

Angular

Ваши спикеры:
Федулов Антон и Сарапульцева Дарья

PARMA
technologies group



Что такое Angular?

Angular

Это JavaScript-фреймворк с открытым исходным кодом, разрабатываемый командой из Google и открытым сообществом разработчиков



История

История

- ⊗ Первая версия вышла в 2009 году, а в 2012 году вышел AngularJS v1.0.0
- ⊗ AngularJS !== Angular
- ⊗ Angular 8
- ⊗ Ныне Angular 11.0.0



Плюсы и фишки Angular

Поддержка



- ⊗ Поддерживается Microsoft и Google
- ⊗ Большое коммьюнити
- ⊗ Конференции, гайды, статьи

Watch ▾

3.2k

Unstar

67,932

Fork

18k

obenjro 30 апреля 2020 в 14:18

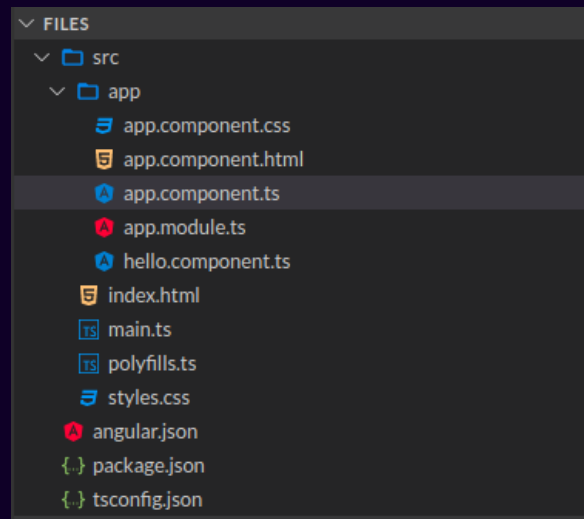
Русскоязычная документация по Angular

Страница на 19k просмотров

файлов без потери форматирования
и/или: кастомного изобрета и поиска из двух видов: «список» позволяющий увидеть название документа
это бесплатно в течение недели, нескольких бесплатных дней и/или с оплатой нескольких месяцев

Единая структура проекта

- ⊗ Удобная иерархия файлов
- ⊗ Разделение кода
- ⊗ Разнообразие структурных единиц



Структурные единицы

- ⊗ Модули
- ⊗ Компоненты
- ⊗ Директивы
- ⊗ Фильтры
- ⊗ Сервисы

```
<div *ngIf="condition">Content to render when condition is true.</div>
```

Пример структурной директивы

```
@Component({  
  selector: 'date-pipe-example',  
  template: `  
    <p>  
      Transformed date:  
      {{ exampleDate | date: 'dd.MM.yyyy' }}  
    </p>  
  `,  
})  
export class DatePipeExampleComponent {  
  exampleDate = new Date(2000, 12, 12)  
}
```

Пример пайпа "date"

Typescript

- ⊗ Код приложения пишется на Typescript
- ⊗ Typescript - код со статической типизацией, который компилируется в JavaScript

The image shows the official TypeScript logo, which consists of the letters 'TS' in a white, bold, sans-serif font, centered within a solid blue square. The logo is positioned on the right side of the slide. The background of the entire slide is dark blue with a pattern of small, semi-transparent dots in various shades of blue, purple, and pink, creating a modern, tech-oriented aesthetic.

TS

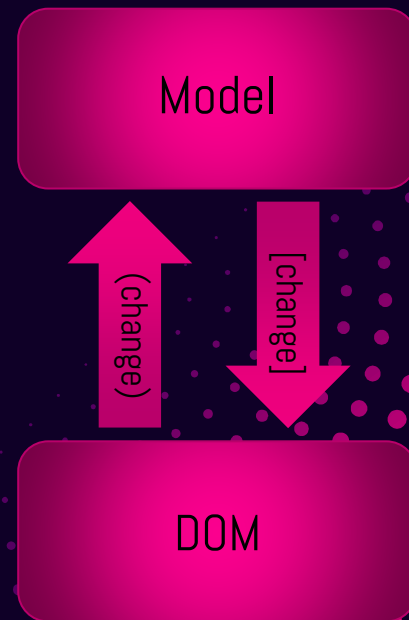
DI и сервисы

Dependency injection - это шаблон, который реализует паттерн “Inversion of Control”. DI позволяет создавать объект, использующий другие объекты. При этом изменения в определении используемых объектов никак не влияют на создаваемый объект.



