

ANGULAR 2

TYPESCRIPT


~~ПРОЛЕМЫ~~ НА БОЛЬШИХ ПРОЕКТАХ.



Education
Academy




DMITRY PODDUBNIY



Dmitry Poddubnyy
mmd7

Ukraine · [@dmitry](#)
[mmd7@yandex.ru](#)
[https://mmd7.ru](#)
 (joined on Mar 5, 2011)

Organizations



Put requests issues Gist

Overview Repositories 40 50k+ 13k Followers 34 Following 25

Pinned repositories

1. representation-kit
Open source kit for people who writes meta code

[JavaScript](#) [41](#)

2. Ghost
Based from Typo3 CMS

A simple powerful publishing platform

[JavaScript](#)

3. ui-select
Ported from angular-ui-select

AngularJS native version of Select2 and Selectize

[JavaScript](#)

Customize your pinned repositories

4. make-contextual-autoscroll
Complete solution for auto scroll in make contextual

[JavaScript](#) [5](#) [Y2](#)

5. react-redux-kit
react-redux-kit


[JavaScript](#)

6. react.js
Ported from payment.js

A cross-platform framework for the universal server-rendered React applications

[JavaScript](#)

1,677 contributions in the last year



Legend: ■ push ■ pull request ■ no contribution

1,097

CONTRIBUTIONS

REACT BASED PRESENTATION.

```
return (  
  <Deck transition={['zoom','slide']} transitionDuration={800}>  
    <Slide bgColor="primary">  
      <Heading size={1} fit caps>  
        React Presentations  
      </Heading>  
      <Heading size={2} fit caps>  
        Written In React  
      </Heading>  
    </Slide>  
    <Slide bgColor="black">  
      <Heading size={1} fit textColor="primary" textFont="secondary">  
        Wait What?  
      </Heading>  
    </Slide>  
    <Slide bgColor="primary" textColor="black" align="center top">  
      <Heading size={1} textColor="black" textFont="primary">  
        Thats right  
      </Heading>  
      <List>  
        <ListItem>Inline style based theme system</ListItem>  
        <ListItem>Autofit Text</ListItem>  
        <ListItem>react-router navigation</ListItem>  
        <ListItem>PDF Export</ListItem>  
      </List>  
    </Slide>  
  </Deck>  
)
```

КАК ЖЕ МЫ РЕШИЛИСЬ НА ANGULAR 2

2К17

ХОТЯ ЕСТЬ REACT ?

Реакт это jQuery 2007?

- «и да и нет.»

ЧТО ЖЕ ДАЛ JQUERY КОГДА ПОЯВИЛСЯ :

- Удобство
- Скорость разработки
- Универсальность

МАГИЯ КОТОРАЯ ПОКОРИЛА НАС:

```
$(document).ready(function(){  
  
    $(".input-prepend > label, .swTableFilters-search +").css("color", "red");  
  
    $(".input-prepend").click(function(){  
        $(this).fadeOut("fast");  
    });  
  
})
```

| Angular 2, Поехали!

- «*Так говорил Гагарин, а его все любят правда ?*»

constructor

ngOnChanges

ngOnInit

ngDoCheck

ngAfterContentInit

ngAfterContentChecked

ngAfterViewInit

ngAfterViewChecked

ngOnDestroy

A component has a lifecycle managed by Angular.

Angular creates it, renders it, creates and renders its children, checks it when its data-bound properties change, and destroys it before removing it from the DOM.

Angular offers **lifecycle hooks** that provide visibility into these key life moments and the ability to act when they occur.

A directive has the same set of lifecycle hooks, minus the hooks that are specific to component content and views.

ANGULAR 2:

- Angular 1.x codebase
- Animations
- Universal
- Tests E2E
- Static Types
- C# ==* TypeScript

Keep it simple, stupid

- «делай это проще, дурачок»






Codebase that smells

- «хорошая кодовая база»

Select a person in the list or search his name/age... ▲



- Adam**
email: adam@email.com age: 12
- Amalie
email: amalie@email.com age: 12
- Estefanía
email: estefania@email.com age: 21
- Adrian
email: adrian@email.com age: 21
- Vladimir

Втроде обычный компонет!?

