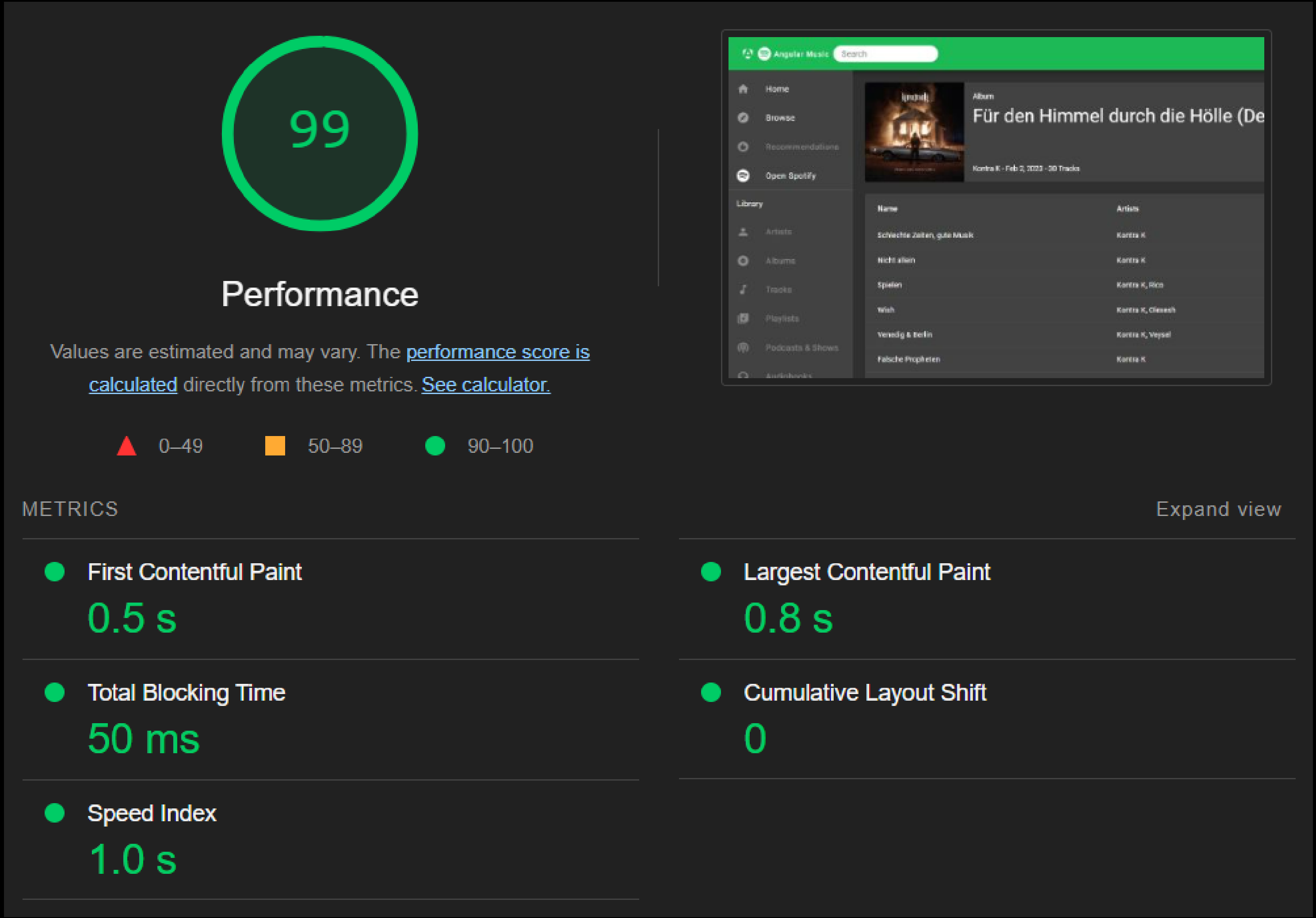
Angular Performance Optimizations

Software Craft Community @DATEV



99

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Angular Music Search

Home
Browse
Recommendations
Open Spotify
Library
Artists
Albums
Tracks
Playlists
Podcasts & Shows
Audiobooks