Связывание данных

- ⊗ Удобно выполнено связывание данных в шаблоне и коде
- ⊗ Имеется двустороннее связывание



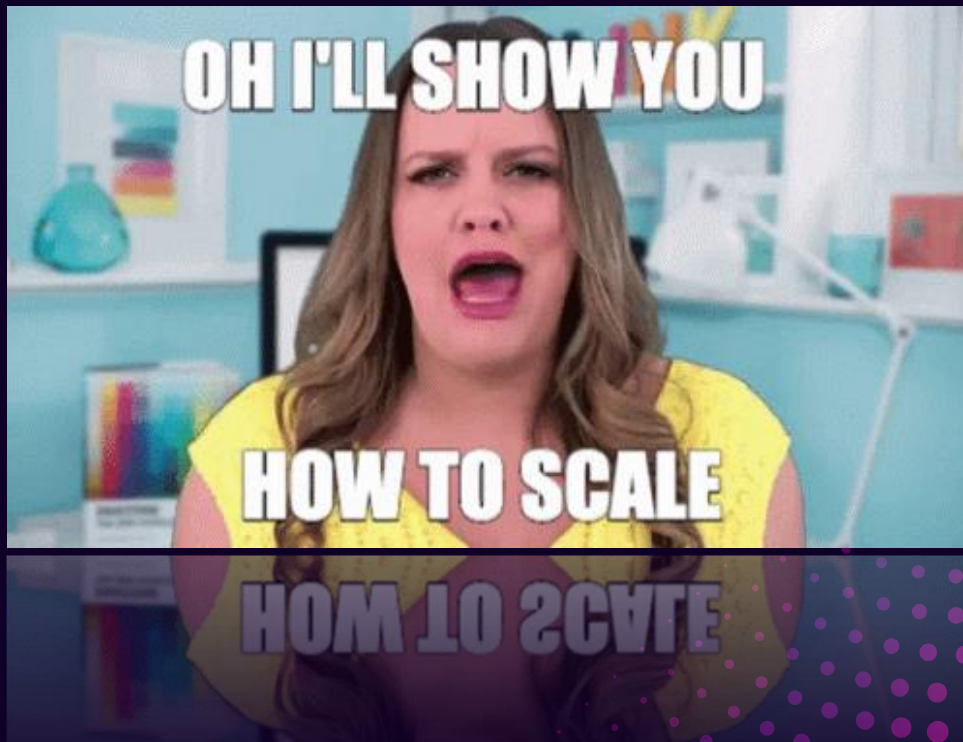
Дополнительные оптимизации ✈️

Есть множество инструментов для оптимизации

- ⊗ Ленивая загрузка
- ⊗ Стратегии предзагрузки
- ⊗ Анализ бандлов
- ⊗ Виртуальная прокрутка
- ⊗ ...



