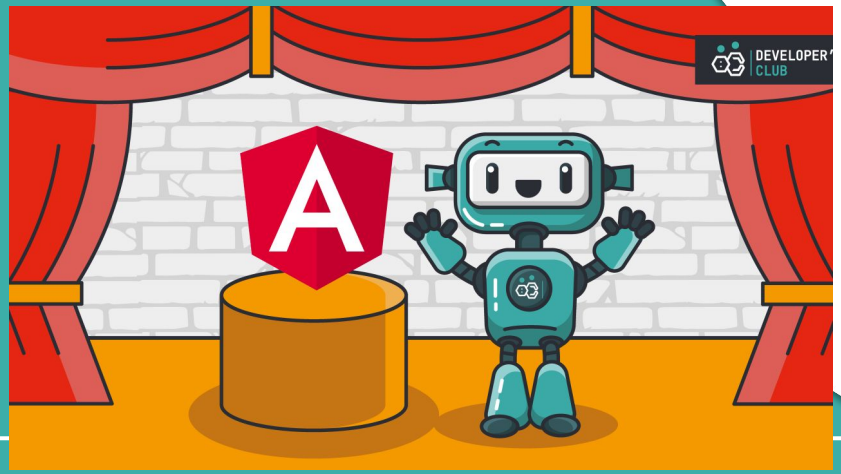
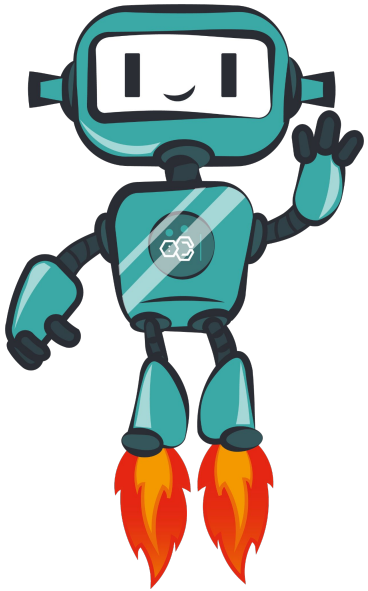
DEVELOPER'S CLUB

BDD#13 http://bit.ly/bdd-13

The Renaissance of Angular: Philosophical Reflections of v17





Cilj predavanja

- Software Craftsmanship prioritet #1
- React postaje Angular, Angular postaje…React?
- Kritički pogled na hvalospev Angularove renesanse
- Diskusija novih stvari
- Benchmarking web aplikacija, JS benchmark projekat

Marcus Vitruvius

DEVELOPER'S CLUB

- Marcus Vitruvius Pollio (/vɪˈtruːviəs ˈpɒlioʊ/; c. 80–70 BC – after c. 15 BC) - Vitruvius, De architectura (On architecture, published as Ten Books on Architecture)

- Ovo znanje je dete prakse i teorije. Praksa je kontinuirano i redovno vežbanje gde se ručni rad obavlja sa bilo kojim potrebnim materijalom prema dizajnu crteža. Teorija, s druge strane, je sposobnost da se demonstrira i objasni produkcija spretnosti na principima proporcije.

React postaje Angular

DEVELOPER'S CLUB

- Redovna tema na Podkastu IT Tipa

redux-observable

DEVELOPER'S CLUB

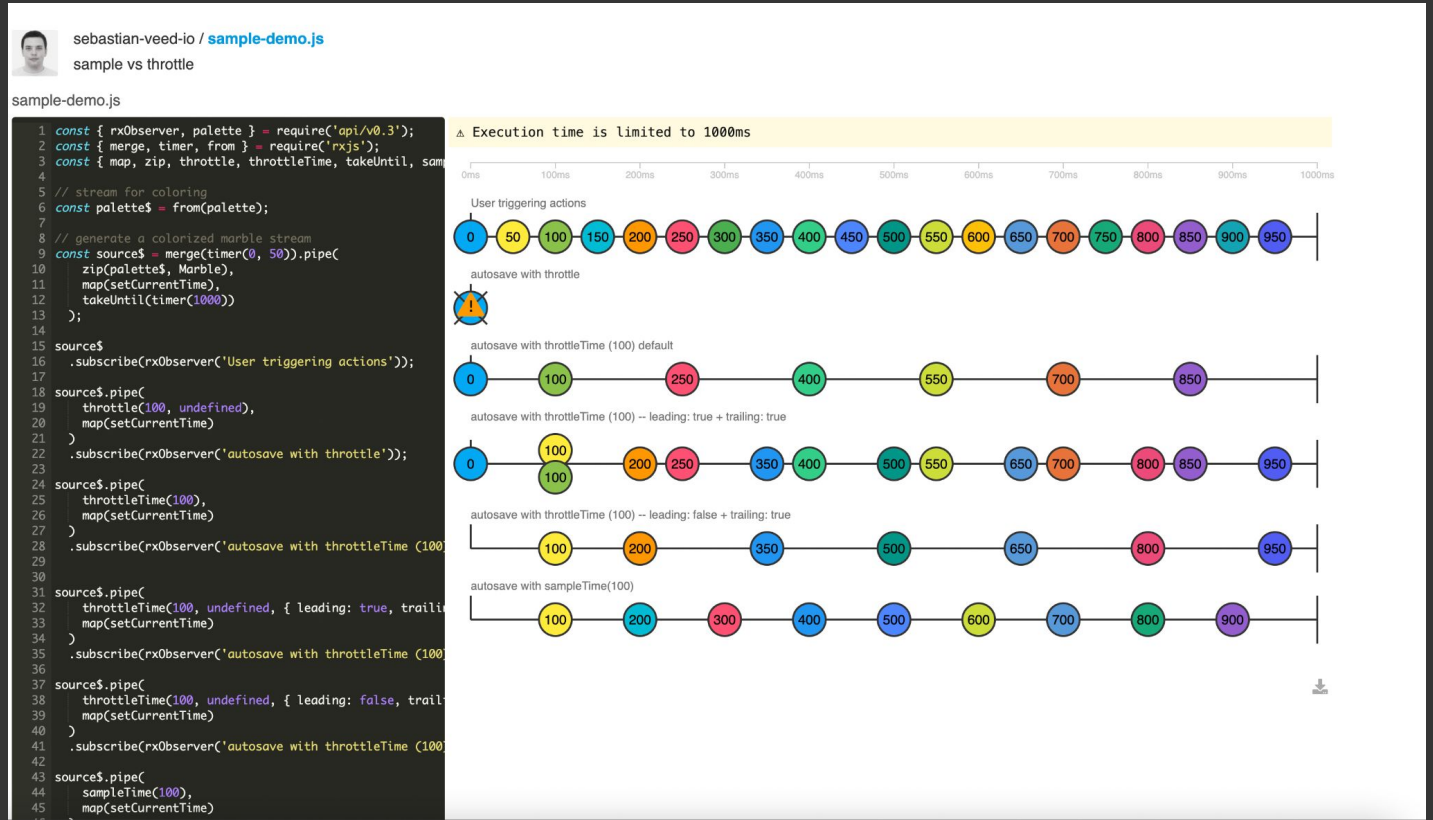
NETFLIX

RxJS + Redux + React = Amazing

Side Effect Management with RxJS

```
const autoCompleteEpic = (action$, store) =>
  action$.ofType('QUERY')
    .debounceTime(500)
    .switchMap(action =>
      ajax('https://api.github.com/search/users?q=' + value)
        .map(payload => ({
          type: 'QUERY_FULFILLED',
          payload
        }))
    );
```

