TECH RADAR 25

18→19 апреля

Тебе не нужен Spring, когда есть Angular

Евгений Щемелинин

Старший проектировщик

ГЛАВНОЕ ИНЖЕНЕРНОЕ СОБЫТИЕ ГОДА

backend / frontend / qa / ml&python / analytics / infosec / devops / database / mobile / support

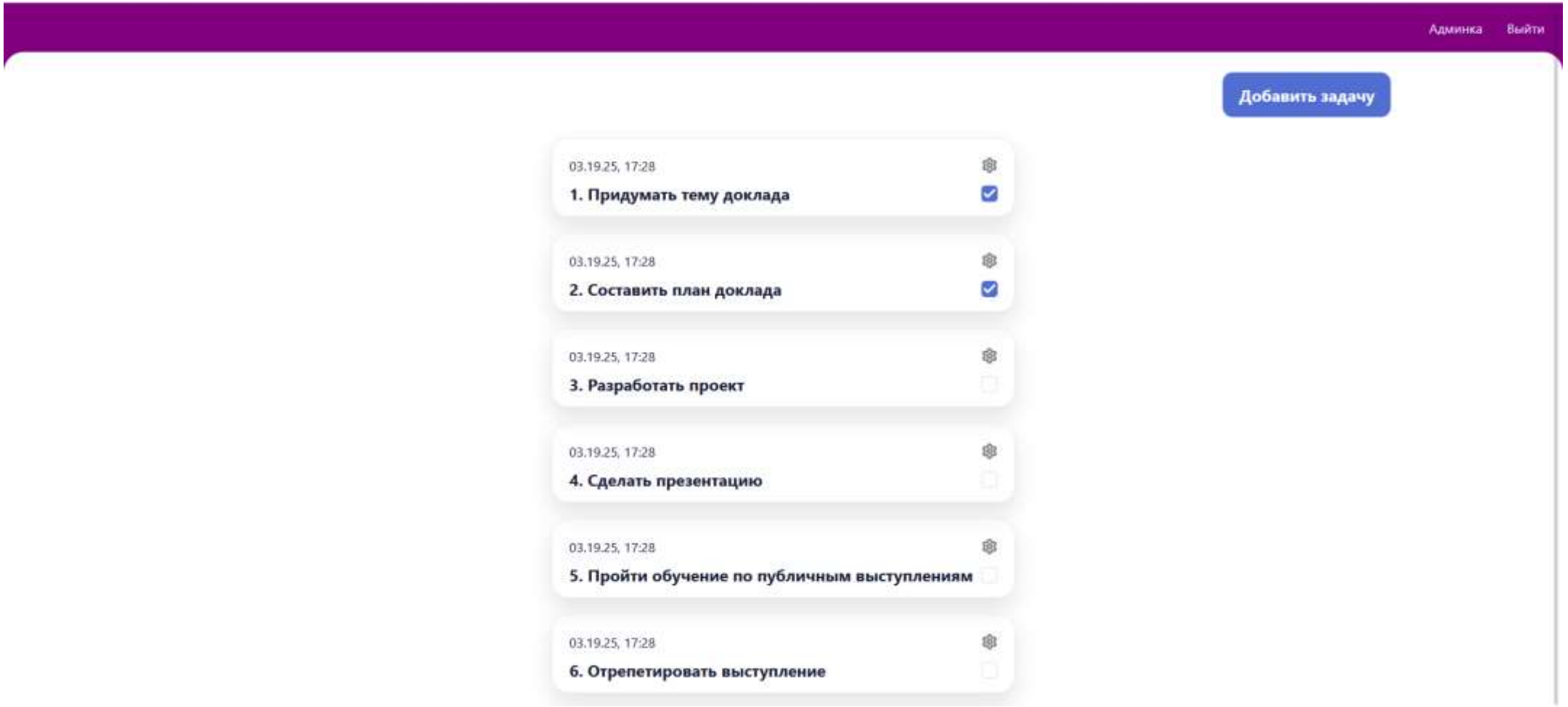
О себе

- Суммарный опыт разработки около 5 лет
- 2.5 года работы в роли fullstack разработчика (Angular/Spring)
- Проходил курсы по Angular от IBS + читал много всякого
- В свободное время коммичу в angdev.ru

