

Learnng Queue

Angular mostly will be **case-learning** (Including unit and e2e testings), however I will still try to find topic, which required deeper knowledge and walk through that course if neccessary.

**Every time I find something that requried deeper knowledge, I will drop under Angular Queue, in order to walk through it later.**

Anything I find that can be downloded and read later, will go to **Resources Archive/Digital Library** directory.

If anything that I will read or watch, will require (Or wish) to implement at the same time, I should create angular project for that in the accordingly named directory, or if it’s just testing something that doesn’t require separate memorizing, I will test that in the **ActiveLearningPractice** project.

After completing (reading or watching) anything from queue, they should me moved to **Resources Archive/Articles and Videos References** directory.

**Here I can look for articles:**

[Angular in-depth](https://indepth.dev/angular)

**Angular Queue (Read Or Watch)**

**Read:**

* *[HTTP/2](https://http2.github.io/faq/" \l "why-is-http2-multiplexed)*
* *[How Angular transofrms your code](https://medium.com/angular-in-depth/do-you-know-how-angular-transforms-your-code-7943b9d32829)*
* *[DI: Node injector](https://medium.com/angular-in-depth/angular-di-getting-to-know-the-ivy-nodeinjector-33b815642a8e)*
* *[Slow loading](https://medium.com/angular-in-depth/angular-show-loading-indicator-when-obs-async-is-not-yet-resolved-9d8e5497dd8)*
* *[RxJs operators that can bite](https://medium.com/angular-in-depth/mastering-rxjs-operators-and-functions-that-can-bite-you-when-you-dont-expect-cb2047cf5d4c)*
* *[The best practices in error handling](https://medium.com/angular-in-depth/expecting-the-unexpected-best-practices-for-error-handling-in-angular-21c3662ef9e4)*
* *[Everything about change detection](https://indepth.dev/posts/1053/everything-you-need-to-know-about-change-detection-in-angular)*
* *[AOT Compilation](https://blog.mgechev.com/2016/08/14/ahead-of-time-compilation-angular-offline-precompilation/)*
* *[Change detection in JS frameworks](https://teropa.info/blog/2015/03/02/change-and-its-detection-in-javascript-frameworks.html)*
* *[Developing tabs component](https://blog.thoughtram.io/angular/2015/04/09/developing-a-tabs-component-in-angular-2.html)*
* *[Micro and Macro tasks in JS](https://stackoverflow.com/questions/25915634/difference-between-microtask-and-macrotask-within-an-event-loop-context" \l ":~:text=task%20in%20microtask%20queue%20will,task(a%20macrotask)%20too.)*
* *[Understanding Zones in JS](https://blog.thoughtram.io/angular/2016/01/22/understanding-zones.html)*
* *[Zones in Angular](https://blog.thoughtram.io/angular/2016/02/01/zones-in-angular-2.html)*
* *[Change detection reinvented [VIDEO]](https://www.youtube.com/watch?v=jvKGQSFQf10)*
* *[ImmutableJS in Angular](https://www.jvandemo.com/how-i-optimized-minesweeper-using-angular-2-and-immutable-js-to-make-it-insanely-fast/)*
* *[Taking advantage of Observables in Angular](https://blog.thoughtram.io/angular/2016/01/06/taking-advantage-of-observables-in-angular2.html)*

*Not free:*

* [Build component library](https://medium.com/angular-in-depth/how-to-build-a-component-library-with-angular-and-storybook-718278ab976)
* [How to unsubscribe, the best way](https://medium.com/angular-in-depth/the-best-way-to-unsubscribe-rxjs-observable-in-the-angular-applications-d8f9aa42f6a0)
* *[Error interceptor](https://michael-karen.medium.com/angular-error-interceptor-1abb8066359c)*

**Watch:**

***After finishing above-mentioned item(‘s), I will get back to Angular in-depth website and will follow articles from top to bottom.***