RxJS fun: https://thinkrx.io/



RxJS marble diagram tests

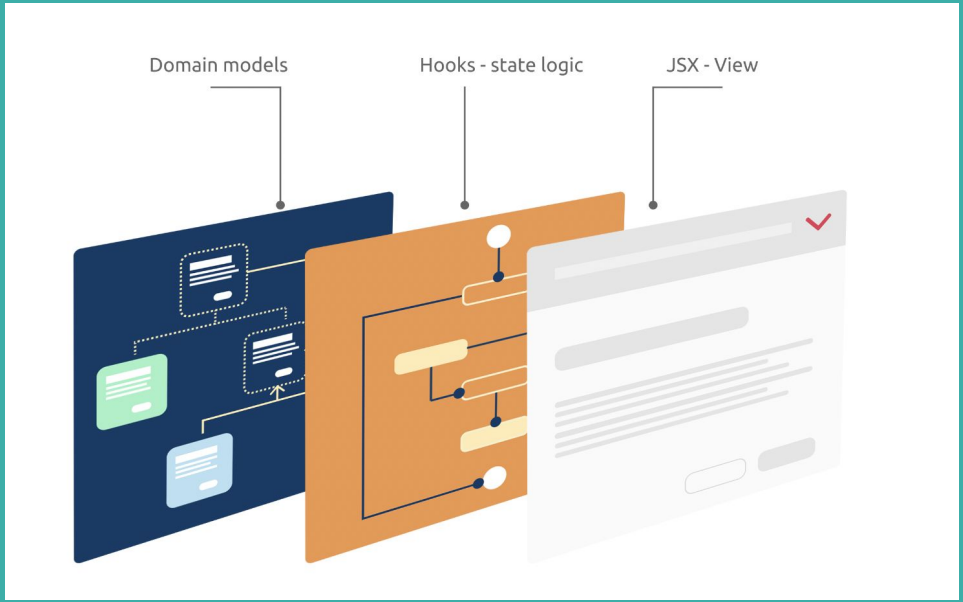
Given a hot source, test multiple subscribers that subscribe at different times:

```
testScheduler.run(({ hot, expectObservable }) => {
  const source = hot('--a--a--a--a--a--a--');
  const sub1 = '      --^-----------!';
  const sub2 = '      ---------^--------!';
  const expect1 = '   --a--a--a--a--';
  const expect2 = '   -----------a--a--a-';

  expectObservable(source, sub1).toBe(expect1);
  expectObservable(source, sub2).toBe(expect2);
});
```

Martin Fowler blog

DEVELOPER'S CLUB

- **Headless components**
  https://martinfowler.com/articles/headless-component.html



Angular postaje React

DEVELOPER'S CLUB

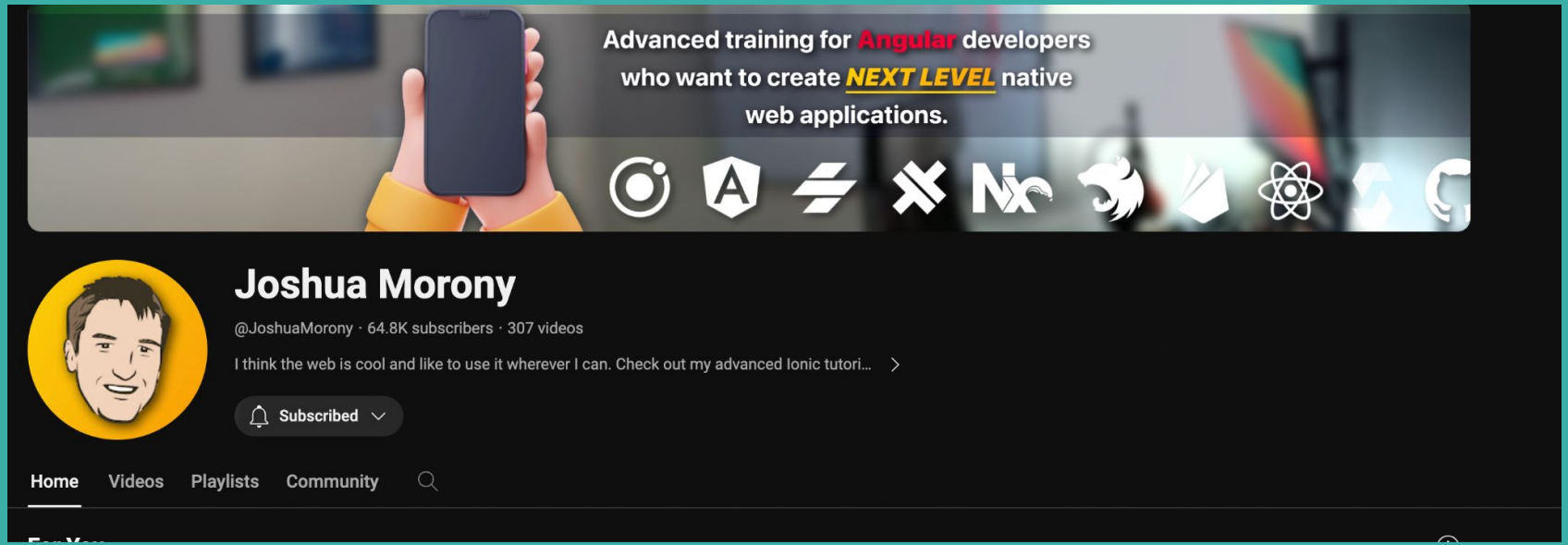
```angular
@Component({
  selector: "counter",
  standalone: true,
  template: `
    <div>
      <p>The count is: {{ count() }}</p>
      <p>The doubled count is: {{ computedDouble() }}</p>
      <button (click)="increment()">+</button>
      @for (item of items; track item) {
        <span>{{ item }}</span>
      }
    </div>
  `,
})
export class CounterComponent {
  items = ['what', 'is', 'this', 'life'] as const;

  count = signal(0);
  computedDouble = computed(() => this.count() * 2);

  increment() {
    this.count.update((value) => value + 1);
  }
}
```

```react
const Counter = () => {
  const [count, setCount] = useState(0);
  const [computedDouble, setComputedDouble] = useState(0);

  const increment = () => {
    setCount(count + 1);
  };

  useEffect( () => {
    setComputedDouble(count * 2);
  }, [count])

  return (
    <div>
      <p>The count is: {count}</p>
      <p>The doubled count is: {computedDouble}</p>
      <button onClick={increment}>+</button>
      {items.map((item, index) => (
        <span key={index}>{item}</span>
      ))}
    </div>
  );
};
```

Angular postaje React

DEVELOPER'S CLUB

➔ https://www.youtube.com/@JoshuaMorony



Angular postaje React

DEVELOPER'S CLUB

```
// sources
private checklistsLoaded$ = this.storageService.loadChecklists();
add$ = new Subject<AddChecklist>();
edit$ = new Subject<EditChecklist>();
remove$ = this.checklistItemService.checklistRemoved$;

// state
private state = signal<ChecklistsState>({
  checklists: [],
  loaded: false,
  error: null,
});

// selectors
checklists = computed(() => this.state().checklists);
loaded = computed(() => this.state().loaded);
error = computed(() => this.state().error);
```

```
constructor() {
  // reducers
  this.checklistsLoaded$.pipe(takeUntilDestroyed()).subscribe({
    next: (checklists) =>
      this.state.update((state) => ({
        ...state,
        checklists,
        loaded: true,
      })),
    error: (err) => this.state.update((state) => ({ ...state, error: err })),
  });

  this.add$.pipe(takeUntilDestroyed()).subscribe((checklist) =>
    this.state.update((state) => ({
      ...state,
      checklists: [...state.checklists, this.addIdToChecklist(checklist)],
    }))
  );
}
```