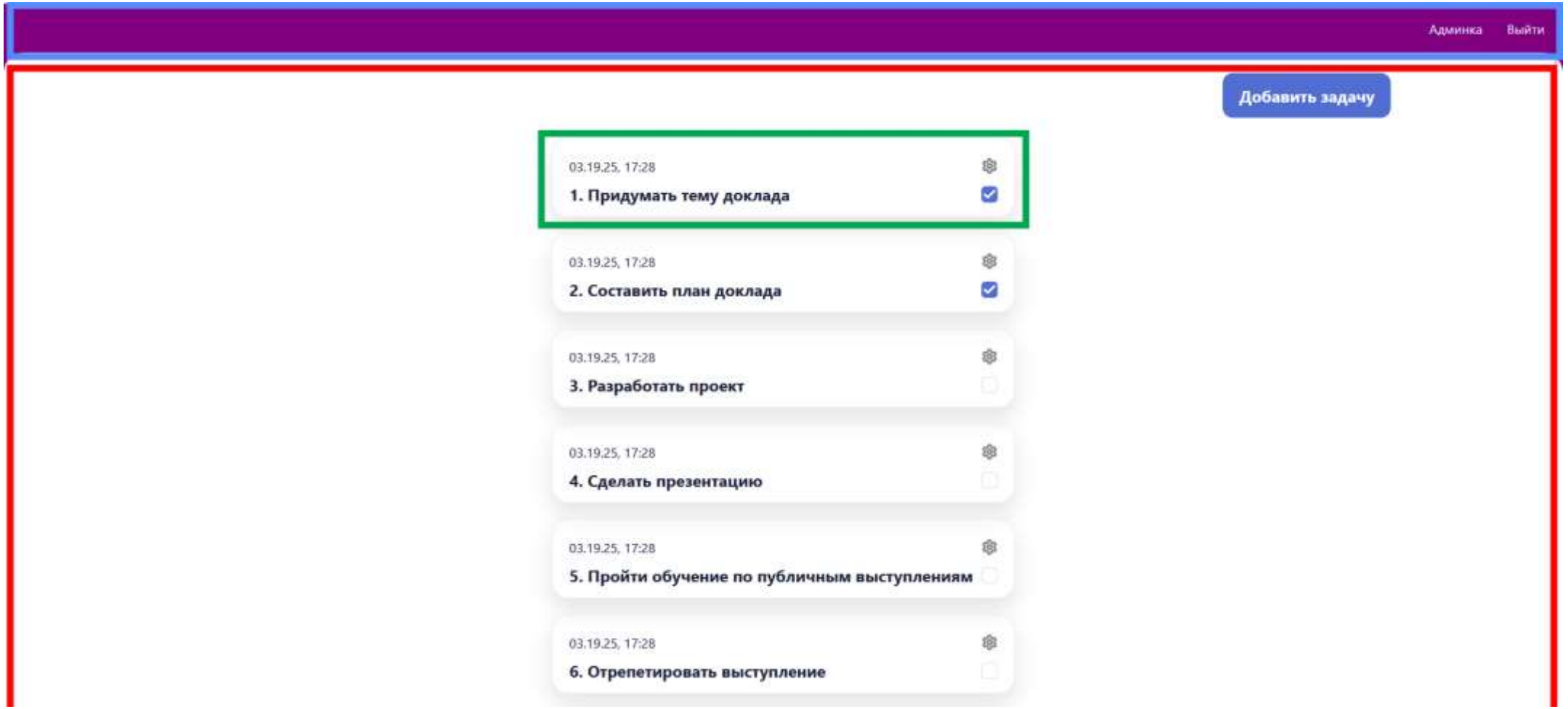
Чем схожи Фреймворки Angular и Spring

- ООП – как основная парадигма
- Java и TypeScript
- Dependency Injection – основной паттерн
- Используют схожие единицы организации кода: Controller, Component, Service
- Поддерживаются Intellij Idea Ultimate