- «а вот и нет»

```
<!-- end ngRepeat: person in $select.items -->
<!-- ngIf: $select.open -->
▶<li role="option" ng-attr-id="ui-select-choices-row-
$select.generatedId }-{{$index}}" class="ui-select-ch
ng-class="{ 'select2-highlighted': $select.isActive(thi
disabled : $select.isDisabled(this))" ng-repeat="persc
ng-if="$select.open" ng-click=
"$select.select(person,$select.skipFocusser,$event)" i
choices-row-1-1">...</li>
<!-- end ngIf: $select.open -->
<!-- end ngRepeat: person in $select.items -->
<!-- ngIf: $select.open -->
▶<li role="option" ng-attr-id="ui-select-choices-row-
$select.generatedId }-{{$index}}" class="ui-select-ch
ng-class="{ 'select2-highlighted': $select.isActive(thi
disabled : $select.isDisabled(this))" ng-repeat="persc
ng-if="$select.open" ng-click=
"$select.select(person,$select.skipFocusser,$event)" i
choices-row-1-2">...</li>
```

```
40     choices.attr('ng-repeat', parserResult.repeatExpression(groupByExp));
41
42     if (showWhen) choices.attr('ng-show', showWhen); // added for speed improves on
43
44     //     .attr('ng-if', '$select.open'); //Prevent unnecessary watches when drop
45
```



ЕЩЕ ТИПИЧНАЯ ПРОБЛЕМА?

```
// production ready code
// some where ..

$filter('i18n')(label) + (item.count ? ' (' + $filter('number')(item.count) + ')' : '');

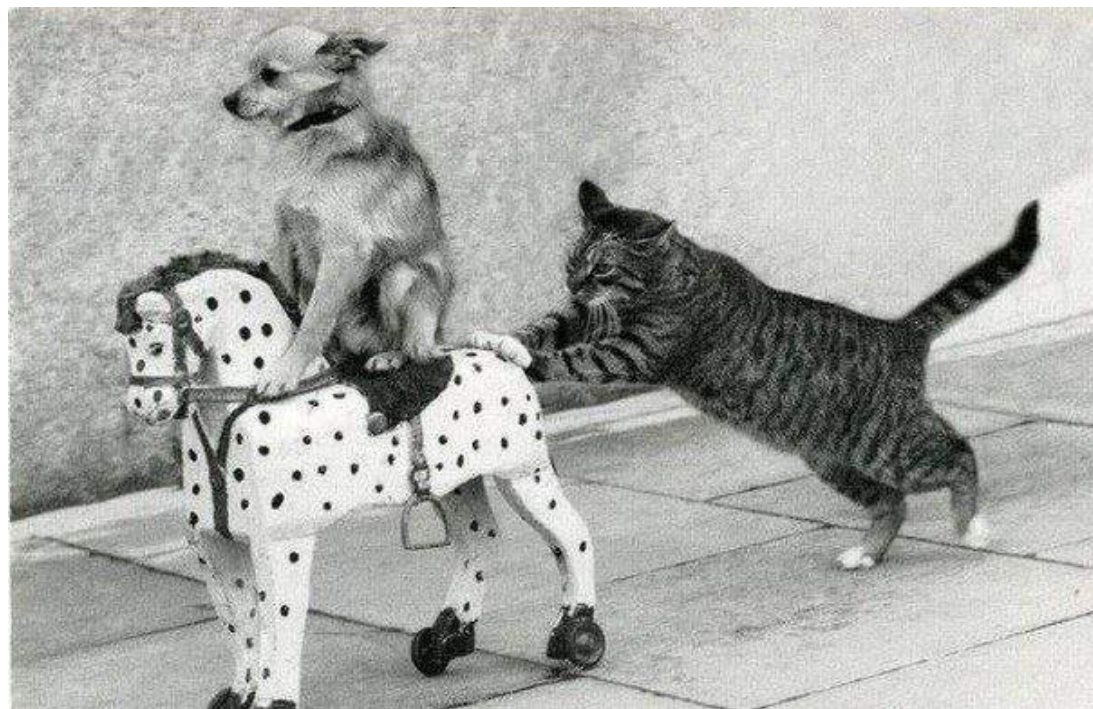
// other ready code
```



```
function findNestedItem(items, itemId) {
  var childResult = null;
  var returnedValue = _.find(items, function (item) {
    if (item.id == itemId) return true;
    else if (item.children && item.children.length) {
      childResult = findNestedItem(item.children, itemId);
      return !_.isEmpty(childResult);
    }
    else return false;
  });
  return childResult || returnedValue;
}
```


One component to rule them all!

- «Один компонент чтобы править ими всеми!»



Так ли просто на Angular 2

- « - нет.»

