



## Trainer

- Manfred Steyer
- **SOFTWARE**architekt.at
- Trainer & Consultant
- Focus: Angular
- Google Developer Expert (GDE)



ANGULAR  
**COLLEGE**  
INSIDE KNOWLEDGE



## Goals

- Understanding the concepts of Angular

## Agenda

- TypeScript “on the fly”
- Databinding
- Components
- HTTP and Service Communication
- Services and Dependency Injection
- Modules
- Navigation Structures with Routing

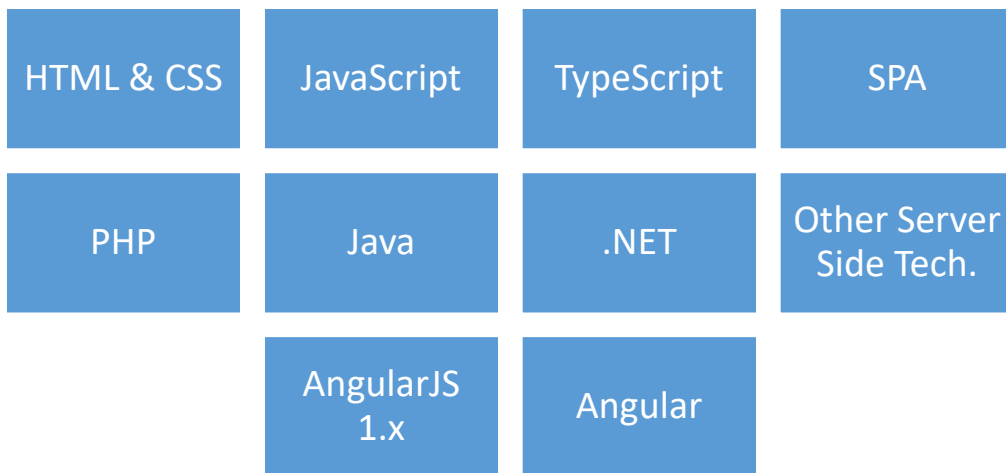
## Didactical Approach

- Workshop
- Consistent example
- Morning
  - Presentation
  - Live coding
- Afternoon
  - Examples

## Setup

- NodeJS (6.11.x)
- Editor/IDE
  - e.g. Visual Studio Code or IntelliJ/WebStorm
- Chrome

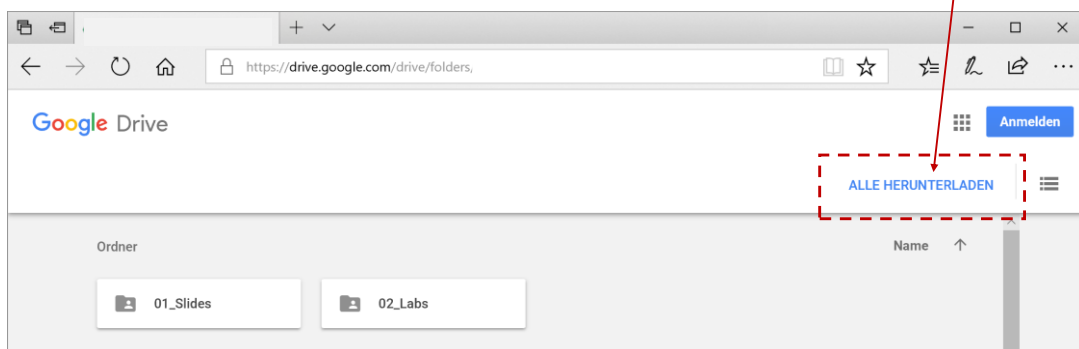
## Quick Poll ...



