

Angular Language Service

AngularMix and `<anglebrackets/>`

Nov. 19, 2019



Keen Yee Liao
@liauky



Keen Yee Liao

Software Engineer @ Google

- Angular tooling team
- Server-side rendering (Universal), Bazel, CLI
- Language Service

Angular Ecosystem



Angular



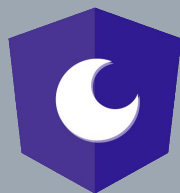
Protractor



Forms



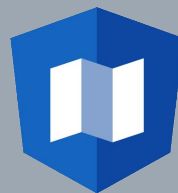
PWA



Augury



Language
Services



Router



Elements



CDK



Universal



Karma



Labs



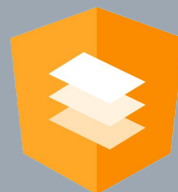
Compiler



i18n



Http



Material

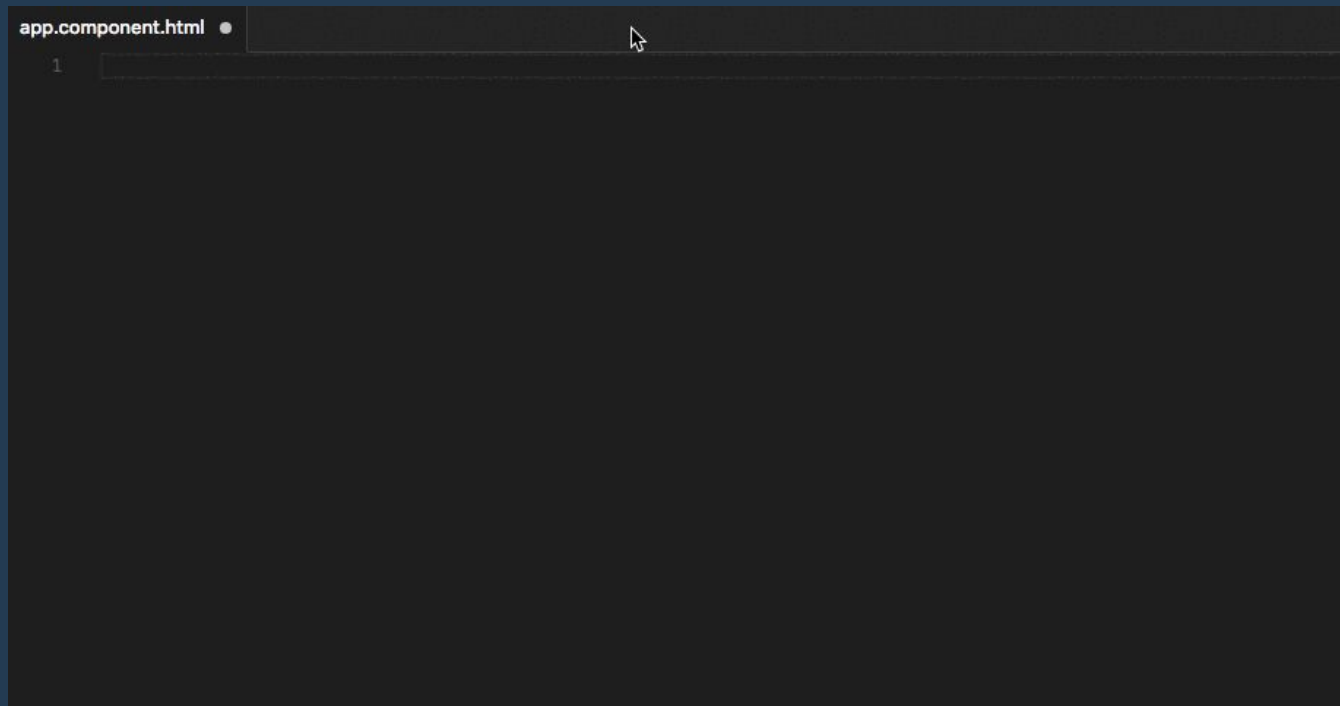


Animations



CLI

Angular Language Service



Angular Language Service

Main features:

1. Code Completion (IntelliSense)
2. Hover Tooltip (Quick Info)
3. Go To Definition
4. Diagnostics Check

Overview

- State of language service
- Architecture
- Requirements
- What's new
- How it works
- Future plans



State of Language Service



[@angular/vscode-ng-language-service](https://github.com/angular/vscode-ng-language-service)



Architecture



```
// app.component.ts
```

```
@Component({
  templateUrl: './app.component.html',
})
export class AppComponent {
  show: boolean = true;
  heroes: Hero[] = [
    { name: 'Thor', id: 42 },
  ];
}
```

```
<!-- app.component.html -->
```

```
<div *ngIf="show && heroes.length > 0">
  {{heroes[0].name}} is a hero.
</div>
```



