

Rust BuildOn #2

Jim Fawcett

<https://JimFawcett.github.io>

February 26, 2021

Build-On

- Learn Rust in small steps
- Start with a simple Rust demo
 - Text Finder
- Build on that in interesting ways, e.g., for Text Finder:
 - Add capability – command line parsing, directory tree walking, ...
 - Use generics – plugin components
 - Use threads – parallel text searches, thread pool based dir traversal
 - Use library components
- Discord server
 - Q&A
 - Comments

How would I use Rust at work?

- Migrate parts of a large project to Rust using its foreign function interface:
 - Piece-meal improvement: performance, safety of existing code-base
 - <https://locka99.gitbooks.io/a-guide-to-porting-c-to-rust/content/>
 - [std::ffi](#)
- Use Rust generated WebAssembly to speed JavaScript processing
 - <https://rustwasm.github.io/book/>
- Start new projects in Rust
 - Create fast, sound code with lots of help from compiler messages
 - Great tooling: [RustBites Tooling](#), works on Windows, Linux, macOS
 - Resources: e-books, active user forum, <https://crates.io>

BuildOn Timeline: Bi-Weekly Sessions

- Session #1 – BuildOn Introduction & Rust Preview
- Session #2 – Project Introduction & Some Rust details
- Session #2 – **Step #1**: TextSearch -> Functional package with demo
- Session #3 – **Step #2**: DirNav -> “
- Session #4 – **Step #3**: CmdInParser -> “
- Session #5 – **Step #4**: Executive, Display -> Functional project
- Session #6 – RegEx Extension ?
- Session #7 – Parallel Text Search ?

BuildOn Session #2 Topics

Focus: TextSearch Package

- Step #0
 - Run TextFinder
 - [DesignBites Intro](#)
 - TestCargo
- Step #1 - TextSearch
 - [Starter Code](#)
 - [Tooling](#)
 - [Functions, Structs](#)
 - [RustStory Libraries.html#fs](#)
 - [std:fs](#)
- BuildOn

Focus: Rust

- [RustBite Data](#)
- [Safety](#)
- [Anim Features](#)
- [Rust Bites](#)
- [Rust Story](#)
- [Code Experiments](#)
- **Questions?**

Building Rust Library Package (Crate)

Windows cmd (terminal)

```
c:\su\temp> cargo new TestCargo --lib --name test_cargo
Created library `test_cargo` package

c:\su\temp> cd TestCargo

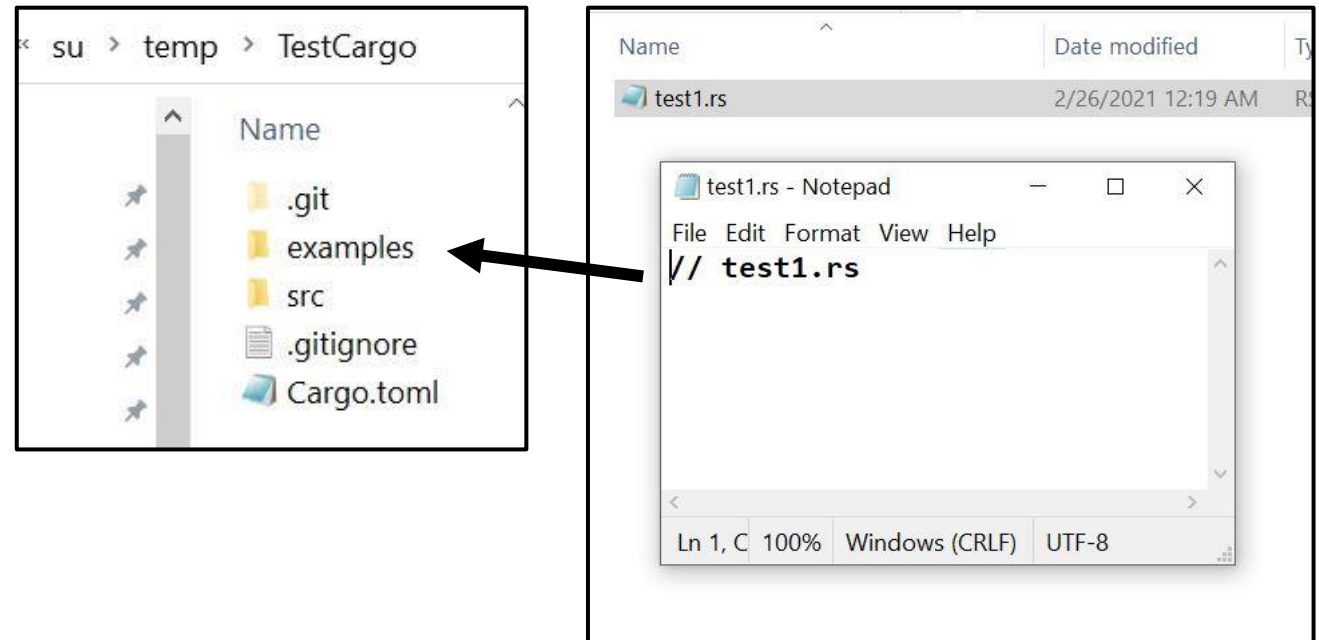
c:\su\temp\TestCargo> mkdir examples

c:\su\temp\TestCargo> cd examples

c:\su\temp\TestCargo\examples> echo // test1.rs >test1.rs

c:\su\temp\TestCargo\examples>
```