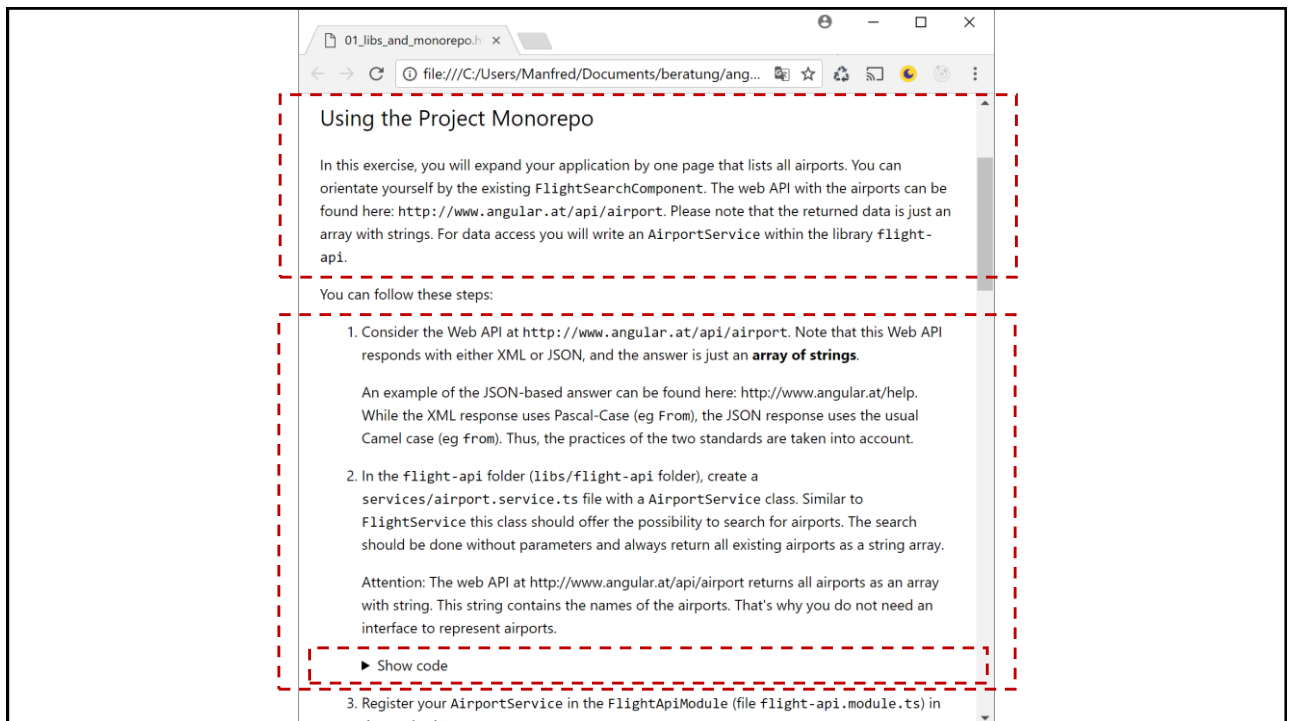
SOFTWAREarchitekt.at

## Downloads

- Start download now; we need it in the afternoon
- <https://tinyurl.com/y7a4rdji>
- Please bookmark this link; we'll need it later too



SOFTWAREarchitekt.at



01\_libs\_and\_monorepo.h x

file:///C:/Users/Manfred/Documents/beratung/ang...

## Using the Project Monorepo

In this exercise, you will expand your application by one page that lists all airports. You can orientate yourself by the existing `FlightSearchComponent`. The web API with the airports can be found here: <http://www.angular.at/api/airport>. Please note that the returned data is just an array with strings. For data access you will write an `AirportService` within the library `flight-api`.

You can follow these steps:

1. Consider the Web API at <http://www.angular.at/api/airport>. Note that this Web API responds with either XML or JSON, and the answer is just an **array of strings**.  
  
An example of the JSON-based answer can be found here: <http://www.angular.at/help>. While the XML response uses Pascal-Case (eg `From`), the JSON response uses the usual Camel case (eg `from`). Thus, the practices of the two standards are taken into account.
2. In the `flight-api` folder (`libs/flight-api` folder), create a `services/airport.service.ts` file with a `AirportService` class. Similar to `FlightService` this class should offer the possibility to search for airports. The search should be done without parameters and always return all existing airports as a string array.  
  
Attention: The web API at <http://www.angular.at/api/airport> returns all airports as an array with string. This string contains the names of the airports. That's why you do not need an interface to represent airports.  
  
► Show code
3. Register your `AirportService` in the `FlightApiModule` (file `flight-api.module.ts`) in the method `forRoot`.