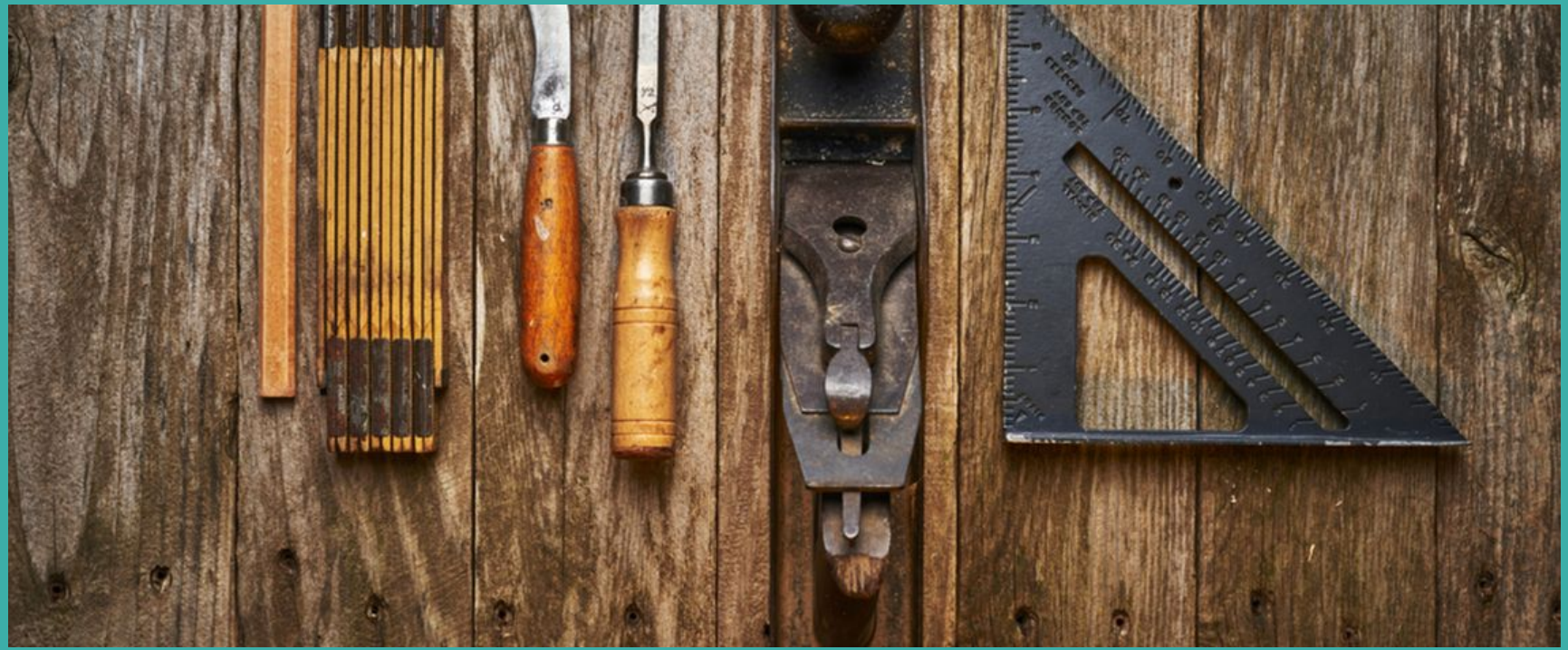
Angular postaje React

DEVELOPER'S CLUB

```
state = signalSlice({
  initialState: this.initialState,
  sources: [this.sources$],
  actionSources: {
    add: (state, action$: Observable<AddChecklist>) =>
      action$.pipe(
        map((checklist) => ({
          checklists: [
            ...state().checklists,
            this.addIdToChecklist(checklist),
          ],
        }))
      ),
  },
});
```

Angular postaje React

DEVELOPER'S CLUB

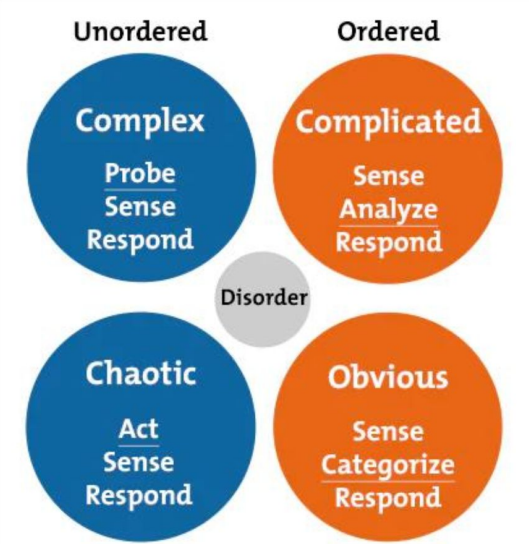


Osvrt na BDD12

DEVELOPER'S CLUB

FRED BROOKS: NO SILVER BULLET, 1986

- **Essential** complexity
  - complexity of **domain**
  - cannot be avoided
  - cannot be changed
- **Accidental** complexity
  - complexity of **implementation**
  - problems engineers create and fix



Special Angular Event

DEVELOPER'S CLUB

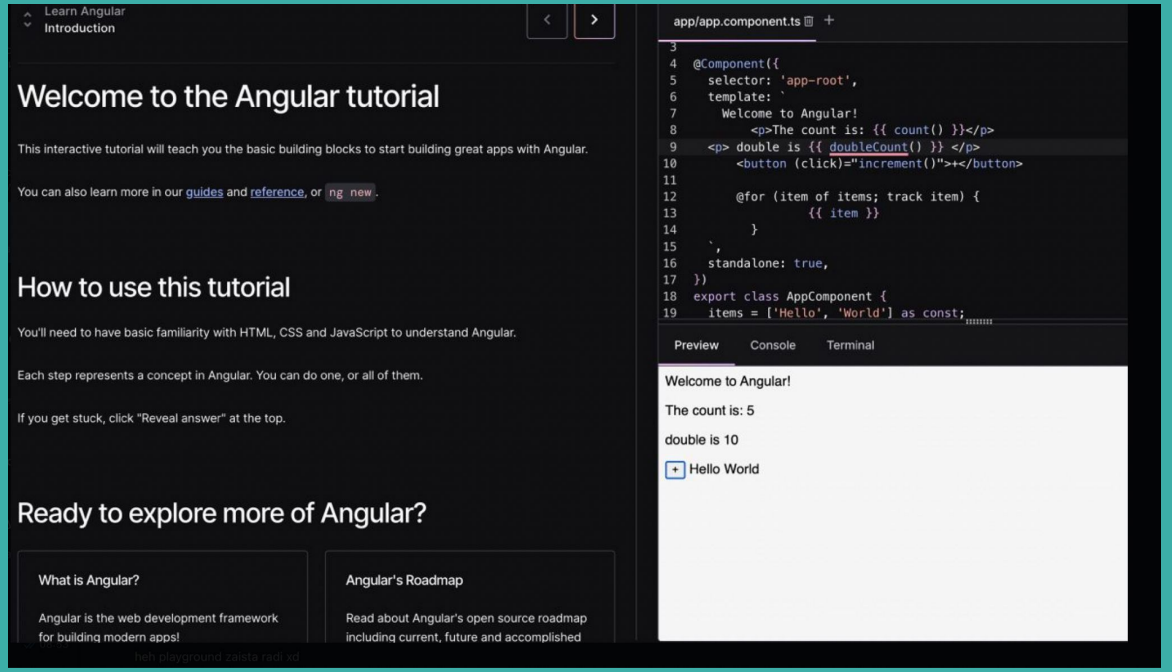
https://www.youtube.com/watch?v=Wq6GpTZ7AX0&ab_channel=Angular

