



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Sybit Essentials

Hosted by Alex Thalhammer

About me ...



My office 😊



Hi, it's me → [@LX_T](https://twitter.com/LX_T) 

<https://alex.thalhammer.name/>

- **Alex Thalhammer** from Graz, Austria (since 1983)
 - Angular Software Tree GmbH (since 2019)
- WebDev for 24+ years (I've come a long way baby)
- WordPress Dev (Web, PHP & jQuery, 2011 - 2017)
- **Angular Dev** (Web, TS, Rx since 2017 - NG 4.0.0 ☺)
- **Angular Evangelist, Coach & Consultant** (since 2020)
 - Member of Angular Architects <https://www.angulararchitects.io/>



About you ...

Tell us sth about yourself

- Hi, my name is ... and I currently work on ...
- I'd like to learn how to ...
- In Angular it's difficult for me to ...
- My goal(s) for the next 4 days is (are) ...

Pollings

- OS
- IDE
- Mouse or TrackPad?

Tell us sth about your team / projects

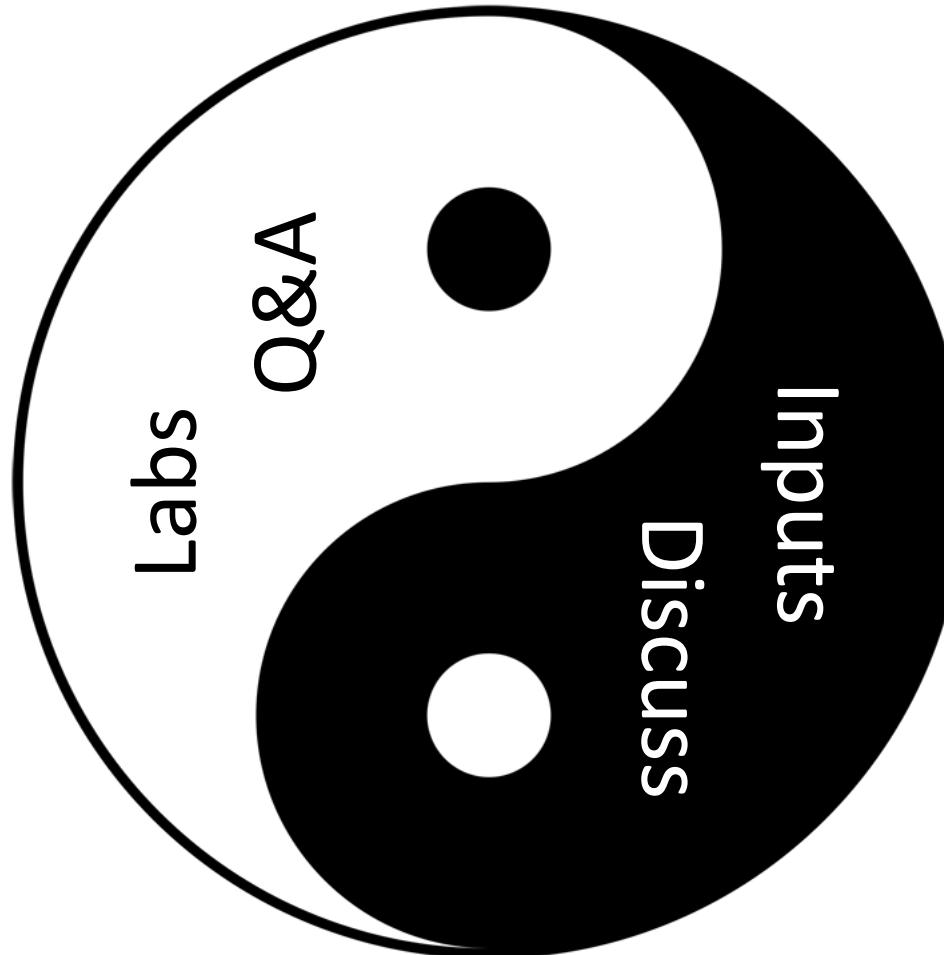
- Before I tell you sth about our workshop ☺

About this workshop ...

My Goals

- **Motivate** you for Angular
- Give **input** on essential Angular topics
- Get your **hands on** those topics (labs / exercises)
- **Discuss** open topics and **answer** your questions

Didactics



We work together I

- Don't hesitate to interrupt me at ***ANY TIME***
 - Questions
 - Remarks or
 - Feedback 😊
- Somebody else might have the same **question**
 - Be a hero and ask it
 - Seriously, it makes the workshop better for all of us 😊



We work together II

- Remote: Please turn your camera on ***ALL THE TIME***
 - Necessary for me to get your feedback
 - This will also improve the workshop for all 😊

Timetable (approx.)