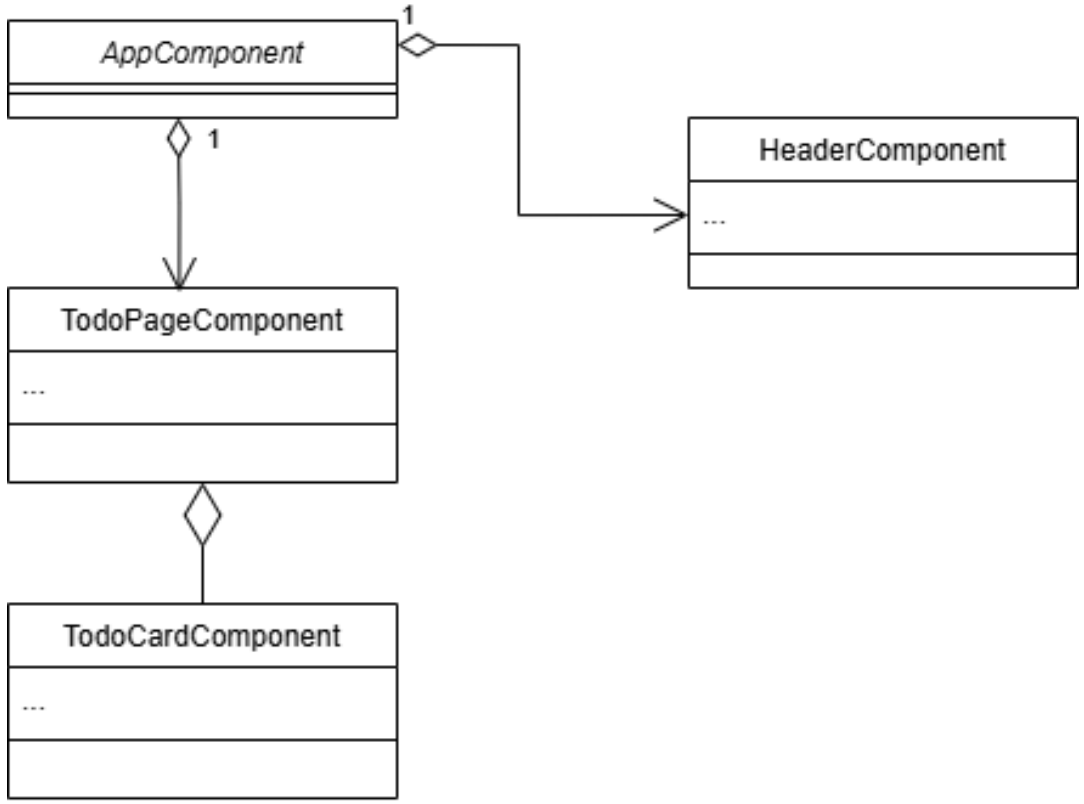
Структура приложения (Найди кота)



Коты приложения



Дерево компонентов



Шаблон корневого компонента AppComponent

```
1  <tui-root>
2    <app-header></app-header>
3    <main>
4      <router-outlet/>
5    </main>
6  </tui-root>
7
```

Код компонента TodoPage

```
@Component({
  selector: 'app-todo-page',
  templateUrl: './todo-page.component.html',
  styleUrl: './todo-page.component.scss',
})
export class TodoPageComponent implements OnDestroy {
  private todoService = inject(TodoService);
  private route = inject(ActivatedRoute);
  private subscription: Subscription;
  protected isOpenModal = false;
  public editTodo: Todo = {
    title: '',
  };
  public action!: 'add' | 'update';

  todos$ = new BehaviorSubject<Todo[]>([]);

  constructor(private auth: AuthService) {
    this.subscription = this.route.url
      .pipe(
        switchMap((url) => {
          return this.todoService.getTodos(url[1].path);
        })
      )
      .subscribe((todos) => this.todos$.next(todos));
```