- ➔ DevEx / DX
- ➔ Performance

Special Angular Event

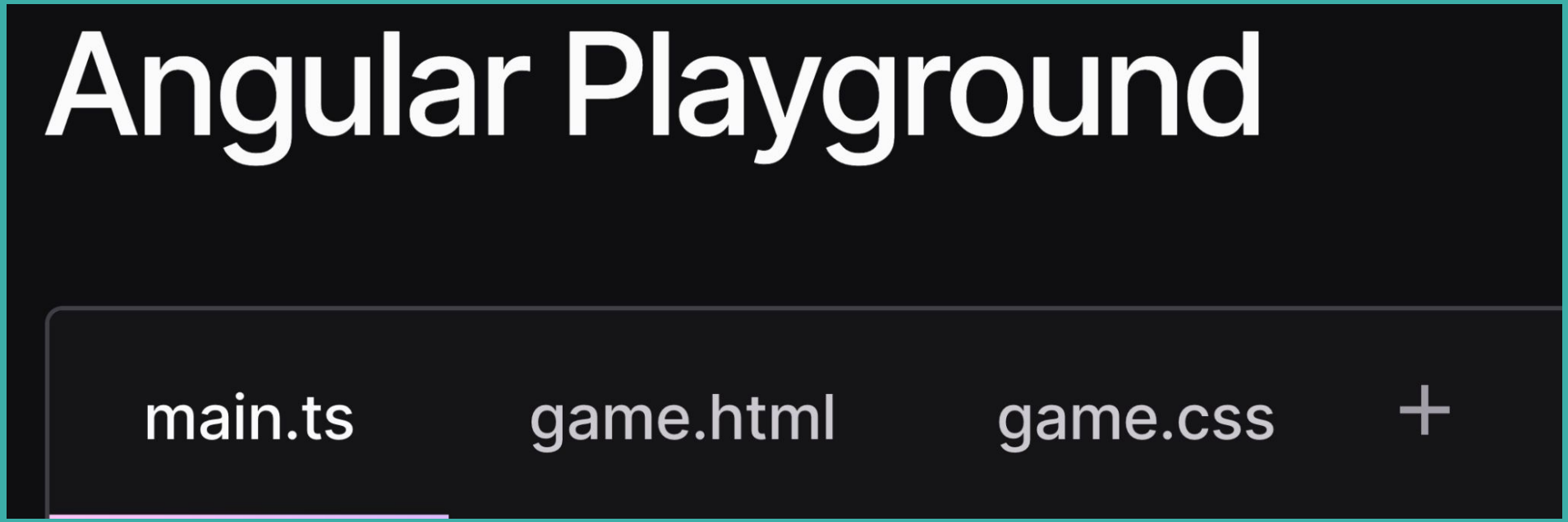
DEVELOPER'S CLUB

➔ On page playground



Special Angular Event

DEVELOPER'S CLUB



Cela Angular aplikacija

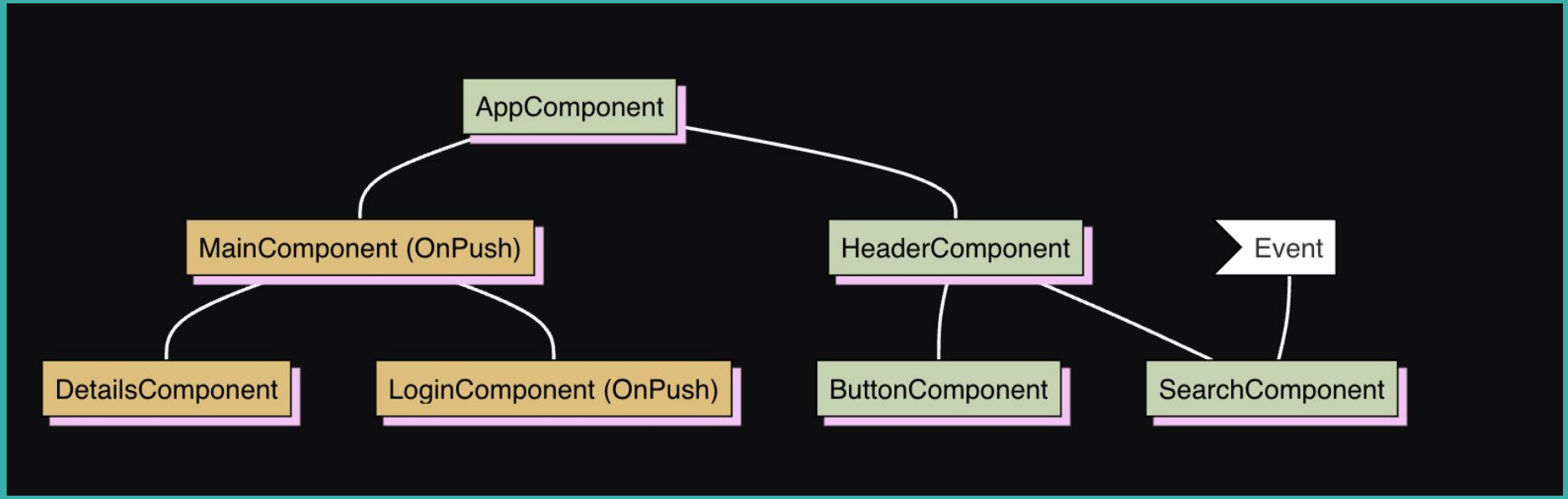
DEVELOPER'S CLUB

Main.ts →

```
@Component({
  selector: "counter",
  standalone: true,
  template: `
    <div>
      <p>The count is: {{ count() }}</p>
      <button (click)="increment()">+</button>
      @for (item of items; let i = $index; track item) {
          <span attr.data-seb="{{i}}">{{ item }}</span>
      }
    </div>
  `,
})
export class CounterComponent {
  items = ['what', 'is', 'this', 'life'] as const;
  count = signal(0);

  increment() {
    this.count.update((value) => value + 1);
  }
}

bootstrapApplication(CounterComponent);
```

