Transformations with Pipes

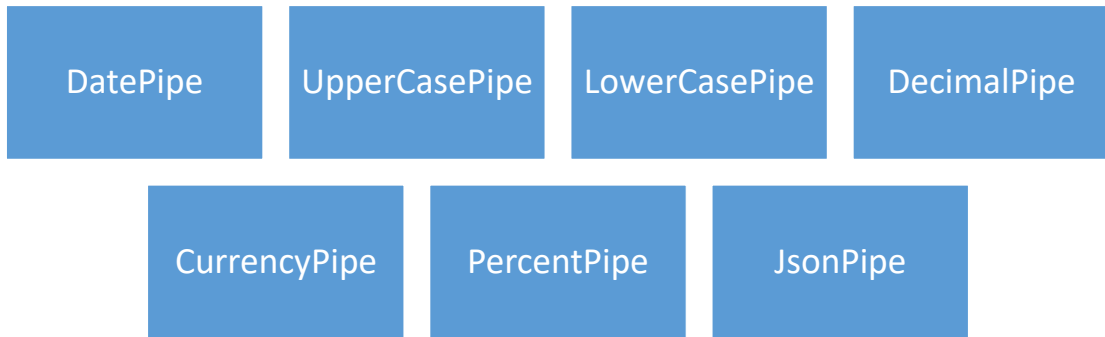**Michael Zikes**
SOFTWAREarchitekt.at

The idea of Pipes

- Transform data while binding them („formatting")
- Grammar:
  - {{ f.von **| location** }}
  - {{ f.von **| location:'short'** }}
  - {{ f.von **| location:'short':'en'** }}
  - {{ f.von **| location:'short':'en' | uppercase** }}

SOFTWAREarchitekt.at

Build-in Pipes

DatePipe	UpperCasePipe	LowerCasePipe	DecimalPipe

CurrencyPipe	PercentPipe	JsonPipe

SOFTWAREarchitekt.at

Pure Pipes

- Output depends on input parameters
- Consequence: possibly better performance

SOFTWAREarchitekt.at

Declaration of a Pipe

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
    name: 'location',
    pure: true
})
export class LocationPipe implements PipeTransform {

    transform(value: any, param1: any, param2: any, …): any {
            […]
            return […];
        }
}
```

SOFTWAREarchitekt.at

Register a Pipe in a Module

```
@NgModule({
    imports: [
        BrowserModule, HttpModule, FormsModule
    ],
    declarations: [
        AppComponent, FlightSearchComponent, LocationPipe
    ],
    bootstrap: [
        AppComponent
    ]
})
export class AppModule {
}
```

SOFTWAREarchitekt.at

DEMO

SOFTWARE*architekt.at*