НЕ В REDUX'Е ДЕЛО!

```
import { Component } from '@angular/core';
import { NgRedux, select } from '@angular-redux/store';
import { CounterActions } from '../actions/counter-actions';
import { RandomNumberService } from '../services/random-number';

@Component({
  selector: 'counter',
  providers: [ CounterActions, RandomNumberService ],
  template: `
    <p>
      Clicked: {{ counter$ | async }} times
      <button (click)="actions.increment()">+</button>
      <button (click)="actions.decrement()">-</button>
      <button (click)="actions.incrementIfOdd()">Increment if odd</button>
      <button (click)="actions.incrementAsync(2222)">Increment async</button>
      <button (click)="actions.randomize()">Set to random number</button>
    </p>
  `
})
export class Counter {
  @select('counter') counter$: any;

  constructor(private actions: CounterActions) {}
}
```

```
// imports etc ...

@Component({
  selector: 'app-common-popup',
  templateUrl: './common-popup.component.html',
  styleUrls: ['./common-popup.component.scss']
})
export class CommonPopupComponent implements AfterViewInit {

  // ...

  constructor(private sanitizer: DomSanitizer, private comPopupService: CommonPopupService) {}

  ngAfterViewInit(): void {
    this.comPopupService.pushData.subscribe( () => {
      const {title, content, footer, disabledSubmit } = this.comPopupService;
      this.title = title;
      this.content = this.sanitizer.bypassSecurityTrustHtml(content);
      this.footer = footer;
      this.disabledSubmit = disabledSubmit;
      this.showPopup = true;
    });
  }
}
```

```
import {Directive, HostListener, ElementRef} from '@angular/core';
import {window} from '@angular/platform-browser/src/facade/browser';

@Directive({
  selector: '[appCommonVerticalAlign]'
})
export class CommonVerticalSetWindowHeightDirective {

  @HostListener('window:resize', ['$event'])
  onResize(event) {
    const result = event.target.innerHeight;
    this.elementRef.nativeElement.style.height = result + 'px';
  }

  constructor( private elementRef: ElementRef) {
    this.elementRef.nativeElement.style.height = window.innerHeight + 'px';
  }
}
```

```
import { Directive, HostListener, ElementRef } from '@angular/core';
import { window } from '@angular/platform-browser/src/facade/browser';

@Directive({
  selector: '[appCommonVerticalAlign]'
})
export class AdaptiveContainer {

  @HostListener('window:resize', ['$event'])
  @debounce(100)
  onResize(event: any) {
    const result = (event && event.target.innerHeight || 0);
    this.elementRef.nativeElement.style.height = `${result}px`;
  }

  constructor( private elementRef: ElementRef) {
    this.elementRef.nativeElement.style.height = (window.innerHeight && `${window.innerHeight}px`);
  }
}
```

```
import { Directive, HostListener, ElementRef } from '@angular/core';
import { window } from '@angular/platform-browser/src/facade/browser';

@Directive({
  selector: '[appCommonVerticalAlign]'
})
export class AdaptiveContainer {

  ngAfterViewInit(private ngzone: NgZone) {
    this.ngzone.runOutsideAngular( () => {
      Observable.fromEvent(window, 'resize')
        .throttleTime(200)
        .subscribe(e => {
          const result = (event && event.target.innerHeight || 0);
          this.elementRef.nativeElement.style.height = `${result}px`;
        });
    });
  }

  constructor( private elementRef: ElementRef) {
    this.elementRef.nativeElement.style.height = (window.innerHeight && `${window.innerHeight}px`);
  }
}
```


| Как ВЫГЛЯДИТ

- разработка на Angular 2?



```
import { Component } from '@angular/core';
import { NgRedux, select } from '@angular-redux/store';
import { CounterActions } from '../actions/counter-actions';
import { RandomNumberService } from '../services/random-number';

@Component({
  selector: 'counter',
  providers: [ CounterActions, RandomNumberService ],
  template: `
    <p>
      Clicked: {{ counter$ | async }} times
      <button (click)="actions.increment()">+</button>
      <button (click)="actions.decrement()">-</button>
      <button (click)="actions.incrementIfOdd()">Increment if odd</button>
      <button (click)="actions.incrementAsync(2222)">Increment async</button>
      <button (click)="actions.randomize()">Set to random number</button>
    </p>
  `
})
export class Counter {
  @select('counter') counter$: any;

  constructor(private actions: CounterActions) {}
}
```