Album
Für den Himmel durch die Hölle (De

Kontra K - Feb 3, 2023 - 30 Tracks

Name	Artists
Schlechte Zeiten, gute Musik	Kontra K
Nicht allein	Kontra K
Spielen	Kontra K, Rico
Wish	Kontra K, Clueso
Venedig & Berlin	Kontra K, Veysel
Falsche Propheten	Kontra K

METRICS Expand view

● First Contentful Paint
0.5 s

● Largest Contentful Paint
0.8 s

● Total Blocking Time
50 ms

● Cumulative Layout Shift
0

● Speed Index
1.0 s

Software Craft Community @DATEV

Angular Performance Optimizations

How to elevate the performance of an Angular App to the next level 🚀

Angular

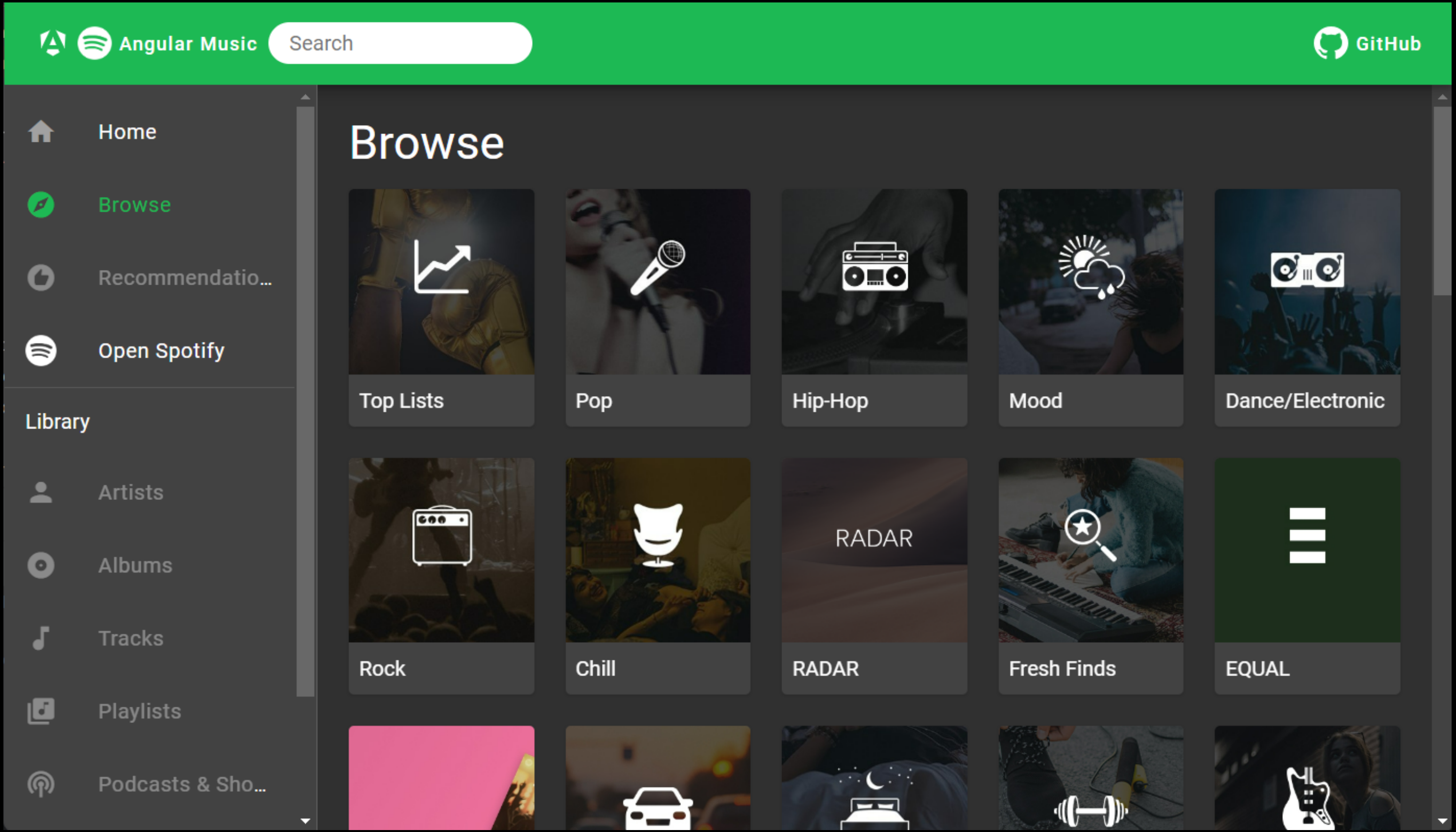
github.com/sonallux @sonallux sonallux.github.io