Basic Requirements

- Parse TypeScript
- Parse HTML
- Understand Angular-specific bits
 - Bind TypeScript and HTML
 - Determine relationship between
 - i. Component ↔ Template
 - ii. Module ↔ Component
 - iii. Third-party modules / components



Compilation vs Real Time IntelliSense

Compilation

- One off
- Short-lived process
- Works with files on disk
- No suggestions
- OK to be slow
- Fail on errors

IntelliSense

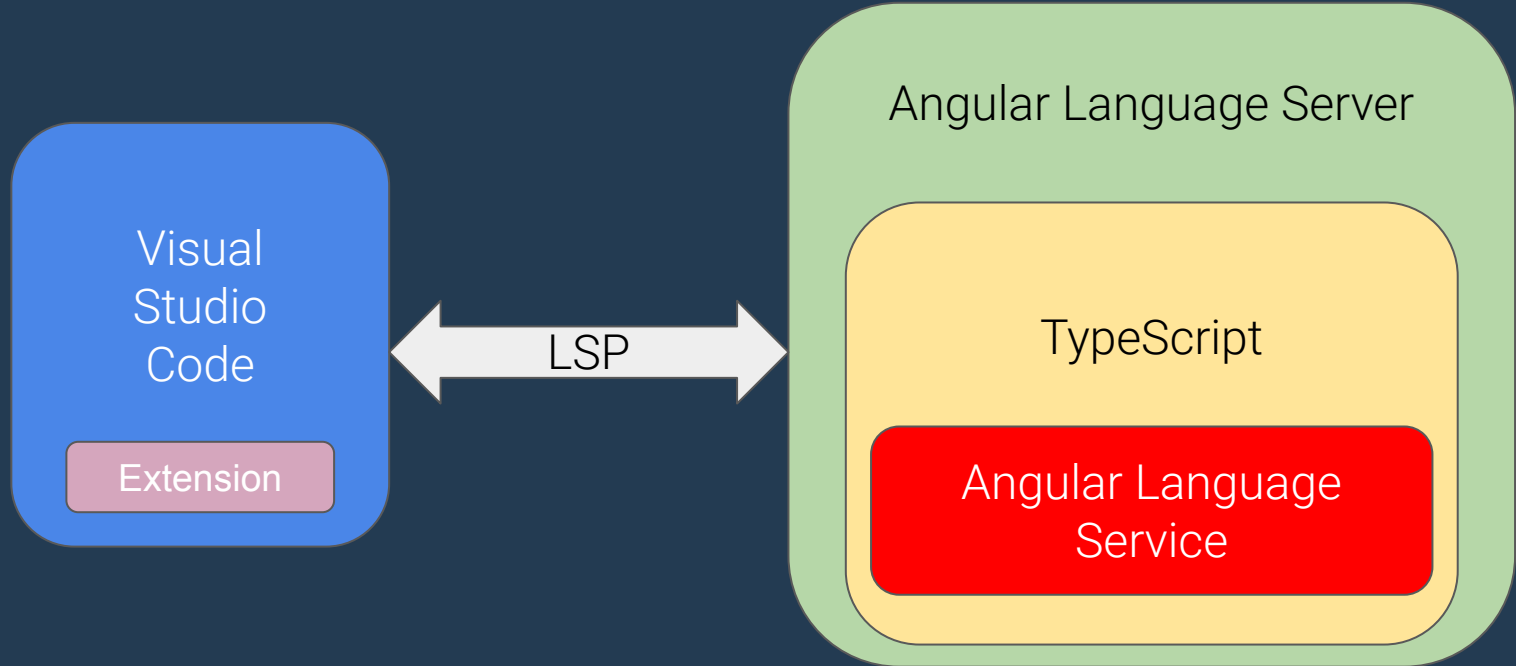
- Continuous, incremental
- Long-lived process
- Works with buffers in editors
- Provide suggestions
- Must not provide stale results
- Must proceed with errors