Шаблон компонента TodoPage

```
<div class="header">
  <button tuiButton type="button" size="m" (click)="openModal()">
    Добавить задачу
  </button>
</div>
<div class="content">
  @for (todo of todos$ | async; track todo.id; let id = $index) {
    <app-todo-card
      [todo]="todo"
      [number]="id + 1"
      (edit)="openModal(todo)"
      (remove)="onRemove(todo)"
      (changeStatus)="onStatusChange($event, todo)"
    ></app-todo-card>
  } @empty {
    У вас нет задачь ;(-
  }
</div>
```

Сущности в Angular

- Component
- Service
- Interceptor
- Pipe
- Guard
- Directive
- + RxJs/Signals

Todo service

```
7  @Injectable({
8    providedIn: 'root',
9  })
10 export class TodoService {
11
12   constructor(private http: HttpClient) {}
13
14   getTodos(userId: string): Observable<Todo[]> {
15     const url = Endpoints.todo.todos({userId});
16     return this.http.get<Todo[]>(url);
17   }
18
19   add(userId: string, todo: Todo): Observable<Todo[]> {
20     const url = Endpoints.todo.todos({userId});
21     return this.http.post<Todo[]>(url, {todo});
22   }
```

JwtInterceptor

```
5   export const jwtInterceptor: HttpInterceptorFn = (req, next) => {
6     const auth = inject(AuthService);
7     const token = auth?.userValue?.token;
8     if (token) {
9       req = req.clone({
10        setHeaders: {
11          Authorization: `Bearer ${token}`,
12        },
13      });
14    }
15
16    return next(req);
17  };
18
```

Шаблон TodoCard

```
<div class="card">
  <div class="card__header">
    <span>{{todo.createdAt | date :'MM.dd.yy, HH:mm'}}</span>
    <button
      [tuiDropdownManual]="isOpen"
      (click)="openContextMenu()"
      (tuiActiveZoneChange)="onActiveZone($event)"
      (tuiObscured)="onObscured($event)"
      tuiIconButton
      appearance="icon"
      size="s"
    >
    </button>
  </div>
  <div class="card__body">
    <p><b>{{ number }}. {{ todo.title }}</b></p>
    <input tuiCheckbox type="checkbox" [ngModel]="todo.completed" size="s" (click)="onStatusChange(!todo.completed)"/>
  </div>
</div>
```