Software Craft Community @DATEV

Content

- Application
- Measurements
- Optimization steps
  - General
  - Standalone components
  - Lazy loading
  - Image loading
  - Built-in Control flow
  - esbuild
  - Server-side rendering
- Results

Angular Music



Measurements

- Lines of code
- Build duration
- Initial bundle size
- Lighthouse performance score
  - First Contentful Paint
  - Largest Contentful Paint
  - Total Blocking Time
  - Cumulative Layout Shift
  - Speed Index

General

- Choose base components wisely (`MatToolbar`, `MatCard`, `MatSidenav`)
- Do not use Components CSS files when using TailwindCSS
- Use CSS instead of JavaScript
- Reduce Backend request count and payload
- Cache JavaScript and CSS bundles

Standalone components [1]

- Available since Angular 15

```ts
@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
})
export class LoginComponent {}

@NgModule({
  declarations: [LoginComponent],
  imports: [CommonModule, MatButtonModule],
  exports: [LoginComponent]
})
export class LoginModule {}
```

```ts
@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  standalone: true,
  imports: [NgIf, MatButtonModule],
})
export class LoginComponent {}
```

1. https://github.com/sonallux/angular-music/pull/106

Standalone components [1]

Stats	Relative change
Build time	+ 2%
Lines of code	- 2%
Initial bundle size	- 2%

1. https://github.com/sonallux/angular-music/pull/106

Lazy loading [1]

- Lazy load routes
- Lazy load animations `provideAnimationsAsync()` (>= Angular 17)
- Defer component loading with `@defer` (only works for standalone components) (>= Angular 17)

```html
@defer (on idle) {
  <large-component />
} @placeholder (minimum 500ms) {
  <p>Placeholder content</p>
}
```

1. https://github.com/sonallux/angular-music/pull/107

Lazy loading [1]

Stats	Relative change
Build time	- 5%
Lines of Code	0%
Initial bundle size	- 33%

1. https://github.com/sonallux/angular-music/pull/107

Image loading [1]

- Adjust image size to render size
- Add `preconnect` instructions
- Use `NgOptimizedImage` directive (>= Angular 15)

Stats	Relative change
Build time	+ 2%
Lines of Code	+ 2%
Initial bundle size	+ 1%

1. https://github.com/sonallux/angular-music/pull/108

Built-in control flow [1]

- Available since Angular 17 in developer preview
- Replaces the existing `NgIf`, `NgFor` and `NgSwitch` Directives

1. https://github.com/sonallux/angular-music/pull/162

Built-in control flow - `@if` block

```html
<h1 *ngIf="isLoggedIn; else loggedOu
  Hello User!
</h1>
<ng-template #loggedOut>
  Please log in!
</ng-template>
```

```html
@if (isLoggedIn) {
  <h1>Hello User!</h1>
} @else {
  <h1>Please log in!</h1>
}
```

Built-in control flow - `@for` block

html

```html
<ul>
  <li *ngFor="let item of items">{{ item.name }}</li>
  <li *ngIf="items.length === 0">There are no items</li>
</ul>
```

html

```html
<ul>
  @for (item of items; track item.name) {
    <li>{{ item.name }}</li>
  } @empty {
    <li>There are no items</li>
  }
</ul>
```