Windows Explorer File Manager



lib.rs - TestCargo - Visual Studio Code

File Edit Selection View Go Run Terminal Help

lib.rs test1.rs launch.json Cargo.toml

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 3: run test1

EXPLORER

- OPEN EDITORS
 - GROUP 1
 - lib.rs src U
 - test1.rs examples U
 - launch.json .vscode U
 - Cargo.toml U
 - GROUP 2
 - test1.rs examples U
 - lib.rs src U
- TESTCARGO
 - .vscode
 - launch.json U
 - examples
 - test1.rs U
 - src
 - lib.rs U
 - target
 - .gitignore U
 - Cargo.lock
 - Cargo.toml U

OUTLINE TIMELINE NPM SCRIPTS

```
examples > test1.rs > main
1 // test1.rs
2
3 use test_cargo::{TestRust};
4
5 ▶ Run | Debug
6 fn main() {
7     let tr: TestRust = TestRust::new();
8     tr.say_hello(name: "Jim");
9
10    print!("\n\n That's all Folks!\n\n");
}
```

```
src > lib.rs > ...
1 // lib.rs
2
3 3 implementations
4 #[derive(Debug, Clone)]
5 pub struct TestRust;
6 impl TestRust {
7     pub fn new() -> TestRust {
8         TestRust {}
9     }
10    pub fn say_hello(&self, name: &str) {
11        print!("\n Hello {:?}, welcome to BuildOn", name);
12    }
13
14    ▶ Run Test | Debug
15    #[cfg(test)]
16    mod tests {
17        ▶ Run Test | Debug
18        #[test]
19        fn it_works() {
20            assert_eq!(2 + 2, 4);
21        }
22    }
23 }
```