- 09:00 – 12:15 (incl. 15m ☕ break)
- 1hr 🍲 break
- 13:15 – 16:30 (incl. 15m ☕ break)
- Including expected delay we will end at 17:00 latest🏁

Timetable (approx.)

- 09:00 – 13:00
- 15/20min ☕ break

Topics

TypeScript,
Angular Basics,
Component,
Service, Pipe

Standalone,
Control Flow,
Components
Deep Dive

Routing,
(Performance-
Optimierung)

RxJS (Observables)
Basics &
Operators,
Signals

Signal Based
Components,
Reactive Forms

Recap Tage 1-3,
RSG Standards vs.
Alex Best Practices

Setup for Labs

- NodeJS version ^**20.19.0** or ^22.12.0 or ^24

<https://angular.dev/reference/versions#actively-supported-versions>

Try "node -v"

Use NVM ☺



Mac Alternative: Use Homebrew



Setup

Setup for Labs I

- NodeJS version **^20.19.0 or ^22.12.0 or ^24**

<https://angular.dev/reference/versions#actively-supported-versions>

Angular	Node.js	TypeScript	RxJS
20.0.x	<code>^20.19.0 ^22.12.0 ^24.0.0</code>	<code>>=5.8.0 <5.9.0</code>	<code>^6.5.3 ^7.4.0</code>
19.2.x	<code>^18.19.1 ^20.11.1 ^22.0.0</code>	<code>>=5.5.0 <5.9.0</code>	<code>^6.5.3 ^7.4.0</code>
19.1.x	<code>^18.19.1 ^20.11.1 ^22.0.0</code>	<code>>=5.5.0 <5.8.0</code>	<code>^6.5.3 ^7.4.0</code>
19.0.x	<code>^18.19.1 ^20.11.1 ^22.0.0</code>	<code>>=5.5.0 <5.7.0</code>	<code>^6.5.3 ^7.4.0</code>
18.1.x 18.2.x	<code>^18.19.1 ^20.11.1 ^22.0.0</code>	<code>>=5.4.0 <5.6.0</code>	<code>^6.5.3 ^7.4.0</code>
18.0.x	<code>^18.19.1 ^20.11.1 ^22.0.0</code>	<code>>=5.4.0 <5.5.0</code>	<code>^6.5.3 ^7.4.0</code>

Try "node -v"

Windows / Linux: Use NVM ☺

Mac Users: Use Homebrew 🍺

Setup for Labs II

- IDE: VS Code (free) or IntelliJ/WebStorm (not free, but better IMHO)
 - Personally I prefer IntelliJ/WebStorm
- Chrome (or Chromium based alternative)
 - Other browsers are okay for private matters

Recommendations

- Use prettier
 - especially for .ts files (if not for all)
- Use ESLint
 - in your IDE of choice and/or from terminal
- Use Git
 - personally, I liked to use Sourcetree → now Fork
 - we recommend to at least commit after every lab
 - easier switching back and forth with my solution