Language service is dynamic

Let's look at the architecture

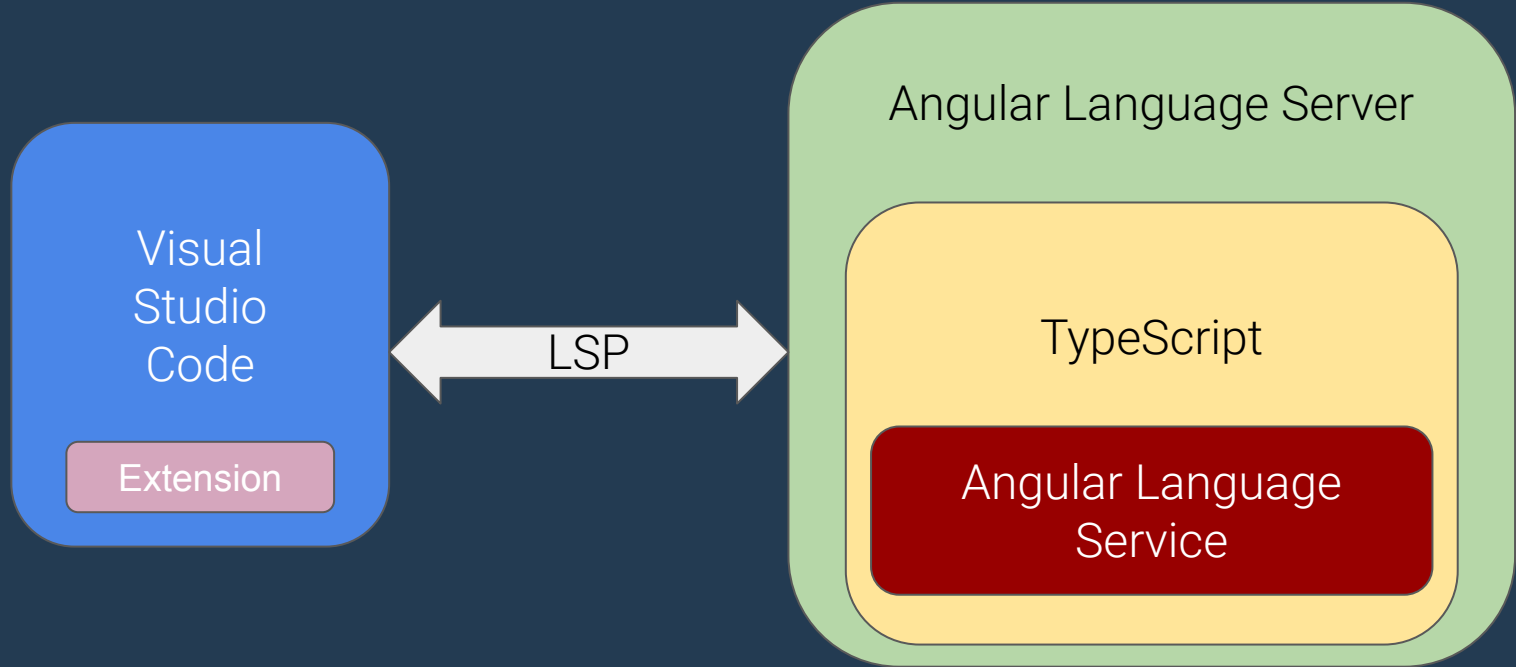
Architecture



Constituents

- Client-side Extension
 - @angular/vscode-ng-language-service
- Language Server Protocol
 - <https://microsoft.github.io/language-server-protocol/>
- Language Server
 - @angular/language-server
- Language Service
 - @angular/language-service

Architecture



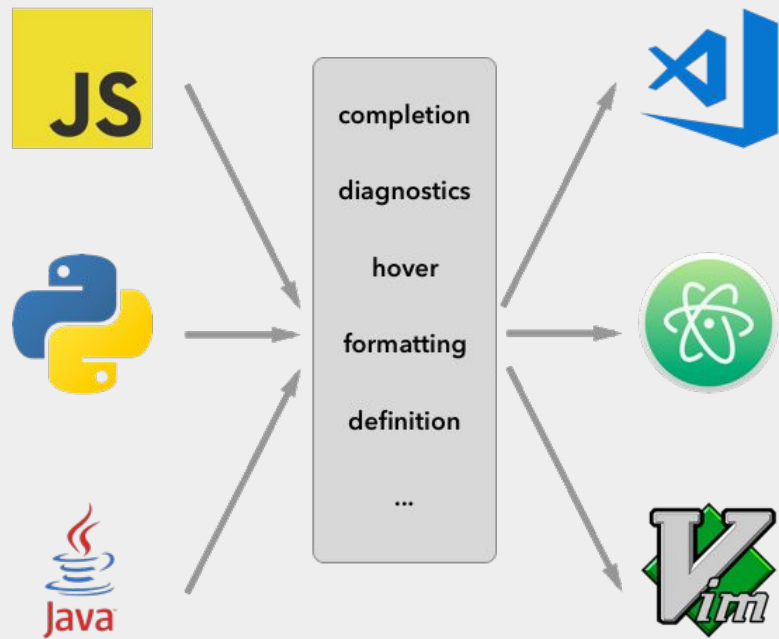
Language Server Protocol (LSP)

- Standardize communication between servers and IDEs.
- A Language Server can be reused in multiple development tools.
- Uses JSON-RPC channel.
- Request vs Notification