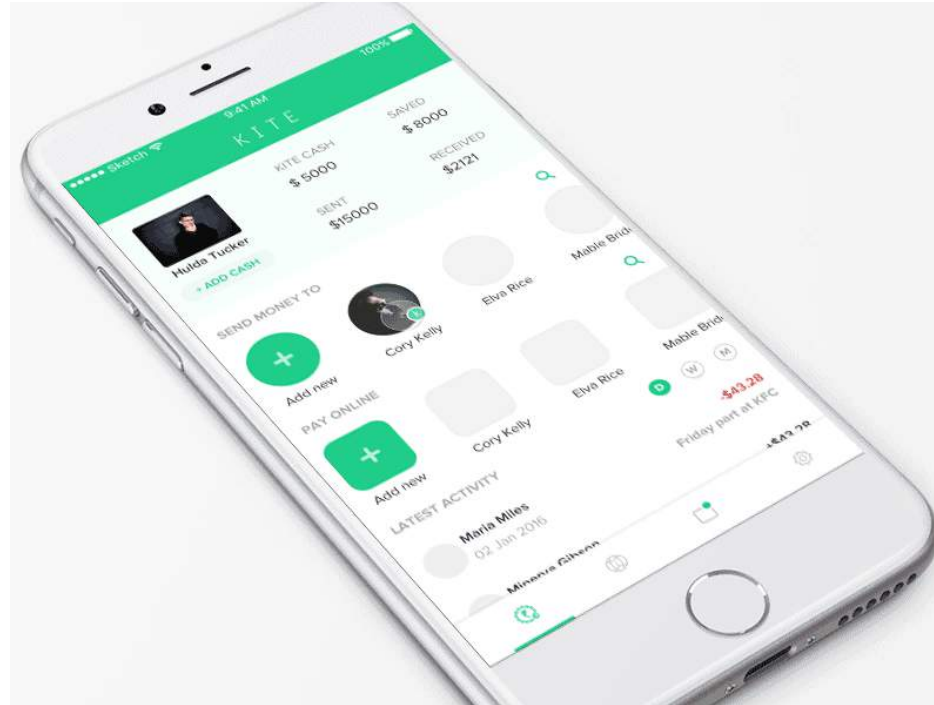
```

var __decorate = (this && this.__decorate) || function (decorators, target, key, desc) {
    var c = arguments.length, r = c < 3 ? target : desc === null ? desc = Object.getOwnPropertyDescriptor(target, key) : desc, d;
    if (typeof Reflect === "object" && typeof Reflect.decorate === "function") r = Reflect.decorate(decorators, target, key, desc);
    else for (var i = decorators.length - 1; i >= 0; i--) if (d = decorators[i]) r = (c < 3 ? d(r) : c > 3 ? d(target, key, r) :
    return c > 3 && r && Object.defineProperty(target, key, r), r;
};
define(["require", "exports", "@angular/core", "@angular-redux/store", "../actions/counter-actions", "../services/random-number"]
"use strict";
Object.defineProperty(exports, "__esModule", { value: true });
var Counter = (function () {
    function Counter(actions) {
        this.actions = actions;
    }
    return Counter;
}());
__decorate([
    store_1.select('counter')
], Counter.prototype, "counter$", void 0);
Counter = __decorate([
    core_1.Component({
        selector: 'counter',
        providers: [counter_actions_1.CounterActions, random_number_1.RandomNumberService],
        template: "\n <p>\n    Clicked: {{ counter$ | async }} times\n    <button (click)=\"actions.increment()\">+</button>"
    })
], Counter);
exports.Counter = Counter;
});

```

**UI/UX become is more
and more important**

- «UI / UX становятся все более и более важным»



The background of the image is a dynamic scene of The Flash in his red and yellow suit, running through a heavy rain. The character is shown from the chest up, angled towards the right, with his head turned slightly back. His suit features the iconic lightning bolt emblem on the chest and a circular logo on the shoulder. The rain is depicted as numerous white streaks, creating a sense of high speed and motion. The overall color palette is dominated by the red of the suit and the yellow of the lightning, set against a dark, rainy background.

**I am the faster man
alive.**

- «*The Flash*»



Performance grade [?]