Local change detection

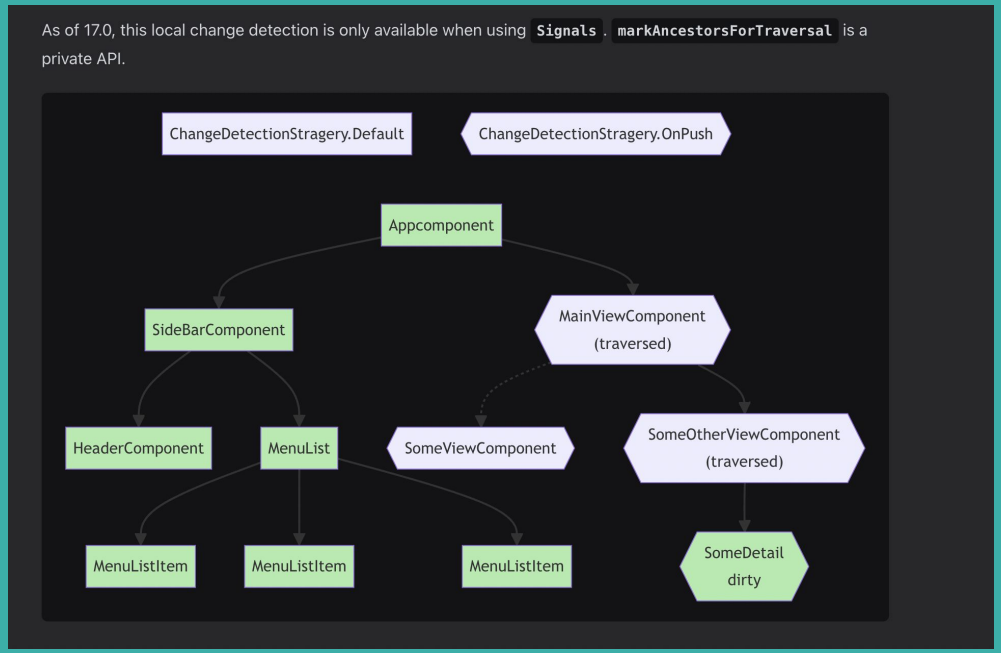
DEVELOPER'S CLUB



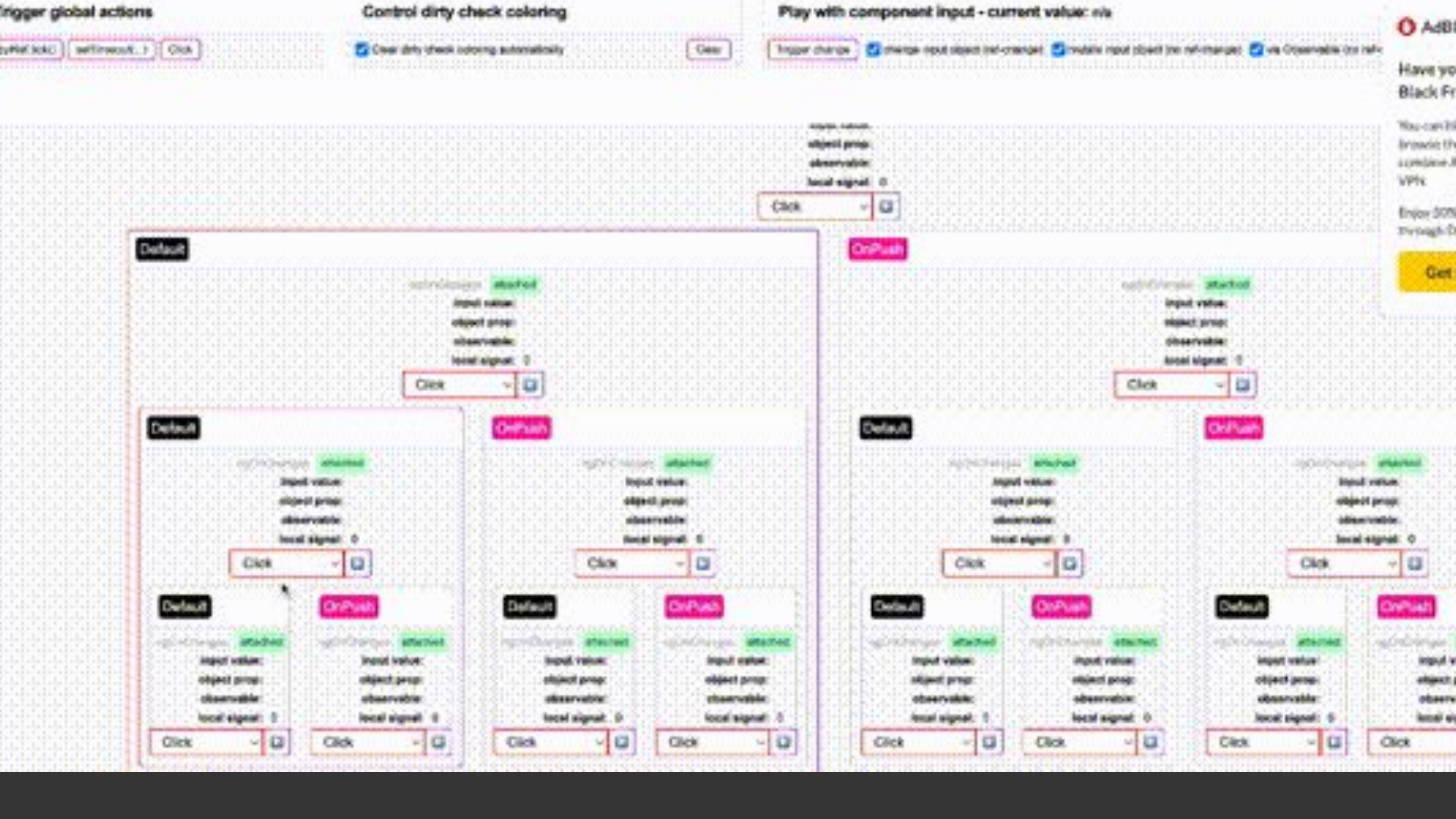
- ➔ https://angular.dev/best-practices/skipping-subtrees
- ➔ Local Change detection almost done

Local change detection

DEVELOPER'S CLUB



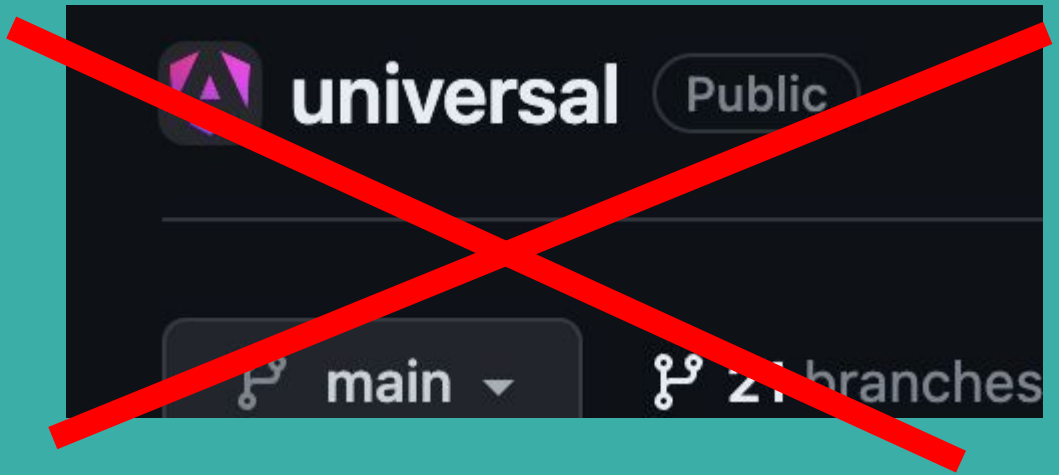
- ➔ https://jeanmeche.github.io/angular-change-detection/
- ➔ markAncestorsForTraversal() markViewDirty()


Trigger global actions
Control dirty check coloring
Play with component input - current value: n/a
Click
Default
OnPush
input value:
object prop:
observable:
local signal: 0
Click