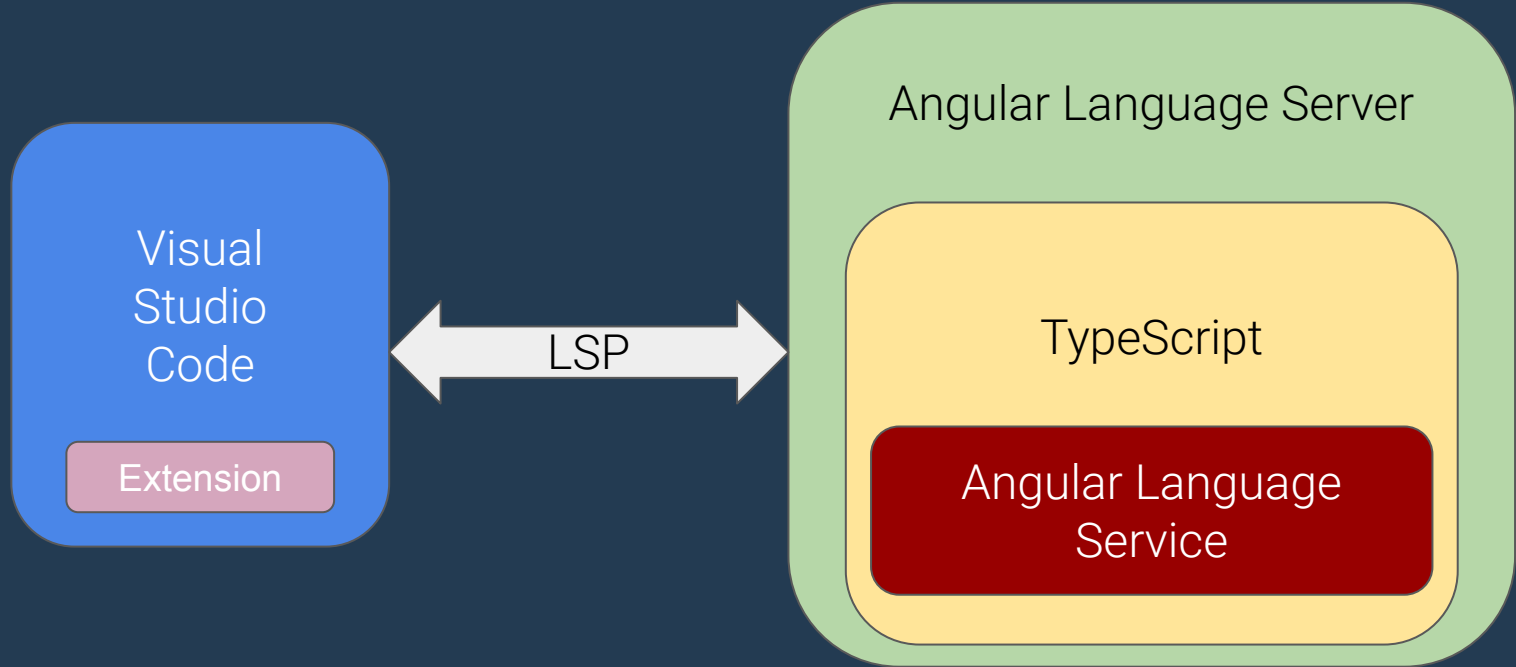
NO LSP



LSP



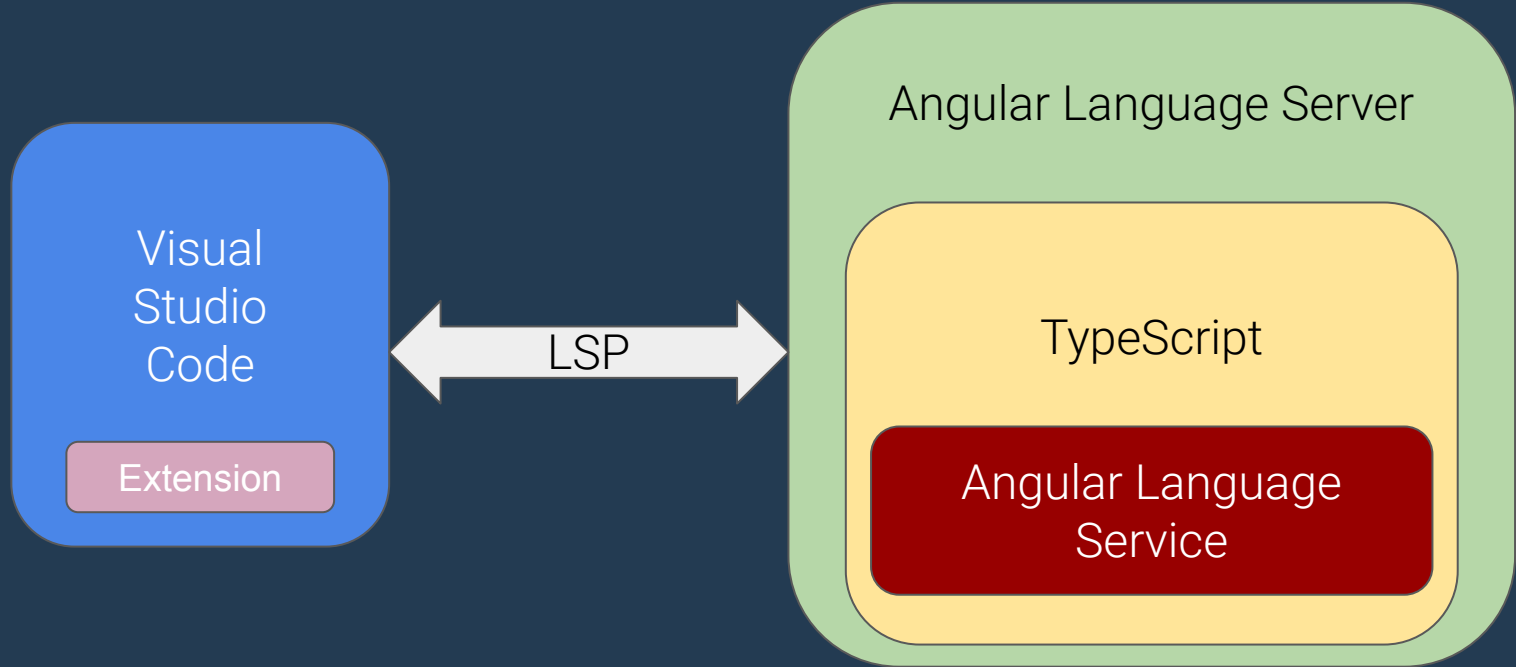
Architecture



Client-side Extension

- Gather user options
 - TypeScript version, log file, etc.
- Spawn server
- Send document events to the server
 - Document selector: TypeScript, HTML
 - Create / Modify / Delete / Save, etc.
- Tiny
