企画番号: 2025-03

既存の設定用マークアップ言語に追記して使用できる型定義言語 TYML 活動報告書

代表・メンバー 阿部 誠大 Y200099
アドバイザー教員 芝公仁

1. 目的

世の中には多くの設定用言語が存在する.代表例としては JSON, TOML, INI 等が挙げられる.
しかし,普及している言語の殆どは静的な型を持っておらず,値が正しいかどうかは実際にアプリケーションを動かすまでわからないことが多い.そこで,既存の設定用言語に対して静的な型を提供することで,この問題を解決するための専用言語を開発した.また,このツールを開発することで設定に型の概念を持ち込む文化を育む一助になれば,なお良いと考えている.

2. 計画と開発

2-1. 基本機能の開発

まず,最初に言語のイメージを設計するところから始めた.ほとんどの設定用言語にはコメント機能があり,これに意味をもたせるアイデアを思いついた.コメントを書くことで型定義ファイルを指定させ,型チェック時にこれを参照して動作する方式を採用した.

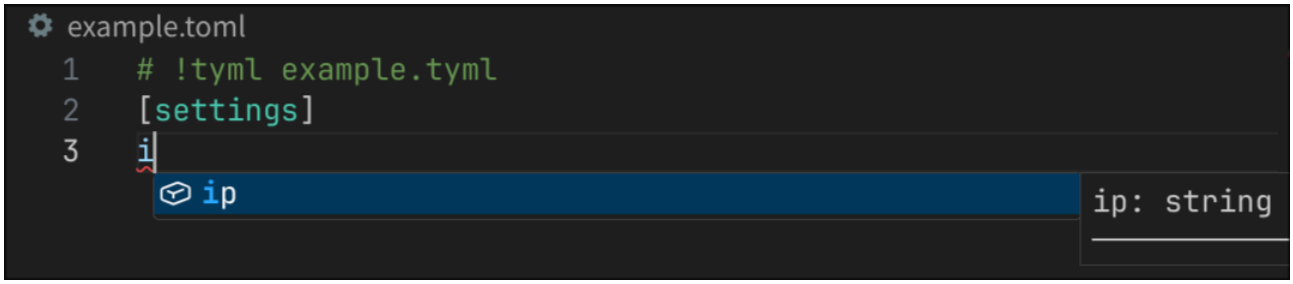
```
// define types for 'settings'
settings: {
    ip: string
    port: int
    mode: Mode
}

/// # Documents for Mode
/// this is documents for Mode type
enum Mode {
    "Debug"
    "Release"
}
```

```
# !tyml ~/Documents/test.tyml
[settings]
ip = "192.168.1.1"
port = 25565
mode = "Debug"
```

開発はプログラミング言語 Rust を用いて行った.採用理由としては,UTF-8 の扱いが簡単かつ高速に動作する点が挙げられる.また,開発においては積極的な unsafe 機能を用いた最適化を行った.

2-2. LSP サーバーの開発

ほとんどの設定用言語はエディタ上で書くことが多いので,書きながらリアルタイムでエラーチェックを行えるように LSP サーバーを開発した.
LSP とは Language Server Protocol の略称で,このプロトコルにしたがってサーバーを実装することで,モダンなエディタであればこのプロトコルにしたがって実装されているものが多く,複数のエディタ上での補完やリアルタイムなエラー表示を実現することができる.



既存の設定用マークアップ言語に追記して使用できる型定義言語 TYML 活動報告書

代表・メンバー 阿部 誠大 Y200099
アドバイザー教員 芝公仁

3. 苦労した点

今回は LSP サーバーの開発を行ったが,この機能が最も実装に苦労した. まず,エディタ上で補完を行う場合,当然ながら取得できるテキストは入力中のもの,つまり構造上不完全なテキストが提供され,それをうまく処理しつつ補完候補を入力位置に応じて取得する必要がある.この機能を実装するには,パーサーを不完全な入力でも対応できるように設計する必要があり,文字の位置情報もすべて持っておく必要がある.なので時間をかけて慎重に設計した.
また,LSP サーバーと共にフォーマッタの開発も行ったが,これにも多大な労力を費やした.今回の LSP サーバーは JSON, TOML, INI, TYML の 4 つに対応しており,すべて共通のルーチンを使うため,どの言語でも安定してなおかつ冪等性のある結果を得られるように設計する必要があり,非常に苦労した.

4. 得られた結果

今回の TYML の開発で,LSP サーバーに対する設計と言語設計の概観を得ることができた.設計や LSP サーバーの開発で苦労することも多かったが,数カ月間にも及ぶ開発期間を経て,設計力と開発力の両方をまた一段と鍛え上げることができたと感じている.最終的にはすべての機能,つまり JSON, TOML, INI に対する型チェックと LSP サーバーの機能を開発し終えることができた.今後も追加機能の開発を続けていきたいと考えている.
また,このレポート執筆時点で GitHub のリポジトリのスター数は 33 に到達しており,多くの方に支持していただくことができた.
よって,このプロジェクトは当初の目的を概ね達成できたと考えている.