SSR je production ready

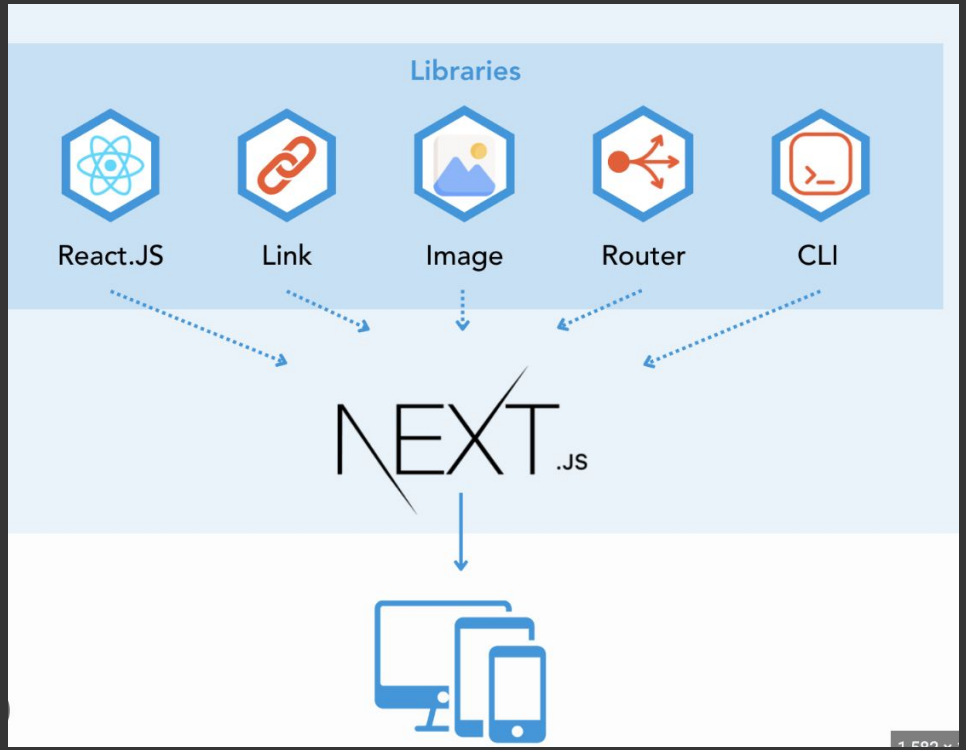
DEVELOPER'S CLUB

- ➔ ng new city-app --ssr
- ➔ pre-rendering + hydration



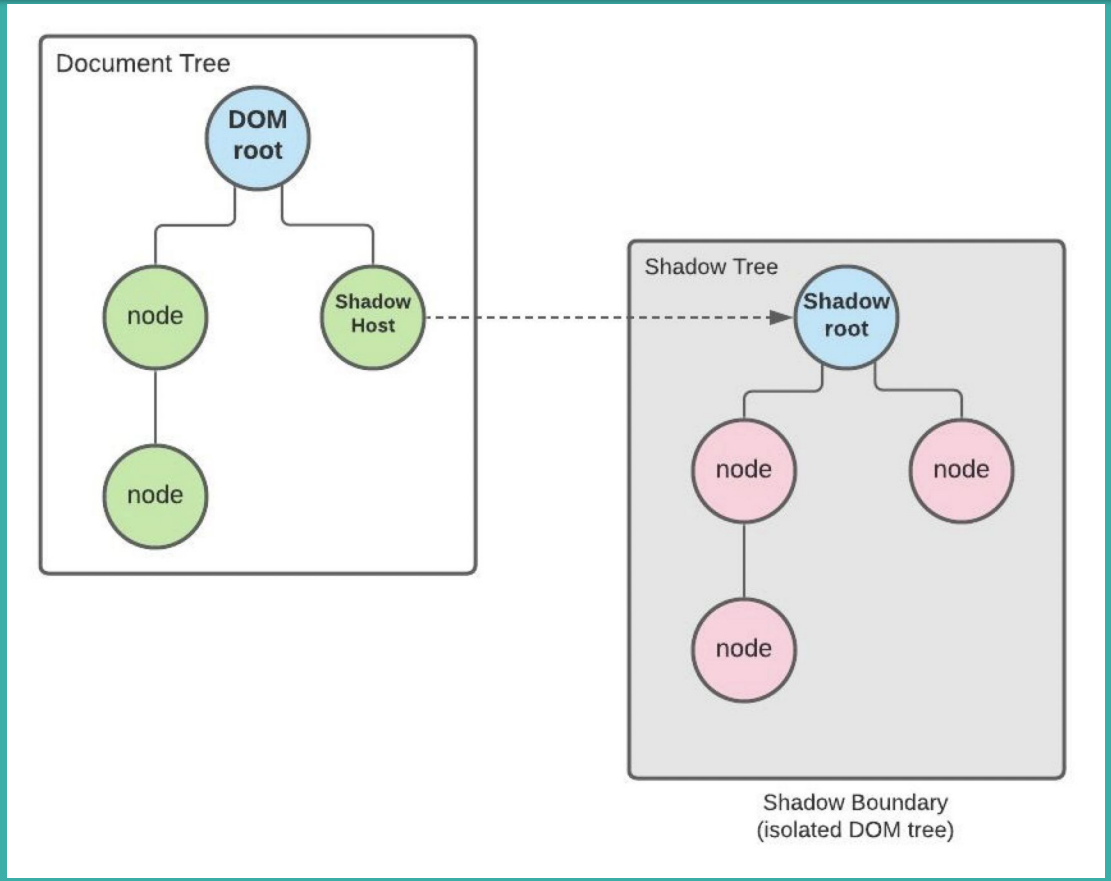

```
import {
  bootstrapApplication,
  provideClientHydration,
} from '@angular/platform-browser';
...

bootstrapApplication(RootCmp, {
  providers: [provideClientHydration()]
});
```

https://web.dev/articles/rendering-on-the-web


Libraries
React.JS
Link
Image
Router
CLI
NEXT.JS

CSS

DEVELOPER'S CLUB



CSS

DEVELOPER'S CLUB

```
styleUrl: ['my-style.css'],
templateUrl: ['my-template.html']
```

```
styleUrl: 'my-style.css',
templateUrl:  'my-template.html'
```

Design tokens - soon

DEVELOPER'S CLUB

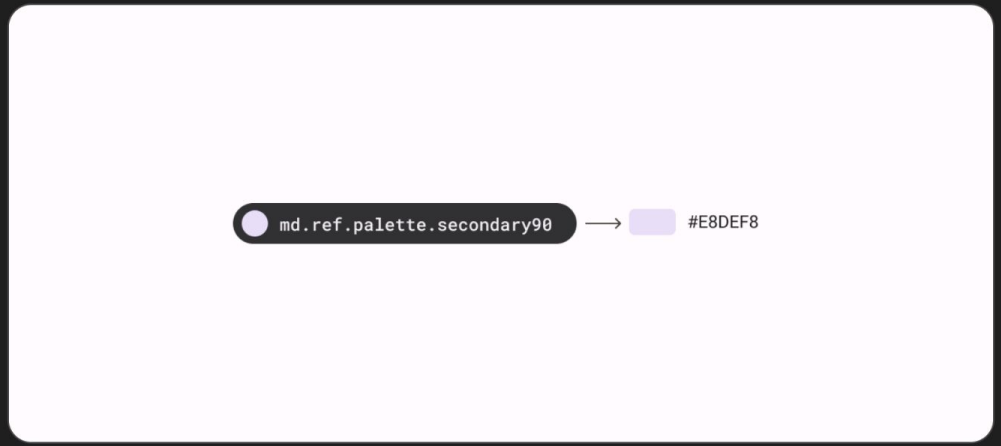
What's a design token?