 - Share extension host with other IDE extensions

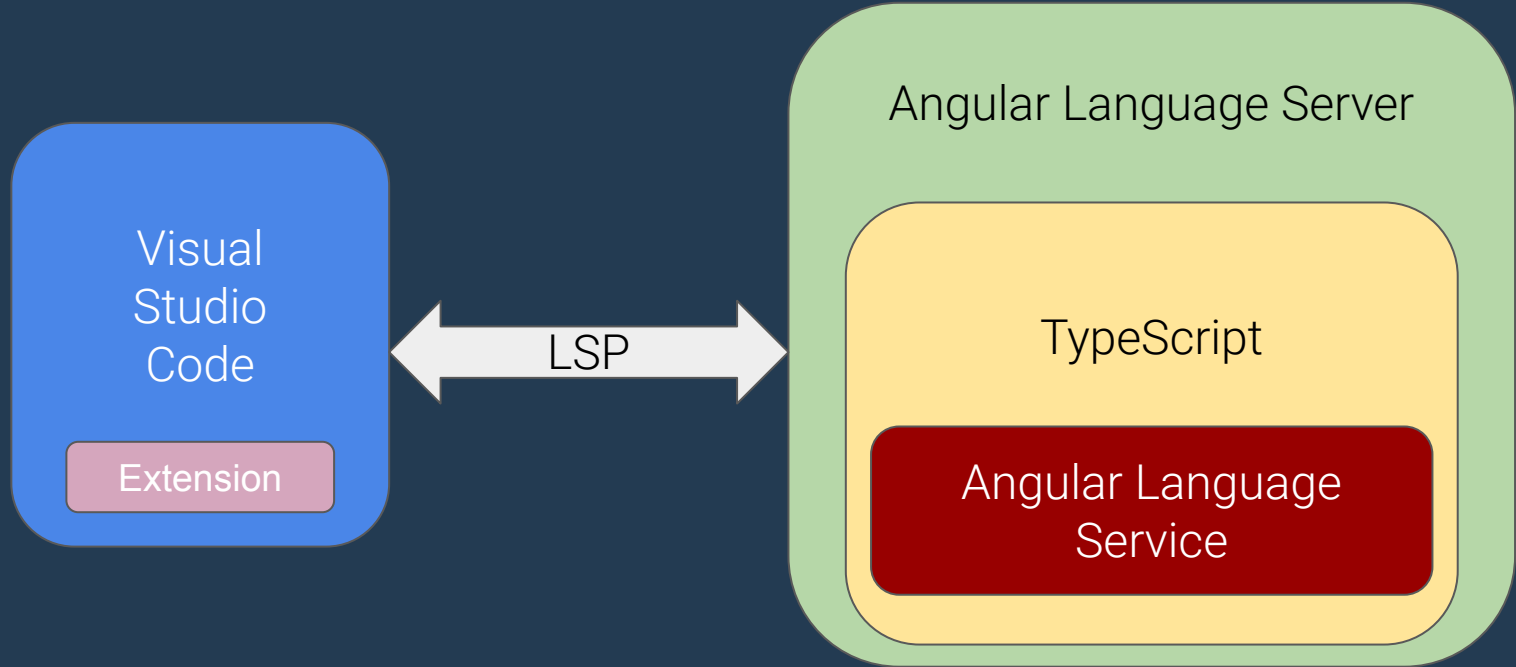
Architecture



Language Server

- IDE-agnostic language server
- Handles the **mechanics** of project management
 - Parse process.argv, create logger, etc.
 - Set up ServerHost for FileSystem / Node.js operations
 - Host TypeScript compiler
 - Convert TypeScript ↔ LSP data types