Хорошая масштабируемость

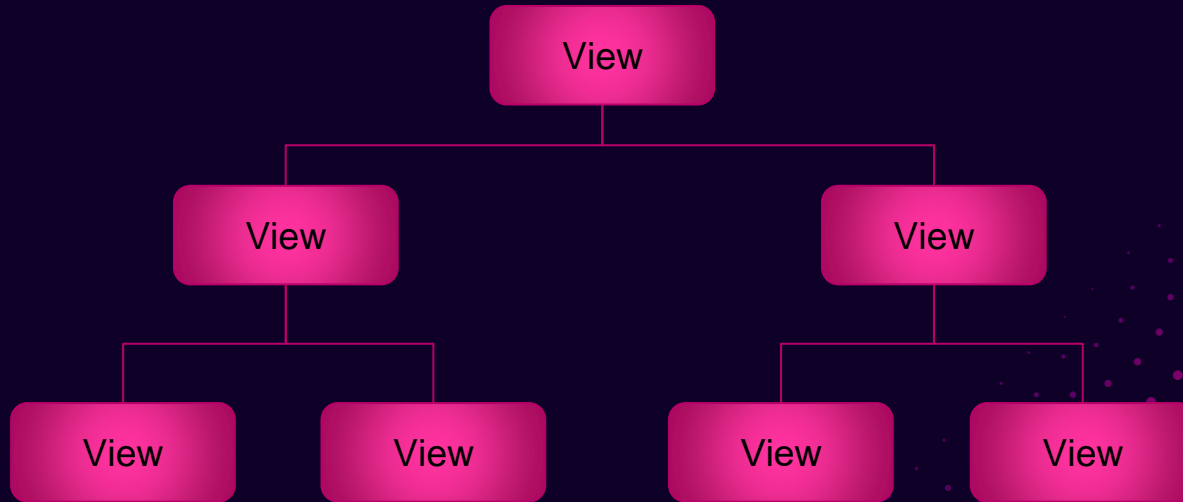


АОТ и JIT компиляция

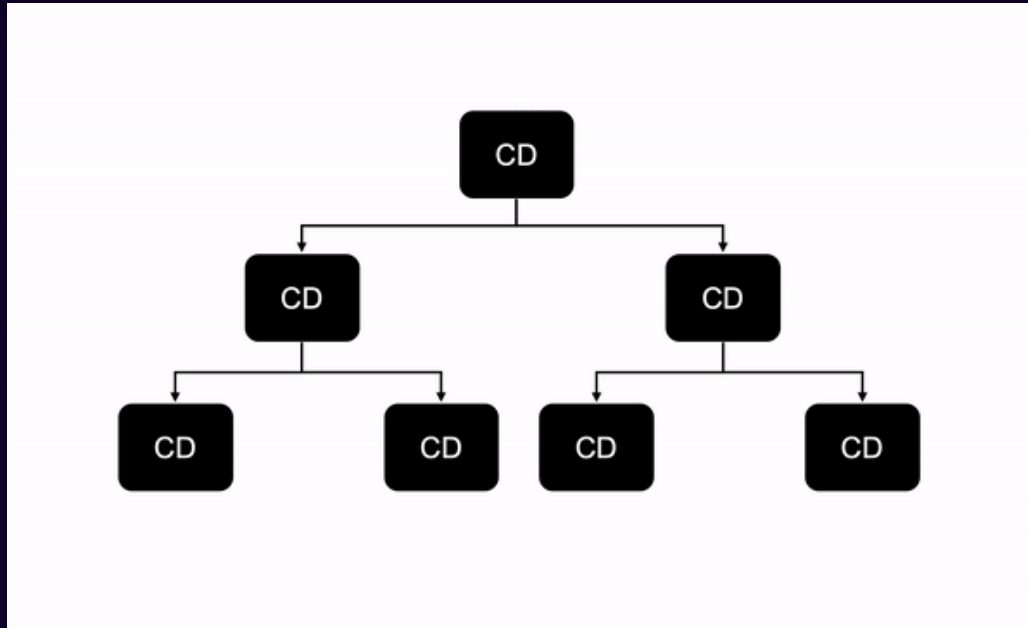
- ⊗ АОТ - компилирует приложение на этапе сборки
 - ⦿ быстро загружается в браузере
 - ⦿ обнаружение ошибок при сборке
 - ⦿ повышенная безопасность
 - ⦿ но сборка занимает значительно больше времени
- ⊗ JIT - компилирует приложение в момент запуска в браузере



