»domain«
A toolbox for developing DNS in Rust.
How it started
The Idea
So, how is it going?
- **zonefile**: reading and writing of zonefiles. This feature enables the `zonefile` module and currently also enables the `bytes` and `std` features.

---

## Modules

- **base**: Handling of DNS data.
- **dep**: Re-exports of dependencies
- **rdata**: Record Data of Well-defined Record Types
- **resolv** (`resolv`): Asynchronous DNS resolving.
- **sign** (`sign`): DNSSEC signing.
- **tsig** (`tsig`): Support for TSIG.
- **utils**: Various utility modules.
- **validate** (`validate`): DNSSEC validation.
- **zonefile** (`zonefile`): Reading and writing of zonefiles.
Oh, no! Distractions!
Where is it going next?
This year for »domain«

Photo by Daniel J. Schwarz on Unsplash
Client Track

• basic transports (Do53, DoT)
• response caching
• DNSSEC validation
Server Track

- basic transports (Do53, DoT)
- zone handling
  - zonefile parsing, zone tree, answer queries
- DNSSEC signing
- zone transfer (AXFR, IXFR)
Bonus Track

• Flexible DNS proxy
  • choose an action based on properties of the request
  • send to upstream server, reject, answer locally, …
Bonus Track II

• DNS diagnostics tool
  • more than just query a server
  • check DNSSEC for a zone
  • compare resolver with authoritative, etc.

• Zone management tools
  • check zone, sign zone, etc.
Thank you!

- https://github.com/nlnetlabs/domain
- labs@nlnetlabs.nl
- @nlnetlabs@fosstodon.org