Starter Kit Repository

- Slides
- Labs (exercises)
- & free book ☺

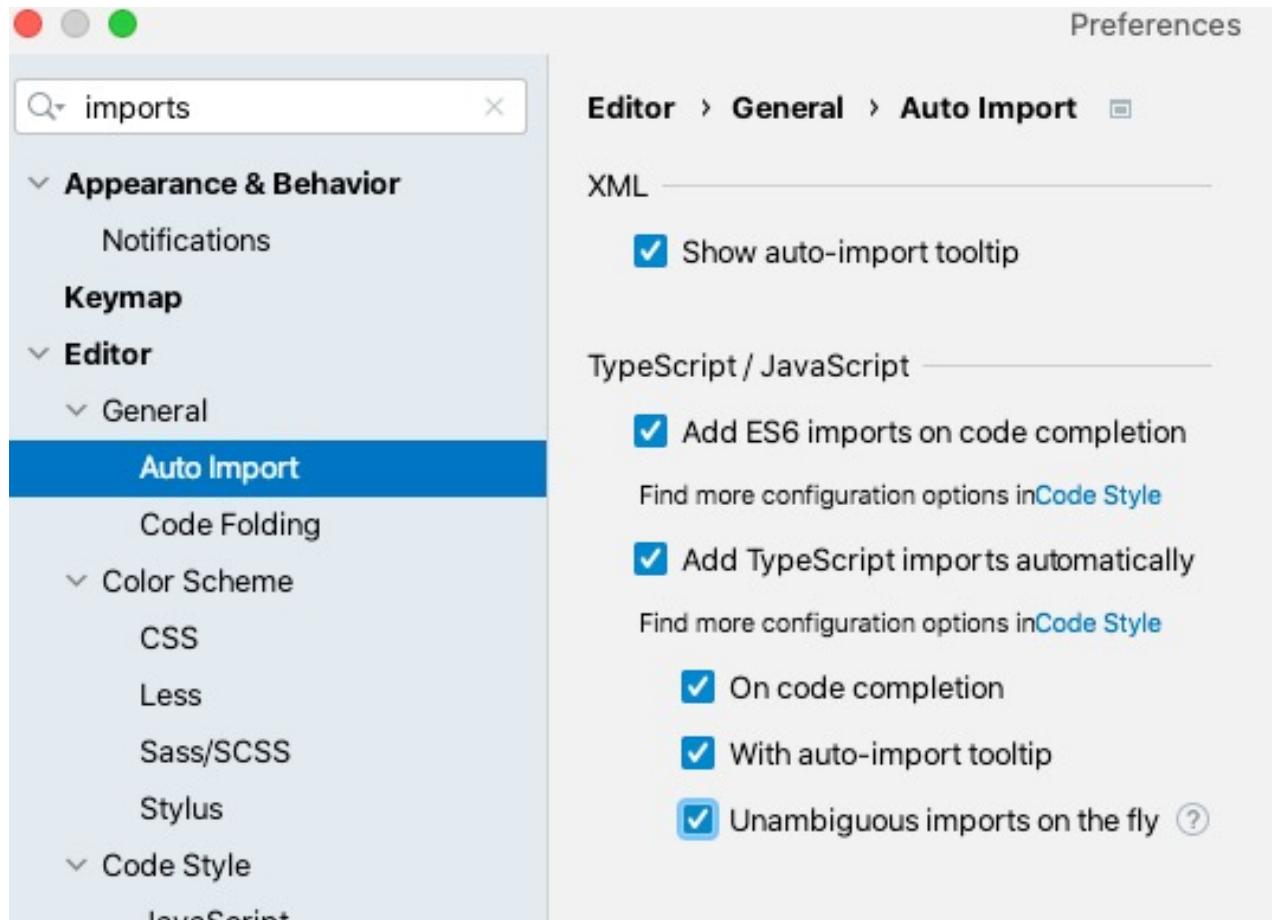
Ready for takeoff!

- What are your questions so far?