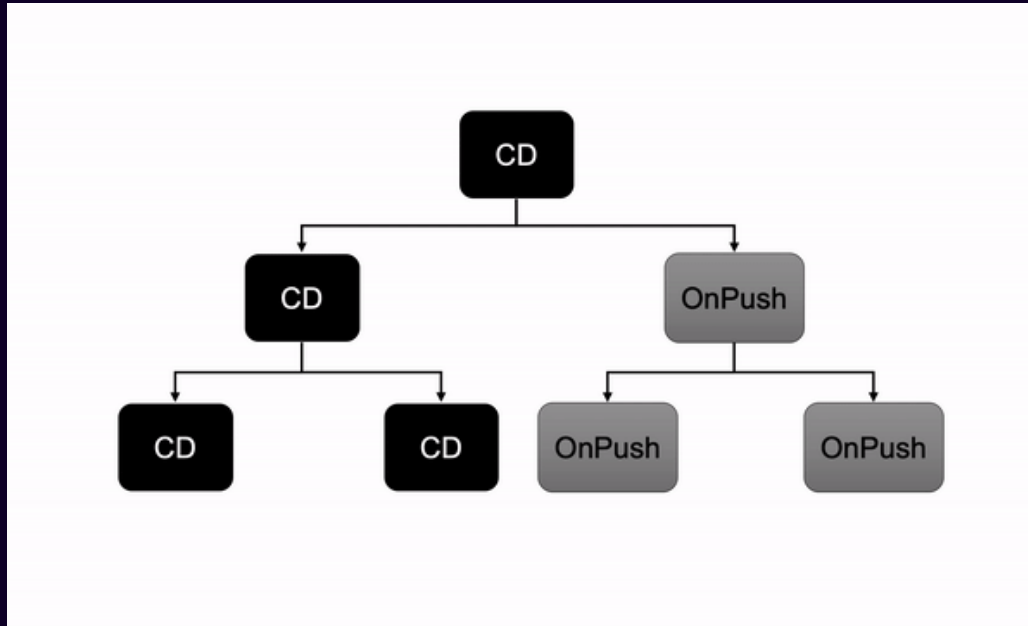
Change detection



Change detection



Change detection c onPush



Реактивные формы

src/app/profile-editor/profile-editor.component.ts (form group)

```
import { Component } from '@angular/core';
import { FormGroup, FormControl } from '@angular/forms';

@Component({
  selector: 'app-profile-editor',
  templateUrl: './profile-editor.component.html',
  styleUrls: ['./profile-editor.component.css']
})
export class ProfileEditorComponent {
  profileForm = new FormGroup({
    firstName: new FormControl(''),
    lastName: new FormControl(''),
  });
}
```

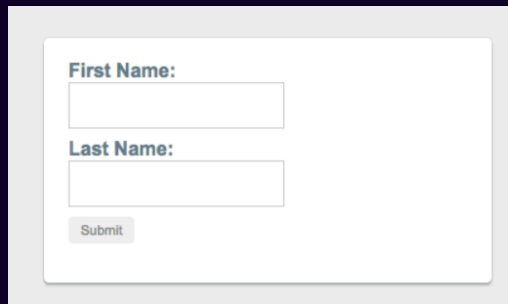
src/app/profile-editor/profile-editor.component.html (template form group)

```
<form [formGroup]="profileForm">

  <label>
    First Name:
    <input type="text" formControlName="firstName">
  </label>

  <label>
    Last Name:
    <input type="text" formControlName="lastName">
  </label>

</form>
```



First Name:

Last Name:

Submit

Маршрутизация

Маршрутизация - переход от одного представления к другому. Причем такой переход может осуществляться внутри представления

При переходе страница не перезагружается

При определении маршрута можно указать:

- ⊗ Маршрут
- ⊗ Компонент для отображения
- ⊗ Группу дочерних маршрутов
- ⊗ Дополнительные данные
- ⊗ Гарды (Guards)

```
const routes: Routes = [  
  {  
    path: 'login',  
    loadChildren: () =>  
      import('./modules/login/login.module').then((m) => m.LoginModule),  
  },  
  {  
    path: 'secretModule',  
    loadChildren: () =>  
      import('./modules/todo/todo.module').then((m) => m.TODOModule),  
    canLoad: [Guard],  
  },  
  {  
    path: 'my-component',  
    component: MyComponent,  
    children: [  
      {  
        path: 'another-component',  
        component: AnotherComponent,  
      },  
    ],  
  },  
  {  
    path: '**',  
    redirectTo: 'login',  
    pathMatch: 'full',  
  },  
];
```

Управление состояниями

Angular уже предоставляет множество готовых решений для управления состояниями.