Built-in control flow - `@switch` block

```html
<ng-container [ngSwitch]="orderStatus">
  <span *ngSwitchCase="'PLACED'">
    Order received, order processing started
  </span>
  <span *ngSwitchCase="'SHIPPED'">
    Order shipped
  </span>
  <span *ngSwitchCase="'DELIVERED'">
    Order delivered! Enjoy your purchase
  </span>
  <span *ngSwitchCase="'CANCELED'">
    Order canceled
  </span>
  <span *ngSwitchDefault>
    Invalid order status: {{orderStatus}}
  </span>
</ng-container>
```

```html
@switch (orderStatus) {
  @case ('PLACED') {
    <span>Order received, order processing started</span>
  }
  @case ('SHIPPED') {
    <span>Order shipped</span>
  }
  @case ('DELIVERED') {
    <span>Order delivered! Enjoy your purchase</span>
  }
  @case ('CANCELED') {
    <span>Order canceled</span>
  }
  @default {
    <span>Invalid order status: {{orderStatus}}</span>
  }
}
```

Built-in control flow [1]

Stats	Relative change
Build time	- 2%
Lines of Code	+ 1%
Initial bundle size	+ 6%

1. https://github.com/sonallux/angular-music/pull/162

esbuild [1]

- Switch bundler from webpack to esbuild
- Use `browser-esbuild` as drop-in replacement or `application` builder (>= Angular 17)

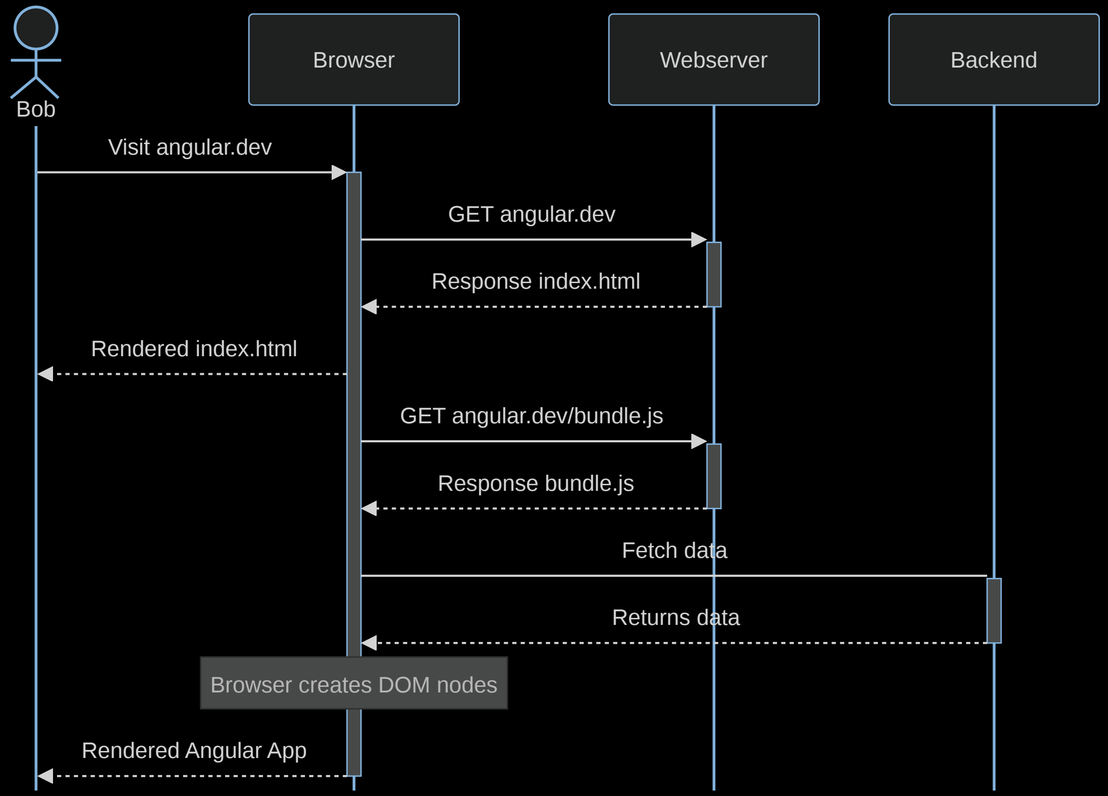
Stats	Relative change
Build time	- 46%
Lines of Code	0%
Initial bundle size	+ 8%

