

Nov 07, 19 11:37

helper.rs

Page 1/1

```
1 use std::error::Error;
2 use std::process::exit;
3
4 use crate::State;
5
6 /// Runs the given closure and exits with a State::Critical after printing out
7 /// the error message if the Result contains an Err.
8 pub fn safe_run<F: Fn() -> Result<(), Box<dyn Error>>>(closure: F) {
9     safe_run_with_state(closure, State::Critical);
10 }
11
12 /// Runs the given closure and exits with the given State after printing out
13 /// the error message if the Result contains an Err.
14 pub fn safe_run_with_state<F: Fn() -> Result<(), Box<dyn Error>>>(closure: F, error_state: State) {
15     if let Err(e) = closure() {
16         println!("CRITICAL: {}", e.to_string());
17         exit(error_state.exit_code());
18     }
19 }
```