Но в средних/больших проектах управление состояниями может становиться сложным в обслуживании

Управление состояниями

NGRX — это группа библиотек, «вдохновленных» шаблоном Redux

Основные принципы:

- ⊗ Единственный источник правды
- ⊗ Состояние read-only
- ⊗ Изменения вносятся с простыми функциями



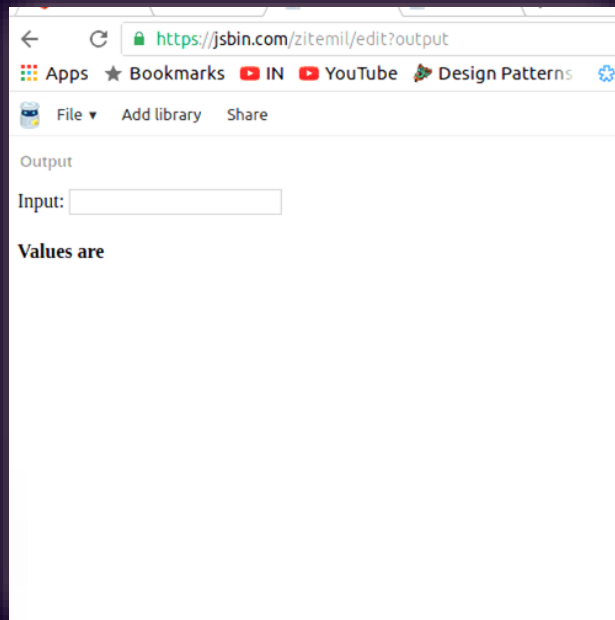
Rxjs

**RxJS**

```
document.addEventListener('click', () => console.log('Clicked!'));
```

```
import { fromEvent } from 'rxjs';

fromEvent(document, 'click').subscribe(() => console.log('Clicked!'));
```



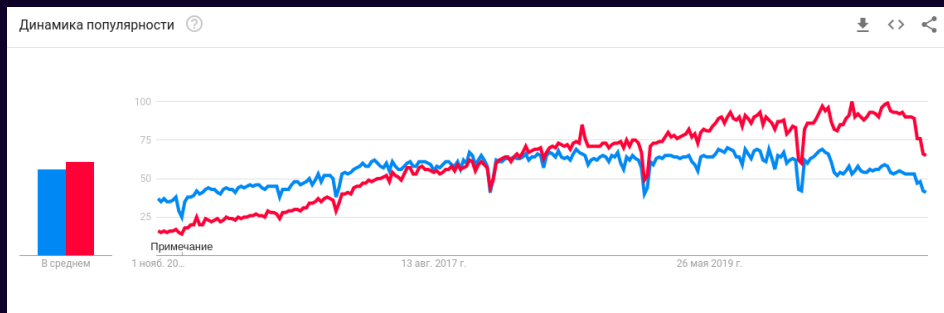
Минусы Angular

1. Выше порог вхождения
2. Потребуется время на оптимизацию
3. Время разработки значительно больше