1. https://github.com/sonallux/angular-music/pull/109

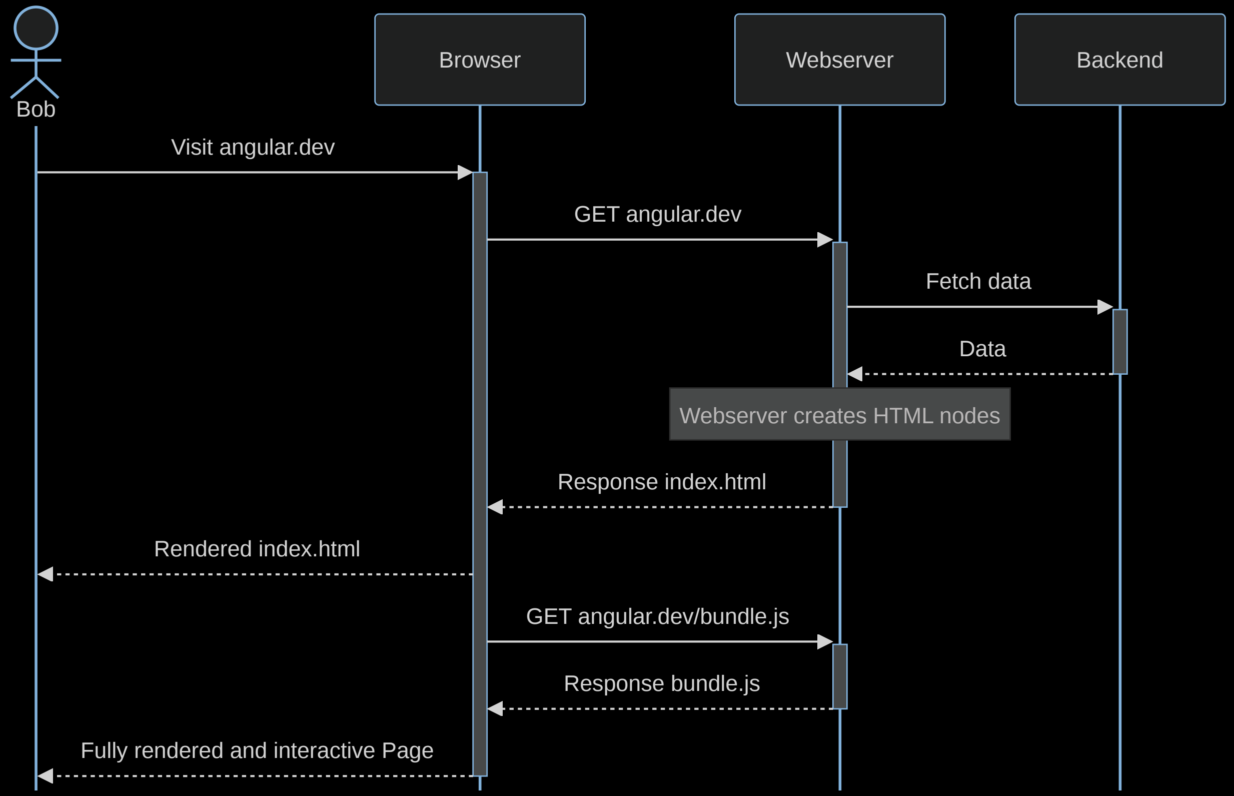
Server-side rendering (SSR) [1]