DatePipe

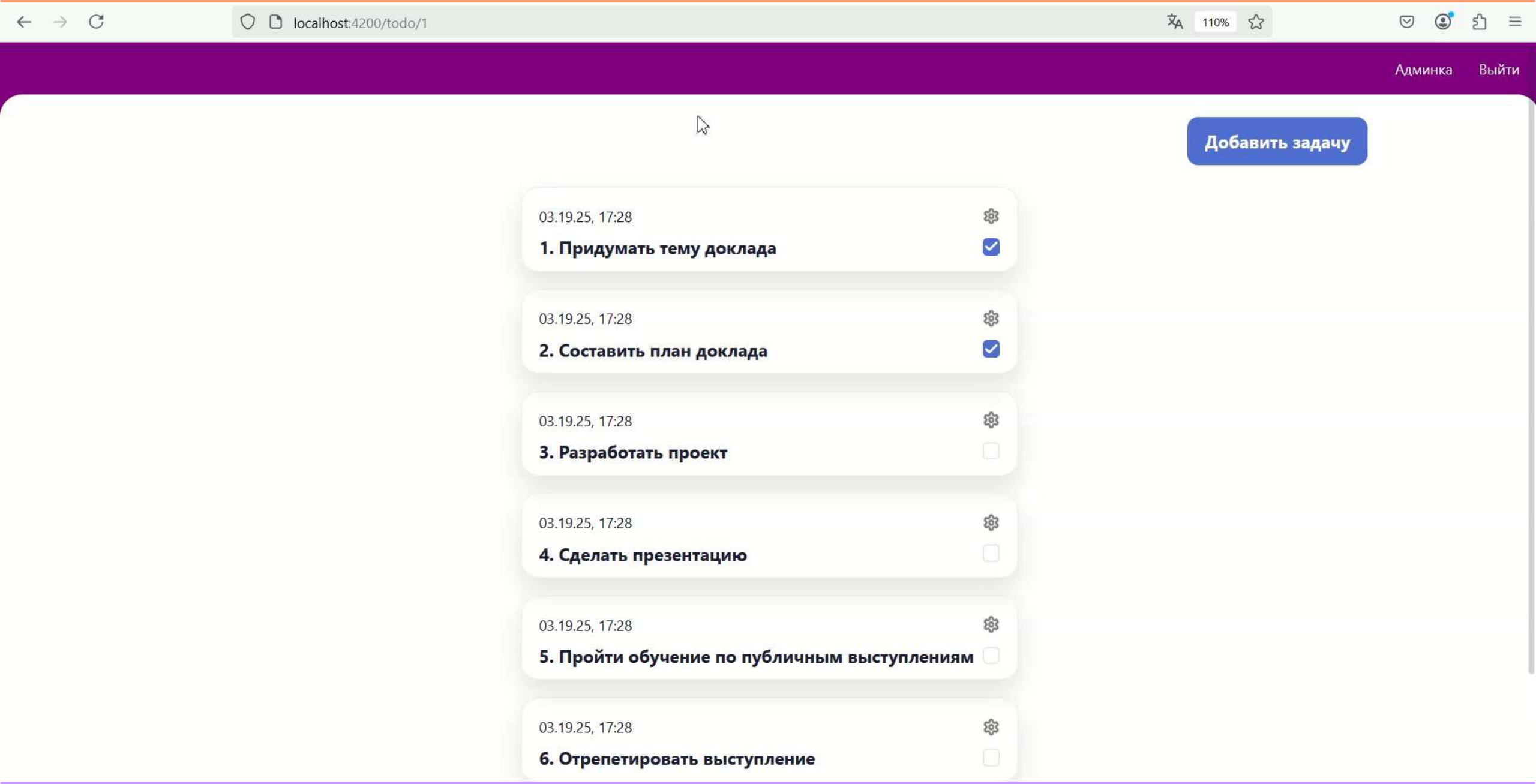
```
@Pipe({
  name: 'date',
})
export class DatePipe implements PipeTransform {
  private defaultOptions: any;
  private defaultTimezone: any;

  transform(
    value: Date | string | number | null | undefined,
    format?: string,
    timezone?: string,
    locale?: string,
  ): string | null {
    if (value == null || value === '' || value !== value) return null;

    try {
      const _format = format ?? this.defaultOptions?.dateFormat ?? DEFAULT_DATE_FORMAT;
      const _timezone =
        timezone ?? this.defaultOptions?.timezone ?? this.defaultTimezone ?? undefined;
      return formatDate(value, _format, locale || this.locale, _timezone);
    } catch (error) {
      throw invalidPipeArgumentError(DatePipe, (error as Error).message);
    }
```

Интерфейс PipeTransform

```
27  interface PipeTransform {
28    transform(value: any, ...args: any[]): any
29  }
```

Стандартные Pipe

- **JsonPipe**: отображает объект в формате Json
- **DatePipe**: форматирует дату
- **CurrencyPipe**: форматирует валюту
- **PercentPipe**: форматирует проценты
- **AsyncPipe**: отображает асинхронную информацию
- **LowerCasePipe и UpperCasePipe** : переводит строку в нужный регистр
- **DecimalPipe**: задает формат числа
- **SlicePipe**: обрезает строку