Architecture



Language Service

- Handles the **semantics** of Angular templates
- tsserver plugin
 - `getSemanticDiagnostics(fileName: string): Diagnostic[]`
 - `getCompletionsAtPosition(fileName: string, position: number)`
 - `getQuickInfoAtPosition(fileName: string, position: number)`
 - `getDefinitionAndBoundSpan(fileName: string, position: number)`

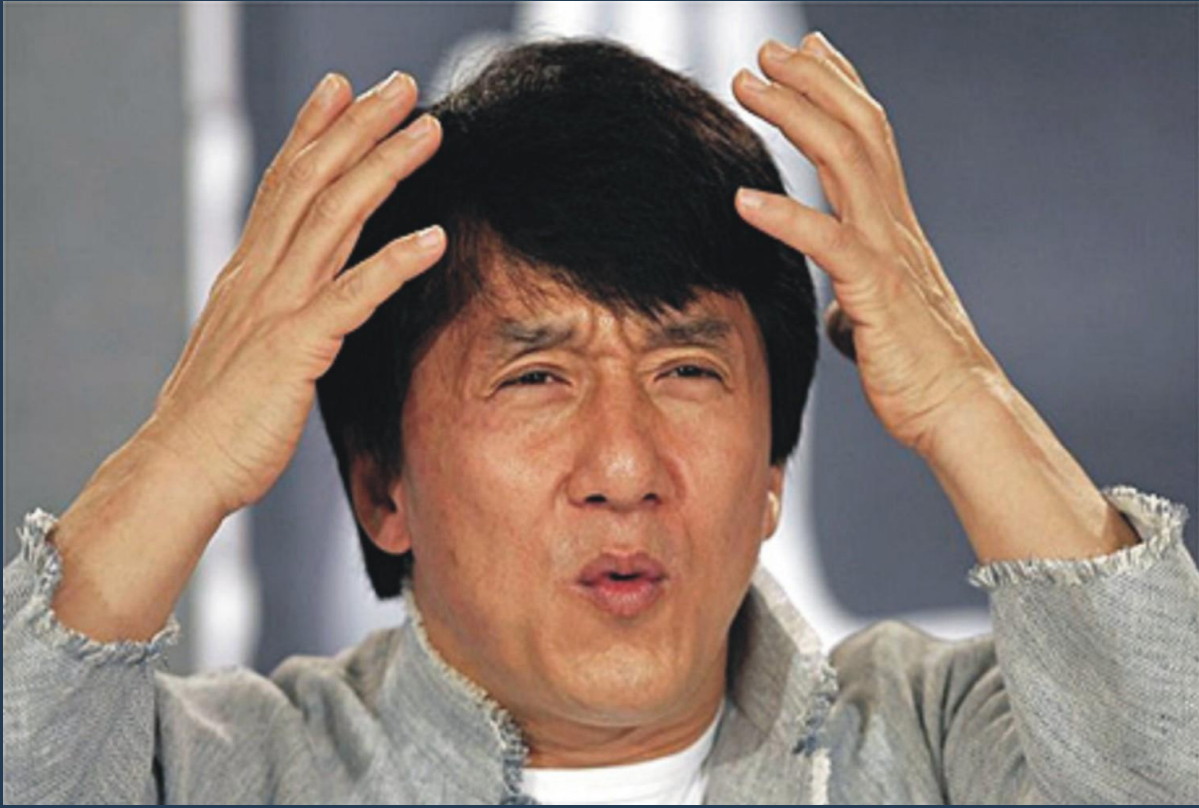


Architecture v1

Pre version 9

Pre version 9

- Forked TypeScript server (~v2.2)
- Duplicate TypeScript compiler
- Issues
 - **Performance**
 - Inconsistent error messages
 - False positives and false negatives



Requirements

- Fast & efficient
- Compatible with other IDEs
- Support external templates
- Version compatibility



Fast & Efficient

- Fast
 - Start up
 - Aggressive caching
- Efficient
 - Minimal computation on each change
 - Reuse existing TypeScript compiler

Compatible with other IDEs

- Client-side
 - Inherently IDE specific
 - As little logic as possible
- Server-side
 - Must be IDE agnostic
 - Language Server Protocol ([LSP](#))

Support External Templates

- Must work with .html
- Proper syntax highlighting
- Reuse editor's native HTML extension



```
// app.component.ts
```

```
@Component({
  templateUrl: './app.component.html',
})
export class AppComponent {
  show: boolean = true;
  heroes: Hero[] = [
    { name: 'Thor', id: 42 },
  ];
}
```

```
<!-- app.component.html -->
```

```
<div *ngIf="show && heroes.length > 0">
  {{heroes[0].name}} is a hero.
</div>
```


