

EXPLORER

dispatch.rs

OPEN EDITORS

- dispatch.rs rls/src/server M

RLS

- ci
- rls
  - src
    - actions
    - build
    - server
      - dispatch.rs M
      - io.rs
      - message.rs
      - mod.rs
      - cmd.rs
      - concurrency.rs
      - config.rs
      - lib.rs
      - lsp\_data.rs
      - main.rs
      - project\_model.rs
  - rls-analysis
  - rls-data
  - rls-ipc
  - rls-rustc
  - rls-span
  - rls-vfs
  - target
  - tests
- .gitignore
- .travis.yml
- architecture.md
- azure-pipelines.yml
- build.rs
- Cargo.lock
- Cargo.toml 1
- changelog.md
- clients.md
- contributing.md
- COPYRIGHT
- debugging.md
- LICENSE-APACHE
- LICENSE-MIT

OUTLINE

```

pub(crate) fn new<O: Output>(out: O) -> Self {
    let (sender, receiver) = mpsc::channel::<(DispatchRequest, InitActionContext, JobToken)>();

    thread::Builder::new()
        .name("dispatch-worker".into())
        .spawn(move || {
            while let Ok((request, ctx, token)) = receiver.recv() {
                request.handle(ctx, &out);
                drop(token);
            }
        })
        .unwrap();

    Self { sender }
}

/// Sends a request to the dispatch-worker thread; does not block.
pub(crate) fn dispatch<R: Into<DispatchRequest>>>(
    &mut self,
    request: R,
    ctx: InitActionContext,
) {
    let (job, token) = ConcurrentJob::new();
    ctx.add_job(job);
    if let Err(err) = self.sender.send((request.into(), ctx, token)) {
        debug!("failed to dispatch request: {:?}", err);
    }
}

/// Stdin-non-blocking request logic designed to be packed into a `DispatchRequest`

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

1: zsh

0.07s sainnhe@surface

```

~/repo/test/rls
> tpal minimat

```

|     | blk | red | grn | yel | blu | mag | cya | wht | def |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| blk | 🟩   | 🟥   | 🟦   | 🟨   | 🟩   | 🟪   | 🟩   | 🟨   | 🟩   |
| red | 🟥   | 🟥   | 🟦   | 🟨   | 🟩   | 🟪   | 🟩   | 🟨   | 🟩   |
| grn | 🟦   | 🟥   | 🟦   | 🟨   | 🟩   | 🟪   | 🟩   | 🟨   | 🟩   |
| yel | 🟨   | 🟥   | 🟦   | 🟨   | 🟩   | 🟪   | 🟩   | 🟨   | 🟩   |
| blu | 🟩   | 🟥   | 🟦   | 🟨   | 🟩   | 🟪   | 🟩   | 🟨   | 🟩   |
| mag | 🟪   | 🟥   | 🟦   | 🟨   | 🟩   | 🟪   | 🟩   | 🟨   | 🟩   |
| cya | 🟩   | 🟥   | 🟦   | 🟨   | 🟩   | 🟪   | 🟩   | 🟨   | 🟩   |
| wht | 🟨   | 🟥   | 🟦   | 🟨   | 🟩   | 🟪   | 🟩   | 🟨   | 🟩   |
| def | 🟩   | 🟥   | 🟦   | 🟨   | 🟩   | 🟪   | 🟩   | 🟨   | 🟩   |