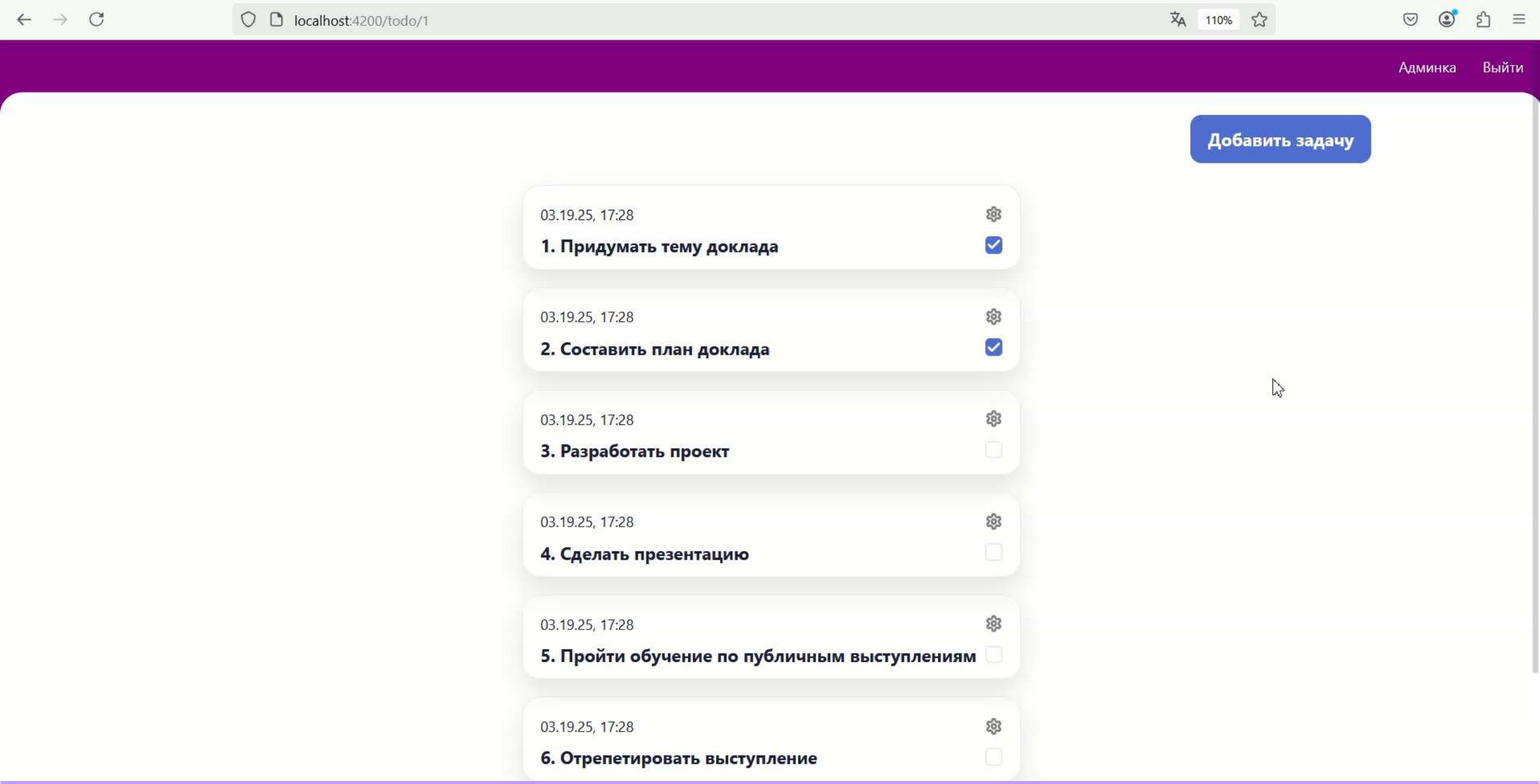

localhost:4200/todo/1
110%
Админка
Выйти
Добавить задачу
03.19.25, 17:28
1. Придумать тему доклада
03.19.25, 17:28
2. Составить план доклада
03.19.25, 17:28
3. Разработать проект
03.19.25, 17:28
4. Сделать презентацию
03.19.25, 17:28
5. Пройти обучение по публичным выступлениям
03.19.25, 17:28
6. Отрепетировать выступление

haveAccessGuard

```
5  export const haveAccessGuard: CanActivateFn = (route, state) => {
6    const auth = inject(AuthService);
7    const router = inject(Router);
8
9    if (route.params['id'] === auth.userValue?.id) {
10     return true;
11   } else {
12     router.navigate(['/']);
13     return false;
14   };
15 };
```