A 97

Load time

1.50 s

Faster than

81 %
of tested sites

Page size

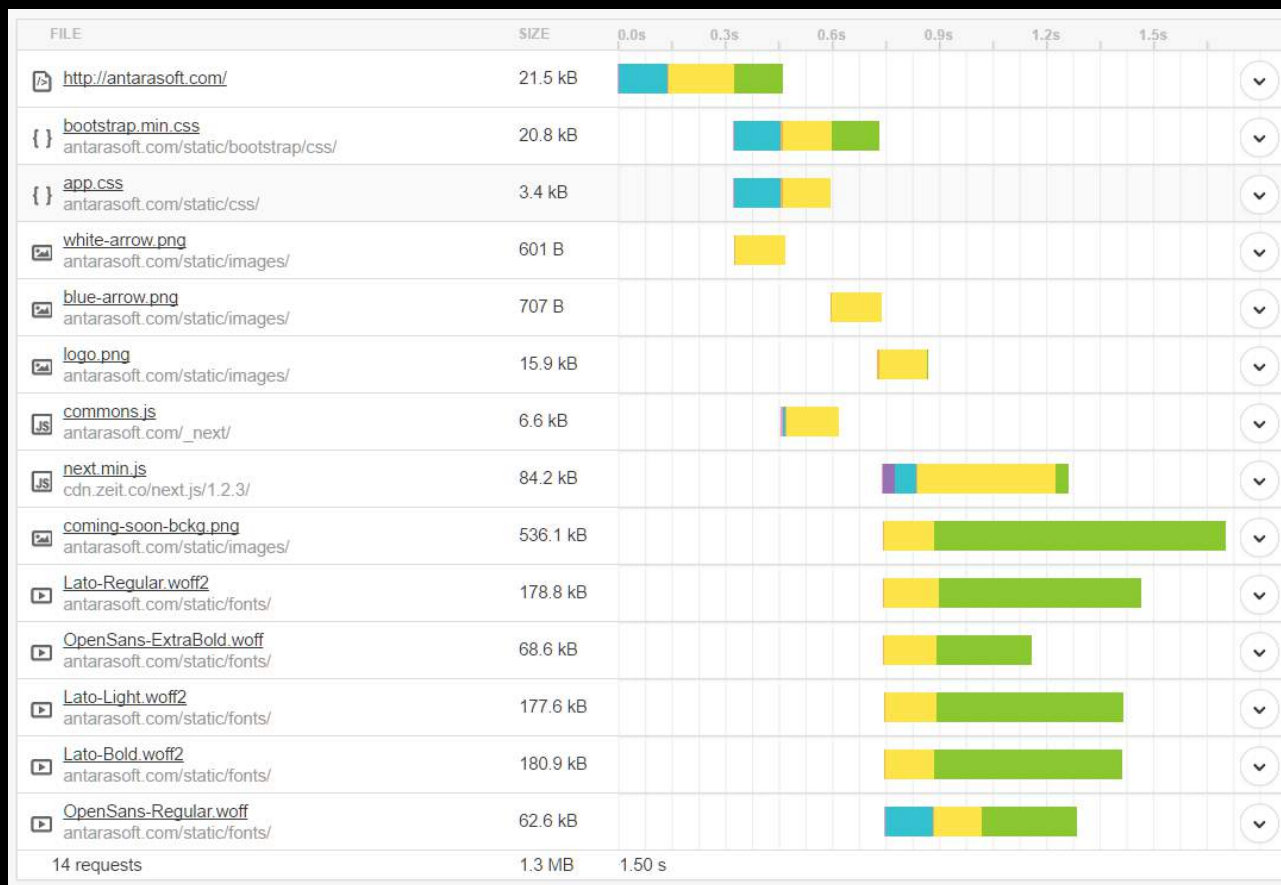
1.3 MB

Requests

14

Tested from

 Stockholm
on Dec 17 at 09:47



NATIVESCRIPT
REACT NATIVE

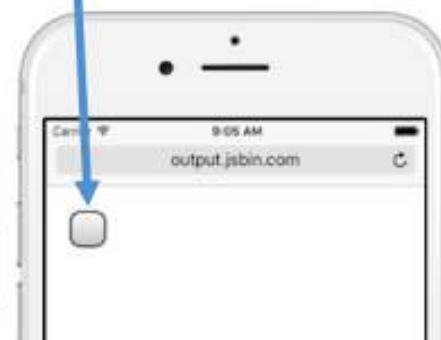
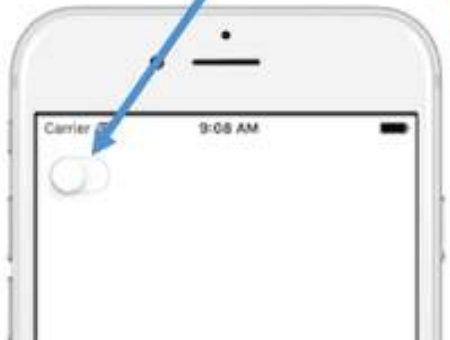