Design tokens represent the small, repeated design decisions that make up a design system's visual style. Tokens replace static values, such as hex codes for color, with self-explanatory names.

A Material Design token consists of 2 parts:

1. A code-like name, such as **md.ref.palette.secondary90**
2. An associated value, such as **#E8DEF8**

The token's value can be one of several things: A color, a typeface, a measurement, or even another token.



Example of a reference token and its associated color value

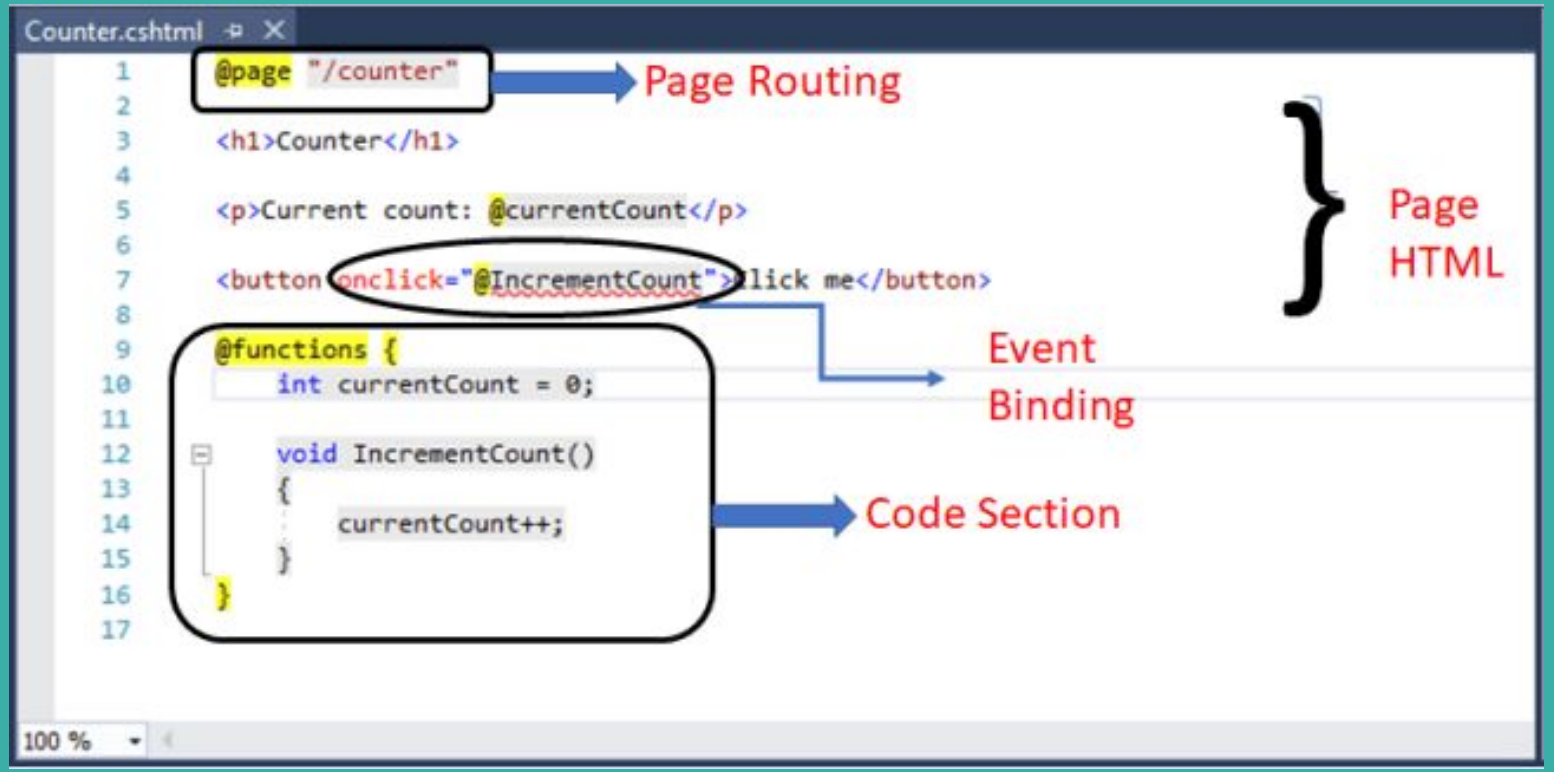
Component Authoring

DEVELOPER'S CLUB

```
@Component({
  selector: "max-pamet",
  standalone: true,
  template: `

      <div *ngIf="pametOK; else pametNijeOk">
          MAX_PAMET
      </div>
      <ng-template #pametNijeOk>
          NEDOVOLJNO_PAMETI
      </ng-template>


      @if (pametOK) {
          MAX_PAMET
      } @else {
          NEDOVOLJNO_PAMETI
      }
      `,
})
```

Component Authoring

DEVELOPER'S CLUB



Component Authoring

DEVELOPER'S CLUB

```
<c:set var="age" scope="request" value="${21}" />

<c:if test="${age >= 18}">
    <p>You are an adult.</p>
</c:if>

<c:if test="${age < 18}">
    <p>You are not an adult.</p>
</c:if>
```

Component Authoring

DEVELOPER'S CLUB

```
<div [ngSwitch]="userType">
    <pro-dashboard *ngSwitchCase="pro"/>
    <basic-dashboard *ngSwitchCase="basic"/>
    <free-dashboard *ngSwitchDefault/>
</div>

@switch (userType) {
    @case ('pro') { <pro-dashboard/> }
    @case ('basic') { <basic-dashboard/> }
    @default { <free-dashboard/> }
```

Component Authoring

DEVELOPER'S CLUB

```
<div *ngFor="let user of users; trackBy: trackById">
    <user-card [user]="user" />
</div>

<div *ngIf="users.length === 0">
    Empty list of users
</div>
```

```
trackById(index: number, user: any): number {
  return user.id;
}
```

Component Authoring

DEVELOPER'S CLUB

```
@for (user of users; track user.id) {
    <user-card [user]="user" />
} @empty {
    <div>Empty list of users </div>
}
```

Defer

DEVELOPER'S CLUB

```
<div #triggerZaLoadMegaKomponente>Ja postojim</div>

<!-- KLIK -->
@defer (when interaction) {
    <moja-mega-komponenta />
}
<!-- trigger druga comp -->
@defer (on viewport(triggerZaLoadMegaKomponente)) {
    <moja-mega-komponenta />
}
@defer (when condition) {
}
@placeholder (minimum 2s) {
<span>Dok se ne klikne ovo stoji</span>
}

@loading (after 200ms; minimum 700ms) {
<span>Kliknuto, loading...</span>
}
```

Defer testing


```
const componentFixture = TestBed.createComponent(ComponentA);
// Retrieve the list of all defer block fixtures and get the first block.
const deferBlockFixture = (await componentFixture.getDeferBlocks())[0];

// Render loading state and verify rendered output.
await deferBlockFixture.render(DeferBlockState.Loading);
expect(componentFixture.nativeElement.innerHTML).toContain('Loading');

// Render final state and verify the output.
await deferBlockFixture.render(DeferBlockState.Complete);
expect(componentFixture.nativeElement.innerHTML).toContain('large works!');
```

