## extern-location

MCP for this feature: #303

---

The `unused-extern-crates` lint reports when a crate was specified on the rustc command-line with `--extern name=path` but no symbols were referenced in it. This is useful to know, but it's hard to map that back to a specific place a user or tool could fix (ie, to remove the unused dependency).

The `--extern-location` flag allows the build system to associate a location with the `--extern` option, which is then emitted as part of the diagnostics. This location is abstract and just round-tripped through rustc; the compiler never attempts to interpret it in any way.

There are two supported forms of location: a bare string, or a blob of json: - `--extern-location foo=raw:Makefile:123` would associate the raw string `Makefile:123` - `--extern-location 'bar=json:{"target":"//my_project:library","dep":"//common:s` would associate the json structure with `--extern bar=<path>`, indicating which dependency of which rule introduced the unused extern crate.

This primarily intended to be used with tooling - for example a linter which can automatically remove unused dependencies - rather than being directly presented to users.

`raw` locations are presented as part of the normal rendered diagnostics and included in the json form. `json` locations are only included in the json form of diagnostics, as a `tool_metadata` field. For `raw` locations `tool_metadata` is simply a json string, whereas `json` allows the rustc invoker to fully control its form and content.