Version Compatibility

- Use workspace version by default
 - typescript
 - @angular/language-service

Options

- Local TypeScript plugin
 - Project-specific
- Global plugin
 - Treat HTML files like TypeScript
 - Experimental
 - Launched whenever tsserver is spawned
- Standalone extension



Local Plugin

Global Plugin

Standalone Extension

Local TypeScript Plugin

- ✓ Fast & efficient
- ✓ Compatible with other IDEs
- ✗ Support external templates
- ✓ Version compatibility

Local Plugin

Global Plugin

Standalone Extension

Global Plugin

- ✓ Fast & efficient
- ✗ Compatible with other IDEs
- ✓ Support external templates *
- ✓ Version compatibility

Local Plugin

Global Plugin

Standalone Extension

Standalone Extension

- ✗ Fast & efficient
- ✓ Compatible with other IDEs
- ✓ Support external templates
- ✓ Version compatibility

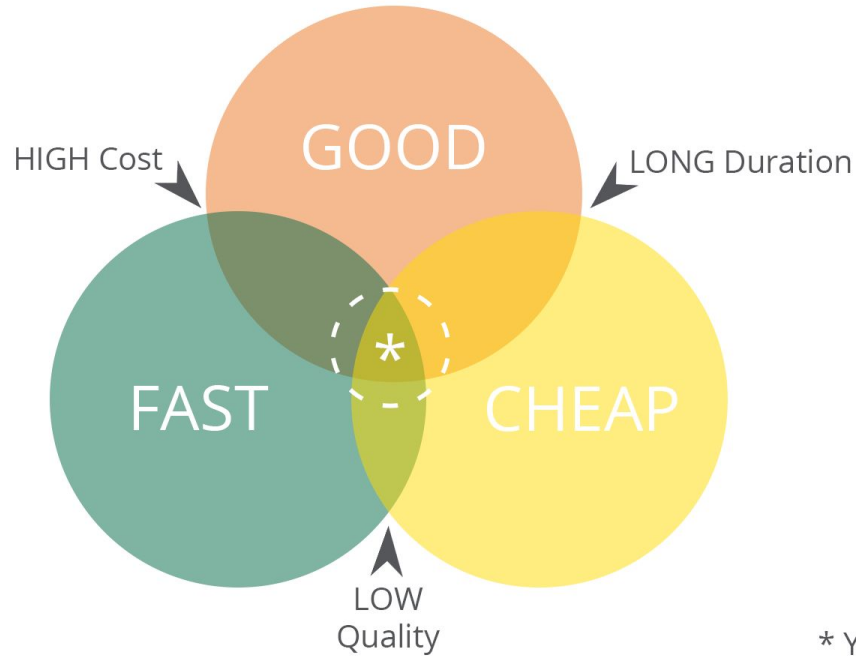


Architecture v2

Version 9 and beyond

GOOD, FAST, or CHEAP?

Please choose 2:



* You're Dreaming

Let the user decide!



external templates ?

standalone extension :

local plugin

Local plugin via tsconfig.json

```
{
  "compilerOptions": {
    "plugins": [
      {"name": "@angular/language-service"}
    ]
  }
}
```

What We Learned

- No one-size-fits-all solution
- Support incremental parsing from the start
- Comprehensive tests gives confidence in refactoring
 - Reduced test execution from 1 min → 15 sec
- TypeScript codebase excellent source of knowledge

Improvements

- Lazy evaluation - zero startup cost
- Cache module resolutions
- Redesign interfaces to invalidate caches
- Per file invalidation vs per program invalidation
- Project management via unmodified tserver
- Optimize for template updates

New Features in Version 9

- Go to definition for `templateUrl` and `styleUrls`
- Template tooltip shows containing `NgModule`
- More consistent error messages
- Fixed long-standing bugs
- Smaller bundle

Tips for Better Performance

- Startup
 - Bundling via Rollup (26MB → 9MB for .vsix)
 - Minimal set of prod dependencies



Running Extensions — ngtemplate

Running Extensions ×

Configuration Editing	Activation: 13ms
Node Debug Auto-attach	Startup Activation: 3ms
Emmet	Startup Activation: 9ms
Extension Authoring	Activation: 12ms
Git	Startup Activation: 16ms
JSON Language Features	Activation: 45ms
Merge Conflict	Startup Activation: 3ms
NPM support for VS Code	Startup Activation: 5ms
TypeScript and JavaScript Language Features	Activation: 58ms
Angular Language Service	Activation: 47ms

⊗ 0 ⚠ 0

Tips for Better Performance

- Startup
 - Bundling via Rollup (26MB → 9MB)
 - Minimal set of prod dependencies
- Runtime
 - TypeScript module resolution
 - `.ts` → `.tsx` → `.d.ts` → `package.json` (“types”)
 - Short circuit `.tsx` lookup (system calls are expensive)
 - Diagnostics check
 - 200ms delay after user stops typing
 - Most recently-used documents first