Defer pre v17

DEVELOPER'S CLUB

```
class DeferRucnoKomponenta {

  async deferRucno() {
      const { MojaMegaKomponenta } = await import(`./moja-mega-komponenta.component`);
      const factory = this.resolver.resolveComponentFactory(MojaMegaKomponenta);
      this.sadrzajZaNgTemplate = this.vcr.createComponent(factory);
  }
}
```

@Input transform

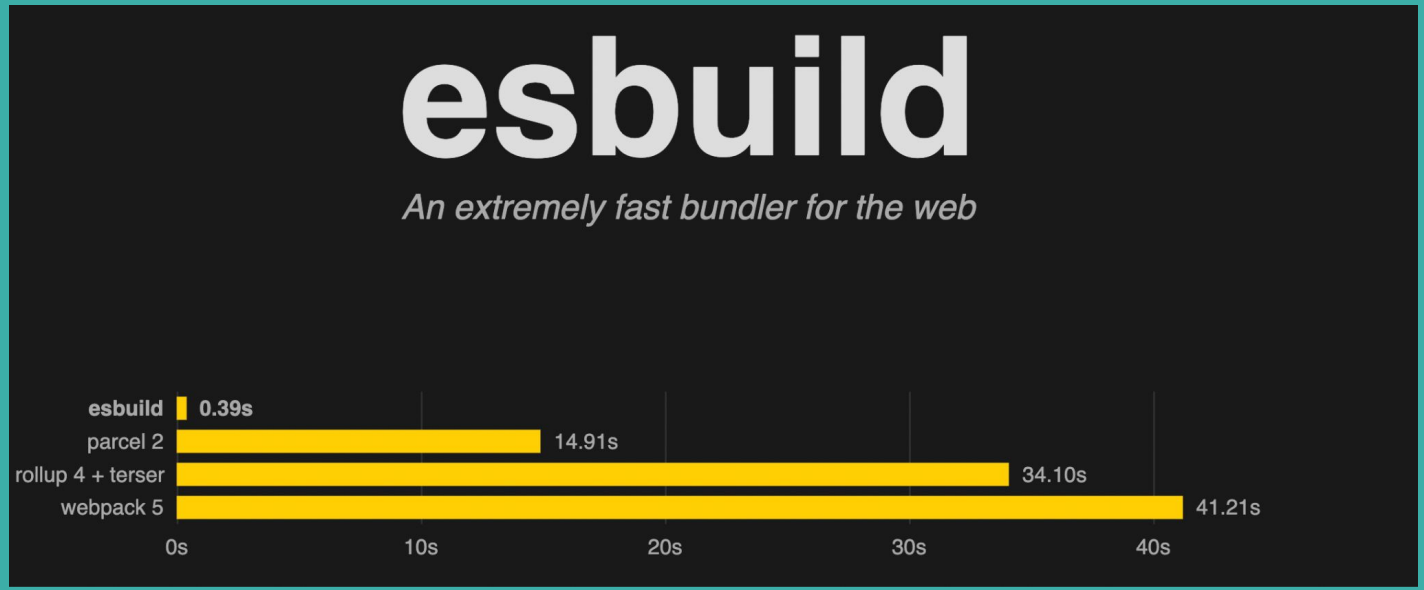
DEVELOPER'S CLUB

```
            <typical-expander expanded />
        `,
})
export class CounterComponent {
 @Input({ transform: booleanAttribute }) expanded: boolean = false;
```

BUILDER improvements

DEVELOPER'S CLUB

- Webpack nije deprecated, za sada



Benchmark

- BLAZING FAST?

Duration None All

- [x] create rows
- [x] replace all rows
- [x] partial update
- [x] select row
- [x] swap rows
- [x] remove row
- [x] create many rows
- [x] append rows to large table
- [x] clear rows

Startup None All

- [x] consistently interactive
- [x] total kilobyte weight

Memory None All

- [x] ready memory
- [x] run memory
- [x] update every 10th row for 1k rows (5 cycles)
- [x] creating/clearing 1k rows (5 cycles)
- [x] run memory 10k

DEVELOPER'S CLUB

Benchmark

DEVELOPER'S CLUB

```
  async init(browser: Browser, page: Page) {
    await checkElementExists(page, "#run");
    for (let i = 0; i < config.WARMUP_COUNT; i++) {
      await clickElement(page, "#run");
      await checkElementContainsText(page, "tbody>tr:nth-of-type(1)>td:nth-of-type(1)", (i*
      await clickElement(page, "#clear");
      await checkElementNotExists(page, "tbody>tr:nth-of-type(1000)>td:nth-of-type(1)");
    }
  }
  async run(browser: Browser, page: Page) {
      await clickElement(page, "#run");
      await checkElementContainsText(page, "tbody>tr:nth-of-type(1000)>td:nth-of-type(1)",
  }
})();
```

Benchmark

DEVELOPER'S CLUB

Name Duration for...	angular-cf-signals-v17.0.0-rc.0	react-hooks-v18.2.0
Implementation notes		
Implementation link	code	code
create rows creating 1,000 rows (5 warmup runs).	46.7 ±0.5 (1.20)	49.6 ±0.5 (1.27)
replace all rows updating all 1,000 rows (5 warmup runs).	52.9 ±0.8 (1.31)	52.3 ±0.5 (1.30)
partial update updating every 10th row for 1,000 rows (3 warmup runs). 4 x CPU slowdown.	22.7 ±0.7 (1.16)	25.1 ±0.8 (1.29)
select row highlighting a selected row. (5 warmup runs). 4 x CPU slowdown.	7.5 ±0.2 (2.44)	6.4 ±0.2 (2.07)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	25.5 ±0.6 (1.10)	161.0 ±1.5 (6.94)

Memory allocation in MBs ± 95%

Name	angular-cf-signals-v17.0.0-rc.0	react-hooks-v18.2.0
ready memory Memory usage after page load.	1.5 (2.85)	1.0 (2.04)
run memory Memory usage after adding 1,000 rows.	5.3 (2.62)	4.7 (2.34)
update every 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	5.4 (2.49)	5.3 (2.43)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	2.2 (3.76)	1.8 (3.08)
run memory 10k Memory usage after adding 10,000 rows.	35.0 (2.47)	34.4 (2.42)
geometric mean of all factors in the table	2.80	2.44

Benchmark

DEVELOPER'S CLUB

Known issues and notes

634 [Issue]: The HTML structure for the implementation is not fully correct.

772 [Note]: Implementation uses manual DOM manipulations

796 [Note]: Implementation uses explicit requestAnimationFrame calls

800 [Note]: View state on the model

801 [Note]: Implementation uses manual event delegation

1139 [Note]: Implementation uses runtime code generation

1261 [Note]: Manual caching of (v)dom nodes

Benchmark

DEVELOPER'S CLUB

```
<table class="table table-hover table-striped test-data">
    <tbody>
        @for (item of data(); track item.id)
        {
        <tr [class.danger]="item.id === selected()">
            <td class="col-md-1">{{item.id}}</td>
            <td class="col-md-4">
                <a href="#" (click)="selected.set(item.id)">{{item.label}}</a>
            </td>
            <td class="col-md-1"><a href="#" (click)="delete(item.id)"><span class="glyphicon glyphicon-remove" aria-hidden="true"></span></a>
            <td class="col-md-6"></td>
        </tr>
        }
    </tbody>
</table>
<span class="preloadicon glyphicon glyphicon-remove" aria-hidden="true"></span>
```

Hvala!
Postani Član:
developersclub.rs