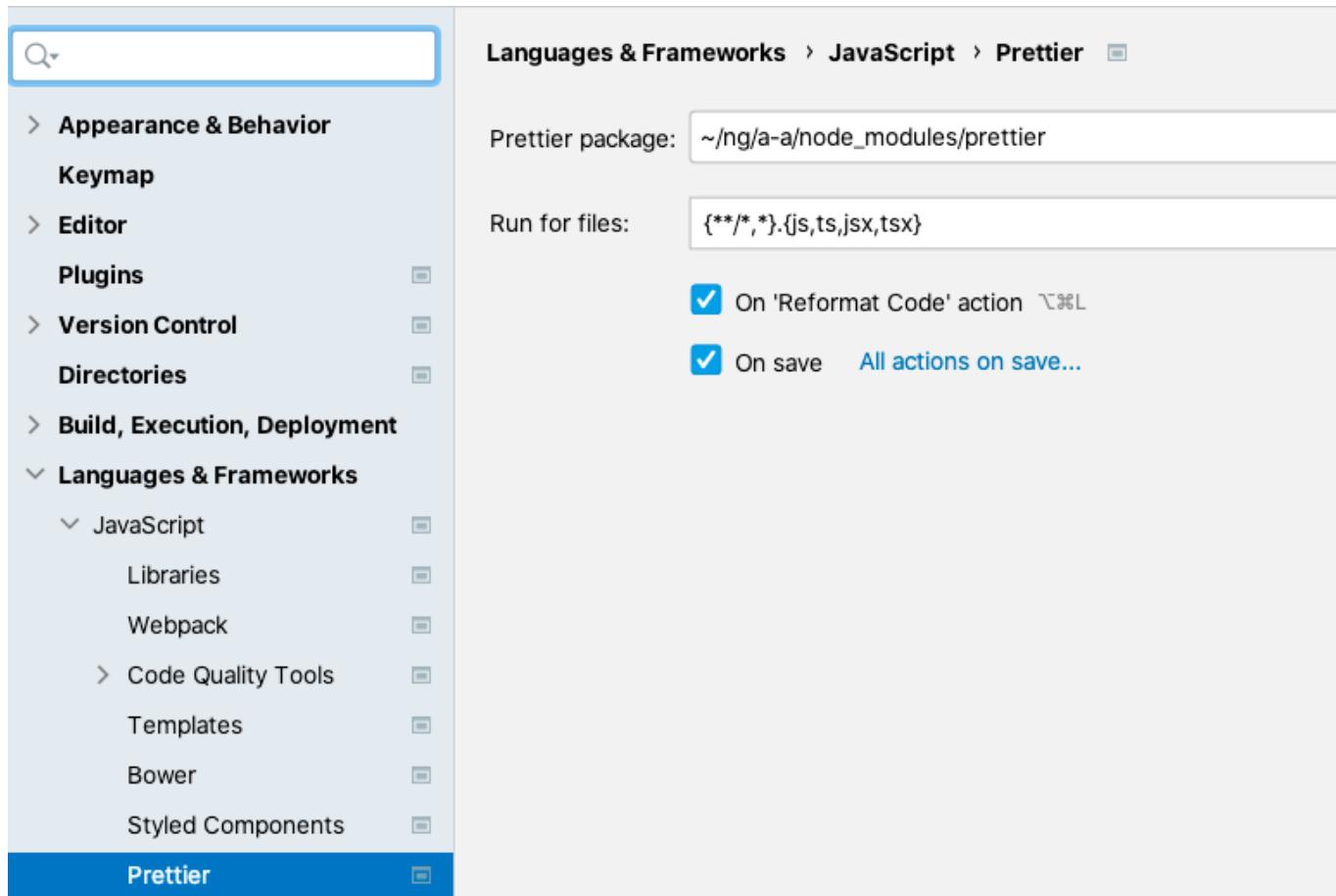
Lab

- Clone the repo [https://github.com/L-X-T/\[...\]](https://github.com/L-X-T/[...])
- Please get started with lab 00_getting_started
- yarn (or npm i) and then yarn start (or npm start)
- Take a closer look at the starter kit
- Optional: If you're using VSCode you may add the plugins mentioned