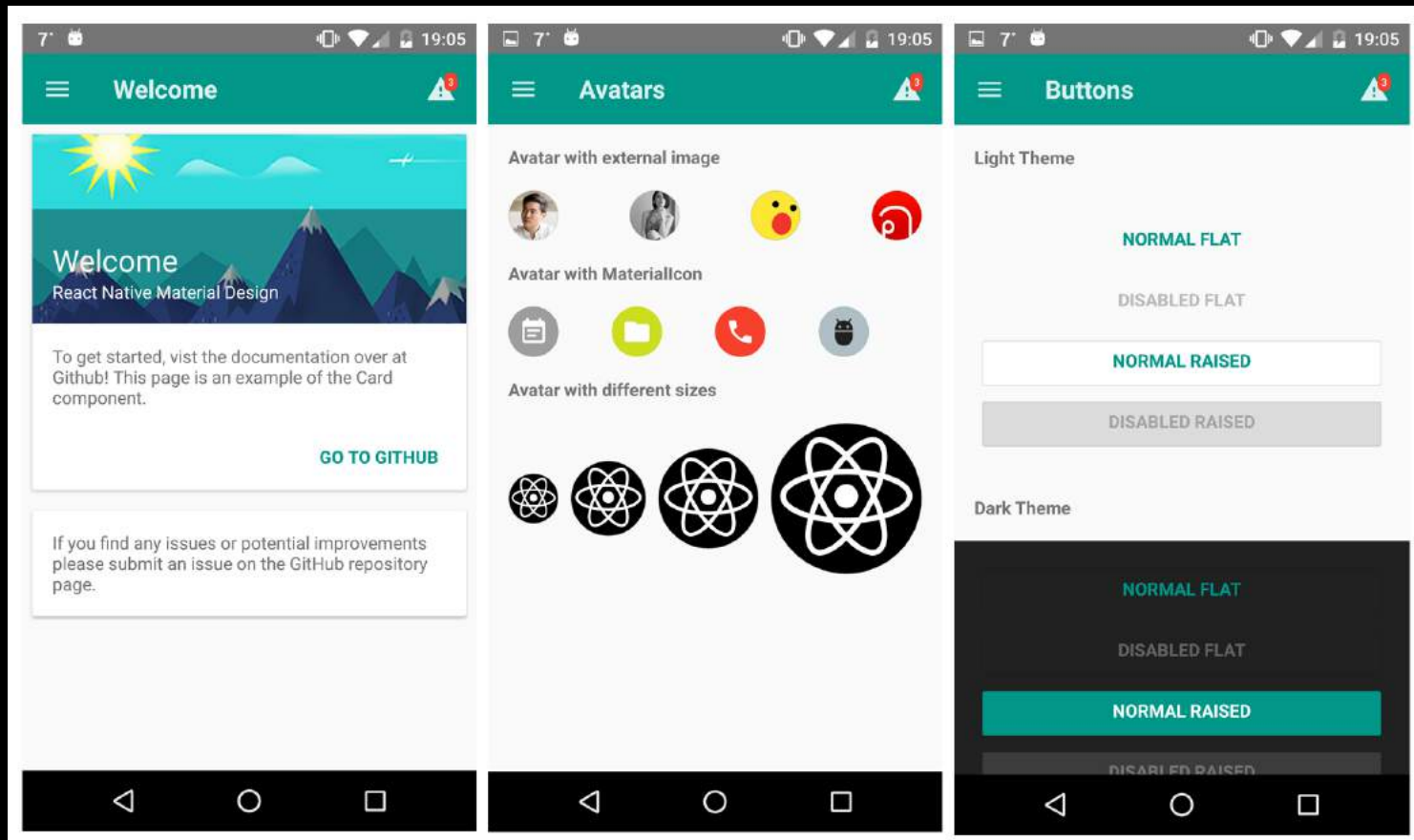
```
@Component({  
  selector: "checkbox"  
  templateUrl: "checkbox.html"  
});
```



<switch>

<input type="checkbox">







**Modern technology is
magic.**

- «Современные технологии похожи на магию.»

- Следите за трендами, а не следуйте
- Изучайте больше
- Отдыхайте
- Курсы хороший способ начать
- KISS



Как выглядит типичный проект

- на *angular 2* конечно!





Education
Academy

- Presentation on Github

<https://github.com/mr47/itea-presentation-kiev>

- Github @ mr47

<https://github.com/mr47>

УВИДИМСЯ В БУДУЩЕМ!

