

Background

This thesis illustrates an approach of implementing a language server for the Nickel language which communicates with its clients, i.e. editors, over the open Language Server Protocol (in the following abbreviated as *LSP*). The current chapter provides the backgrounds on the technological details of the project. As the work presented aims to be transferable to other languages using the same methods, this chapter will provide the means to distinguish the nickel specific implementation details.

The primary technology built upon in this thesis is the language server protocol. The first part of this chapter introduces the LSP, its rationale and improvements over classical approaches, technical capabilities and protocol details. The second part is dedicated to Nickel, elaborating on the context and use-cases of the language followed by an inspection of the technical features Nickel is based on.

Language Server Protocol

Language servers are today's standard of integrating support for programming languages into code editors. Initially developed by Microsoft for the use with their polyglot editor Visual Studio Code^[<https://code.visualstudio.com/>] before being released to the public in 2016 by Microsoft, RedHat and Codeenvy, the LSP decouples language analysis and provision of IDE-like features from the environment used to write. Developed under open source license on GitHub¹, it allows developers of editors and languages to work independently on the support for new languages. If supported by both server and client, the LSP now supports more than 24 language features^[<https://microsoft.github.io/language-server-protocol/specifications/specification-current/>] including code completion, hover information, resolution of type and variable definitions, controlling document highlighting, formatting and more.

Rationale

Since its release, the LSP has grown to be supported by a multitude of languages and editors^{[[@langservers](#) [[@lsp-website](#)]]}.

Commands and Notifications

File Notification

Diagnostics

Hover

Completion

¹<https://github.com/microsoft/language-server-protocol/>

Go-To-*

Symbols

code lenses

Shortcomings

Infrastructure as Code

Software defined Networks

Data oriented languages

Nickel

Gradual typing

Row types

Contracts