-username/ExtendedSearch.git
cd ExtendedSearch
```

2. 安装 Rust（如果尚未安装）

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
source ~/.cargo/env
```

3. 构建项目

```
cargo build --release
```

4. 运行应用

```
cargo run --release
```

预编译二进制文件

前往 [Releases](#) 页面下载适合您操作系统的预编译版本。



使用指南

基本搜索

1. 添加搜索目录

- 点击左侧边栏的"添加目录"按钮
- 选择要搜索的文件夹
- 勾选目录以包含在搜索中

2. 配置过滤器

- 点击"过滤设置"选项卡
- 设置文件大小范围、日期范围等条件
- 可选择是否计算文件哈希值

3. 开始搜索

- 切换到"搜索"选项卡
- 点击"开始搜索"按钮
- 查看搜索结果列表

高级功能

正则表达式搜索

```
# 搜索所有 .txt 文件
.*\.txt$
```

```
# 搜索以 "log" 开头的文件
^log.*
```

```
# 搜索包含日期格式的文件名
\d{4}-\d{2}-\d{2}
```

文件管理操作

- **选择文件:** 点击复选框选择单个或多个文件
- **全选/取消全选:** 使用顶部的全选按钮
- **删除文件:** 选择文件后点击"删除选中文件"
- **文件映射:** 指定源文件夹后, 点击"映射"将文件复制到新位置

数据导入导出

- **导出:** 点击"导出结果"将当前搜索结果保存为 JSON
- **导入:** 点击"导入结果"加载之前保存的搜索结果

项目结构

```
ExtendedSearch/
├── src/
│   ├── main.rs          # 应用程序入口点
│   ├── ui_handler.rs    # UI 事件处理和回调
│   ├── filter.rs        # 搜索过滤器逻辑
│   ├── helper.rs        # 搜索和文件操作工具
│   └── search_file/     # 文件搜索模块
│       ├── mod.rs
│       ├── build_tree.rs # 目录树构建
│       └── settings.rs  # 搜索设置
├── ui/
│   ├── app-window.slint # 主窗口布局
│   └── components/      # UI 组件
│       ├── content-area.slint
│       ├── custom-button.slint
│       ├── filter-settings-optimized.slint
│       ├── scope-selection.slint
│       ├── search-page.slint
│       ├── search-result-view.slint
│       └── sidebar.slint
├── assets/
│   ├── fonts/           # 字体文件
│   └── icons/           # 应用图标
├── Cargo.toml           # 项目依赖配置
└── README.md            # 项目说明文档
```

技术栈

- 核心语言: [Rust](#) - 系统级编程语言，保证性能和安全性
- UI 框架: [Slint](#) - 现代化的 Rust GUI 工具包
- 文件对话框: [native-dialog](#) - 原生系统对话框
- 正则表达式: [regex](#) - 高性能正则表达式引擎
- 日期时间: [chrono](#) - 日期和时间处理
- 哈希计算: [sha2](#) - SHA-256 哈希算法
- 序列化: [serde](#) + [serde_json](#) - 数据序列化
- 文件删除: [trash](#) - 安全文件删除（移到回收站）

贡献指南

欢迎贡献代码！请遵循以下步骤：

1. **Fork 此仓库**
2. **创建功能分支** (`git checkout -b feature/AmazingFeature`)
3. **提交更改** (`git commit -m 'Add some AmazingFeature'`)
4. **推送到分支** (`git push origin feature/AmazingFeature`)
5. **开启 Pull Request**

开发环境设置

```
# 克隆仓库
git clone https://github.com/your-username/ExtendedSearch.git
cd ExtendedSearch
```

```
# 安装开发依赖
cargo install cargo-watch
```

```
# 开发模式运行（自动重载）
cargo watch -x "run"
```

```
# 运行测试
cargo test
```

```
# 代码格式化
cargo fmt
```

```
# 代码检查
cargo clippy
```

更新日志

v0.1.0 (2025-01-XX)

- 🎉 初始版本发布
- ✅ 基本文件搜索功能
- ✅ 高级过滤器支持
- ✅ 文件管理操作
- ✅ 数据导入导出
- ✅ 现代化用户界面

许可证


本项目基于 [MIT License](#) 许可证开源。

致谢

- [Slint UI](#) - 提供优秀的 Rust GUI 框架
- [Rust 社区](#) - 提供丰富的生态系统
- 所有贡献者和用户的支持

联系方式

- 问题反馈: [GitHub Issues](#)
- 功能建议: [GitHub Discussions](#)
- 邮箱: your-email@example.com

如果这个项目对您有帮助，请考虑给它一个  ！