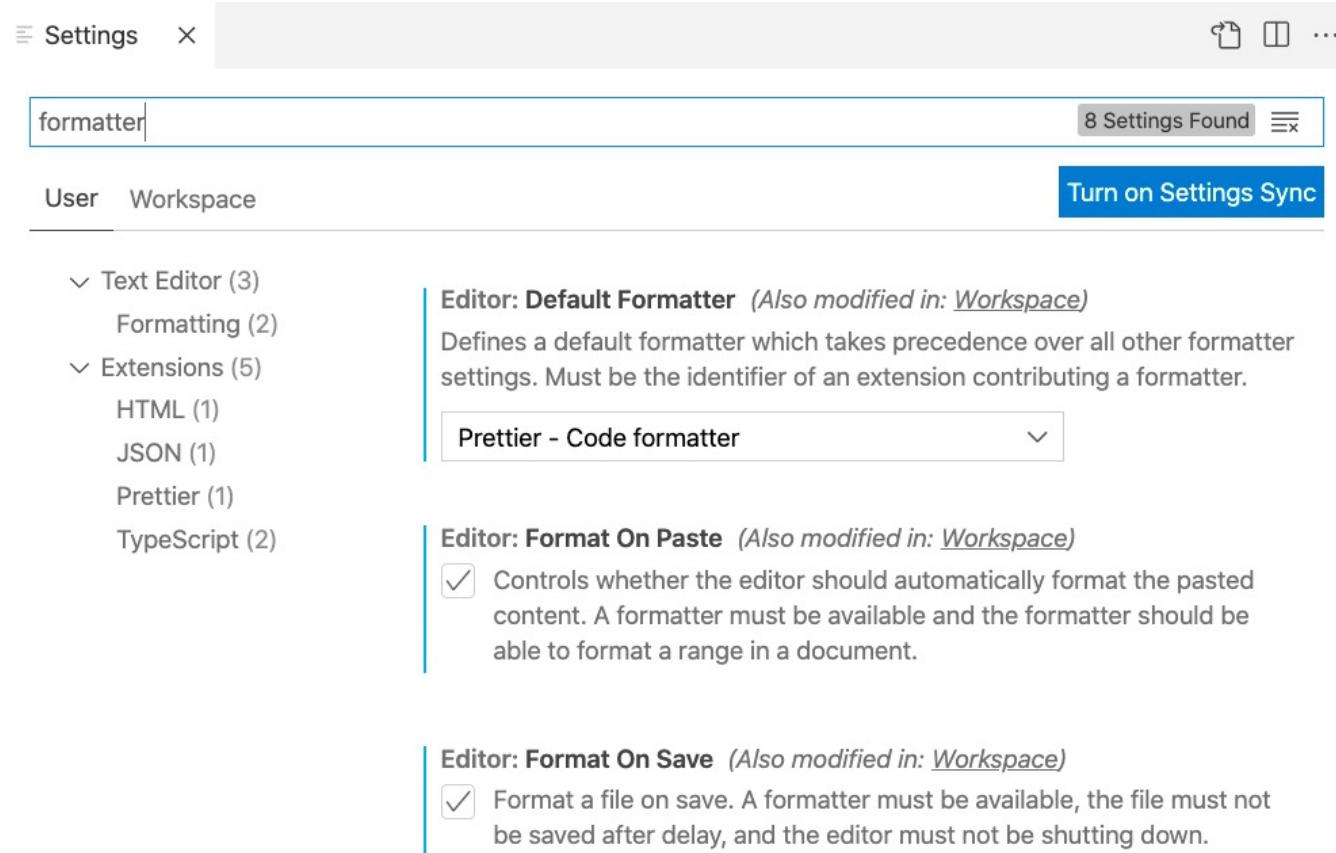
1 more thing: Web-/PhpStorm auto import



1 more thing: WebStorm prettier



1 more thing: VS Code Prettier



The screenshot shows the VS Code settings interface with the search bar containing 'formatter'. It displays 8 settings found across User and Workspace scopes. The 'User' scope is selected. A 'Turn on Settings Sync' button is visible. The settings listed are:

- Editor: Default Formatter** (Also modified in: [Workspace](#))
Defines a default formatter which takes precedence over all other formatter settings. Must be the identifier of an extension contributing a formatter.
Value: Prettier - Code formatter
- Editor: Format On Paste** (Also modified in: [Workspace](#))
 Controls whether the editor should automatically format the pasted content. A formatter must be available and the formatter should be able to format a range in a document.
- Editor: Format On Save** (Also modified in: [Workspace](#))
 Format a file on save. A formatter must be available, the file must not be saved after delay, and the editor must not be shutting down.

Disclaimer: This workshop is not about

- TypeScript / JavaScript
 - <https://www.udemy.com/course/understanding-typescript/>
 - By Maximilian Schwarzmüller who also offers *Angular*, *React* & *VueJS* (in English)
- HTML5 / CSS3 (SCSS)
 - <https://www.udemy.com/course/design-and-develop-a-killer-website-with-html5-and-css3/>
- Or all together premium course with mentor
 - <https://www.udacity.com/course/front-end-web-developer-nanodegree--nd0011>

Day 1 – Intro & Building Blocks

- Introduction
- Overview & Speedrun TypeScript
- Angular Introduction
- Angular Services

Day 2 – Async & Components

- Angular Pipes
- Recap building blocks
 - Components
 - Built-in Directives
 - Services
 - Pipes
- RxJS Observables
 - Basics
 - Operators
 - Signals
- Angular Standalone Components
- Components Deep Dive
 - Databinding
 - Content Projection
 - Smart vs dumb components
 - Component - Life cycle hooks
 - AAAND new control flow

Day 3 – Forms, Routing & Performance

- Forms & Validation
 - ~~Template driven Forms~~
 - Reactive Forms
- Routing
- Performance
 - Lazy Loading
 - CD OnPush
- Recap & Feedback

Day 4 – Forms, Routing & Performance

- Performance
 - Lazy Loading / @defer
 - CD OnPush'
- Testing
 - Unit Testing
 - E2E Testing
- RSG Styleguide vs Alex Best Practices
- Recap & Feedback