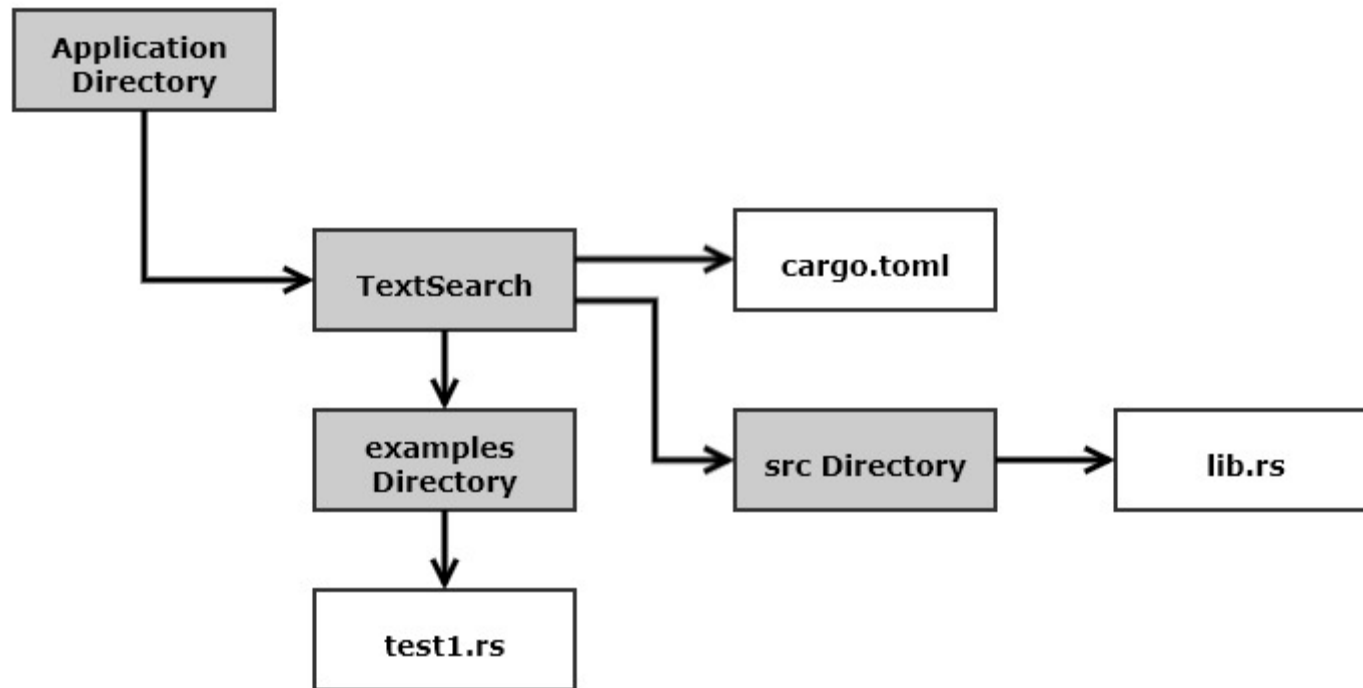
```
C:\su\temp\TestCargo>
cargo run --example test1
Compiling test_cargo v0.1.0 (C:\su\temp\TestCargo)
Finished dev [unoptimized + debuginfo] target(s) in 0.63s
Running `target\debug\examples\test1.exe`

Hello "Jim", welcome to BuildOn

That's all Folks!

C:\su\temp\TestCargo>
```

Rust Project Structure for TextSearch



BuildOn Session #2 Topics

Focus: TextSearch Package

- [Step #0](#)
 - Run TextFinder
 - [DesignBites Intro](#)
 - TestCargo
- [Step #1](#) - TextSearch
 - [Starter Code](#)
 - [Tooling](#)
 - [Functions, Structs](#)
 - [RustStory Libraries.html#fs](#)
 - [std:fs](#)
- [BuildOn](#)

Focus: Rust

- [RustBite Data](#)
- [Safety](#)
- [Anim Features](#)
- [Rust Bites](#)
- [Rust Story](#)
- [Code Experiments](#)
- **Questions?**

Next BuildOn – Session #3 – DirNav

Project Packages

- [Step #1](#) – Quick Review
 - Text Search package
 - Posted solution
- [Step #2](#) - DirNav
 - DirNav package spec
 - Look at Starter Code

Rust

- [Enumerations](#)
- [Error Handling](#)
- [Conversions](#)
- [Generics & Traits](#)
- Answer Questions

BuildOn

- Session #3
 - DirNav
- Session #4
 - CmdInParser
- Session #5
 - Executive & Display
- Session #6 ?
 - RegEx Search
- Session #7 ?
 - Parallel Text Search

That's All Folks!