Routes

```
5  export const routes: Routes = [
6    {
7      path: '',
8      pathMatch: 'full',
9      component: AboutPageComponent,
10   },
11   {
12     path: 'todo/:id',
13     canActivate: [isAuthGuard, haveAccessGuard],
14     loadComponent: () =>
15       import('./components/todo-page').then((comp) => comp.TodoPageComponent),
16   },
17   {
18     path: 'login',
19     canActivate: [isNotAuthGuard],
20     loadComponent: () =>
21       import('./components/auth-page').then((comp) => comp.AuthPageComponent),
22   },
```


localhost:4200/todo/1
110%
Админка
Выйти
Добавить задачу
03.19.25, 17:28
1. Придумать тему доклада
03.19.25, 17:28
2. Составить план доклада
03.19.25, 17:28
3. Разработать проект
03.19.25, 17:28
4. Сделать презентацию
03.19.25, 17:28
5. Пройти обучение по публичным выступлениям
03.19.25, 17:28
6. Отрепетировать выступление

HeaderComponent

```
<header tuiNavigationHeader tuiTheme="light">
  @if (auth.user$ | async) {
    <button type="button" tuiButton *hasRole="['admin']">Админка</button>
    <button type="button" tuiButton (click)="logout()">Выйти</button>
  } @else {
    <a type="button" tuiButton [routerLink]="url.signup">
      Зарегистрироваться
    </a>
    <a type="button" tuiButton [routerLink]="url.login">
      Войти
    </a>
  }
</header>
```

HasRoleDirective

```
12  @Directive({
13    selector: '[hasRole]',
14  })
15  export class HasRoleDirective implements OnInit, OnDestroy {
16    @Input() hasRole!: string[];
17    subscription!: Subscription;
18
19    constructor(
20      private authService: AuthService,
21      private template: TemplateRef<any>,
22      private viewContainerRef: ViewContainerRef,
23    ) {}
24
25    ngOnInit(): void {
26      this.subscription = this.authService?.user$.subscribe((user) => {
27        if (!user) return;
28        this.checkRoles(user?.roles as string[]);
29      });
30    }
31
32    checkRoles(userRole: string[]) {
33      if (!userRole?.length) return;
34      if (this.hasRole.some((acessRole) => userRole.includes(acessRole))) {
35        this.viewContainerRef.createEmbeddedView(this.template);
36      } else {
37        this.viewContainerRef.clear();
38      }
```

Итог

Рассмотрели схожие концепции и сущности:

- DI
- Service
- Component
- ООП

Рассмотрели особенности Angular:

- Interceptor
- Pipe
- Guard
- Directive
- + RxJs/Signal

Итог

Рассмотрели схожие концепции и сущности:

- DI
- Service
- Component
- ООП

Рассмотрели особенности Angular:

- Interceptor
- Pipe
- Guard
- Directive
- + RxJs/Signal

Итог

Рассмотрели схожие концепции и сущности:

- DI
- Service
- Component
- ООП

Рассмотрели особенности Angular:

- Interceptor
- Pipe
- Guard
- Directive
- + RxJs/Signal

TECH RADAR 25

18→19 апреля

Тебе не нужен Spring, когда есть Angular

Евгений Щемелинин

ГЛАВНОЕ ИНЖЕНЕРНОЕ СОБЫТИЕ ГОДА

backend / frontend / qa / ml&python / analytics / infosec / devops / database / mobile / support