TypeScript Module Resolution

100x

Loading
@angular/core
symbols

- No cache: 2581
- Cache: 26

10x

Loading Tour of
Heroes symbols

- No cache: 6291
- Cache: 550

3x

Incremental
change

- No cache: 257
- Cache: 84



How It Works

How It Works

- TypeScript AST
 - `ts.Node`
 - `TypeChecker`
- Angular AST
 - HTML parser
 - Expression parser

```
// app.component.ts
```

```
@Component({
  templateUrl: './app.component.html',
})
export class AppComponent {
  show: boolean = true;
  heroes: Hero[] = [
    { name: 'Thor', id: 42 },
  ];
}
```

```
<!-- app.component.html -->
```

```
<div *ngIf="show && heroes.length > 0">
  {{heroes[0].name}} is a hero.
</div>
```

How It Works

- Initialization: Read metadata.json
- Element → Attribute
 - Attribute name → Directive
 - Attribute value → Expression
- Element → BoundedText → Interpolation → Expression
- Template Context



What's Next?

List of IDEs

- Visual Studio Code
- Visual Studio
- Eclipse via [Wild Web Developer](#)
- Emacs via [lsp-mode](#)
- Vim (?)
- Sublime (?)
- Atom (?)

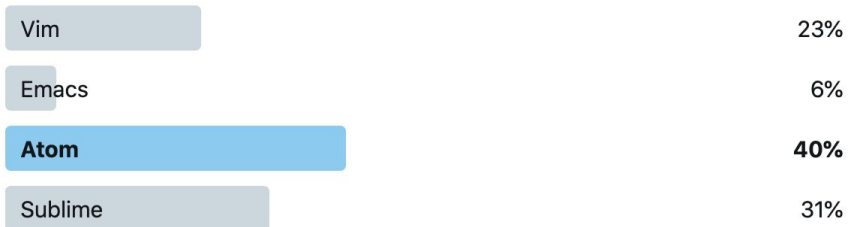




Keen
@liauky



Hey @angular developers, I'd like to find out if the language service is available in your favorite IDE. If your IDE is ****not**** vscode, please leave a comment below to let me know if the language service works!

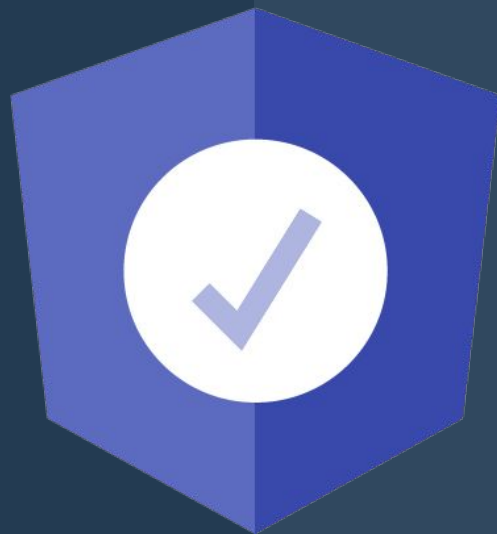


35 votes · Final results

3:33 PM · Nov 11, 2019 · [Twitter Web App](#)

Timeline

- Will be published in two weeks!
- Along with Angular version 9
- Please try it out
 - <https://github.com/angular/vscode-ng-language-service/releases>





Mission

“

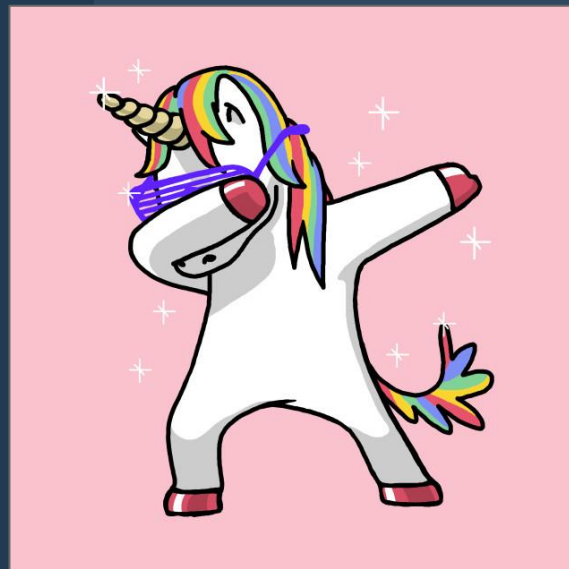
Provide a delightful user experience by bringing type, structural, and Angular knowledge into IDEs to help developers write better code.

Angular Team

”

Future Plans

- Integration with Ivy compiler (version 10)
 - More **consistent** type checking
 - Strict type checking
 - Even better performance
- Syntax highlighting for inline templates
- Code refactoring
- Make it awesome!



Thank You!

- [@angular/language-service](#)
- [@angular/language-server](#)
- [Visual Studio Marketplace](#)
- [@angular/vscode-ng-language-service](#)



Keen Yee Liao
@liauky