1. https://github.com/sonallux/angular-music/pull/110

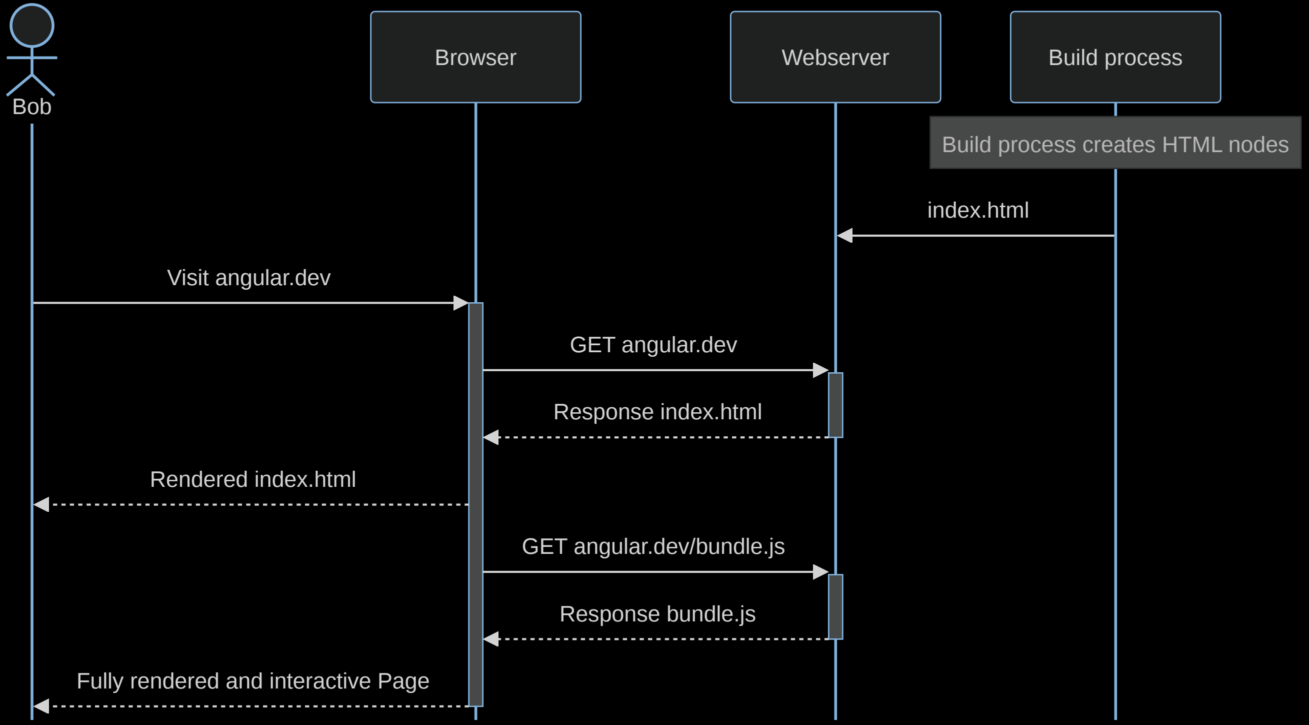
Client-side rendering



Server-side rendering (SSR)



Static Side generation (SSG)



Server-side rendering (SSR) [1]

Stats	Relative change
Build time	+ 55%
Lines of Code	+ 3%
Initial bundle size	+ 2%

1. https://github.com/sonallux/angular-music/pull/110

Lighthouse Score

Page	Baseline	Standalone	Lazy loading	Image loading	Control flow	esbuild	SSR
Home	89	88	87	88	89	89	89
Browse	61	69	63	82	76	76	75
Category	87	81	88	87	87	87	79
Playlist	93	93	93	93	93	93	95
Album	93	93	93	93	93	93	98
Artist	90	91	93	91	92	91	97
Average	86	86	86	89	88	88	89

Next steps

- Unit-Tests (Karma vs Jest vs Web Test Runner)
- OnPush change detection
- Angular signals
- Zoneless change detection

Fragen ?

- Angular Music App
- Core Web Vitals
- Deferrable Views
- NgOptimizedImage
- Built-in control flow
- Server-side rendering
- Hydration
- Prerendering (SSG)
- Angular Movies App

github.com/sonallux @sonallux sonallux.github.io