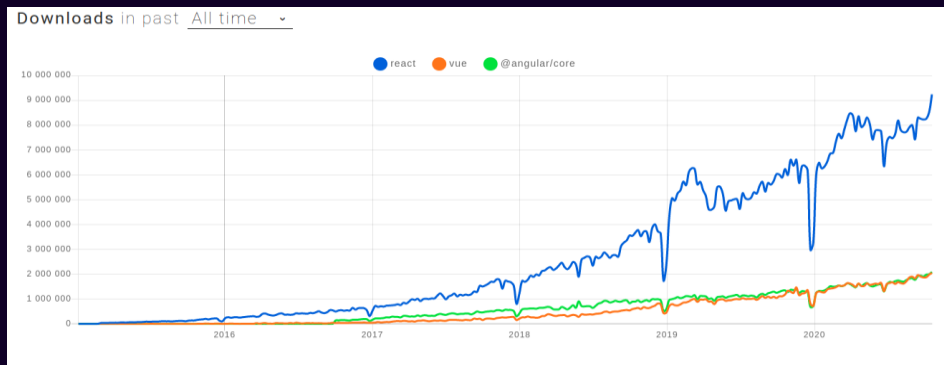
Сравнение

Сравним с другими JS-фреймворками

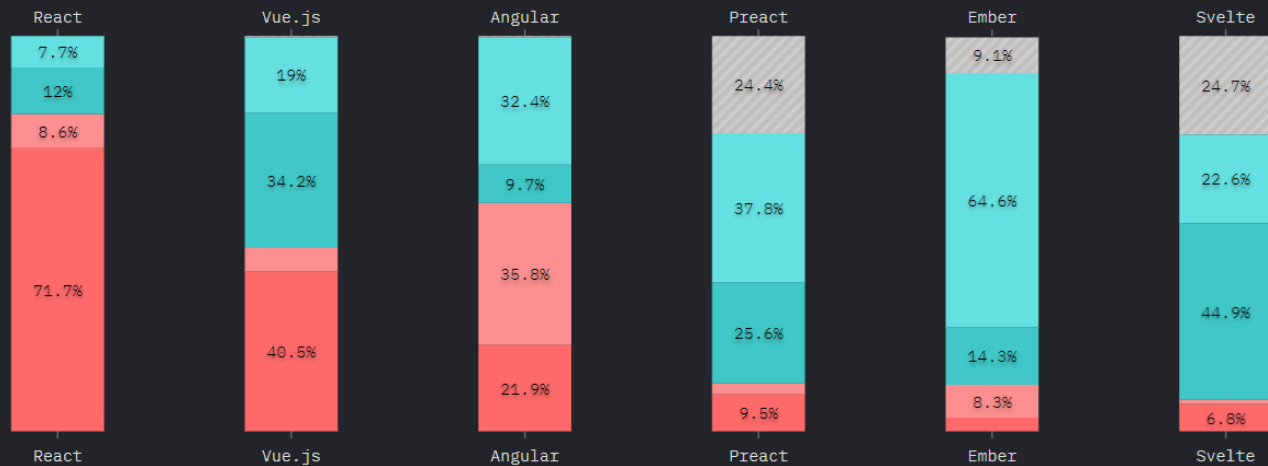
Популярность



Тренды google



Тренды npm



■ I've USED it before, and WOULD use it again
 ■ I've USED it before, and would NOT use it again
 ■ I've HEARD of it, and WOULD like to learn it
 ■ I've HEARD of it, and am NOT interested
 ■ I've never heard of it

■ I've USED it before, and WOULD use it again
 ■ I've USED it before, and would NOT use it again
 ■ I've HEARD of it, and WOULD like to learn it
 ■ I've HEARD of it, and am NOT interested
 ■ I've never heard of it



Итоги

Актуально?

Под вопросом

Стоит попробовать?

Стоит

Найдете работу?

Да

Спасибо за
внимание ;)

Полезные ссылки

Angular:

- ⊗ Оф. сайт: [Angular.io](https://angular.io)
- ⊗ Понятные статьи: [Angular | Директивы](#)
- ⊗ Создание проекта онлайн: <https://stackblitz.com>
- ⊗ Ivy: [All you need to know about Ivy. The new Angular engine!](#)
- ⊗ Ivy: [What is Angular Ivy?](#)
- ⊗ ChangeDetection: <https://medium.com/@buhtatyalexander90/последний-гайд-который-вы-прочтете-по-обнаружению-изменений-change-detection-в-angular-919bb70b3e53>

Rxjs:

- ⊗ [Learn RxJS: Introduction](#)
- ⊗ [Introduction](#)

Ngrx:

- ⊗ ngrx.io

Наш небольшой список материалов:

<https://www.notion.so/f5f1131256e94ed5b508a4b658a7388d?v=ea29948fa68c4e4ba720ffe3365422a9>