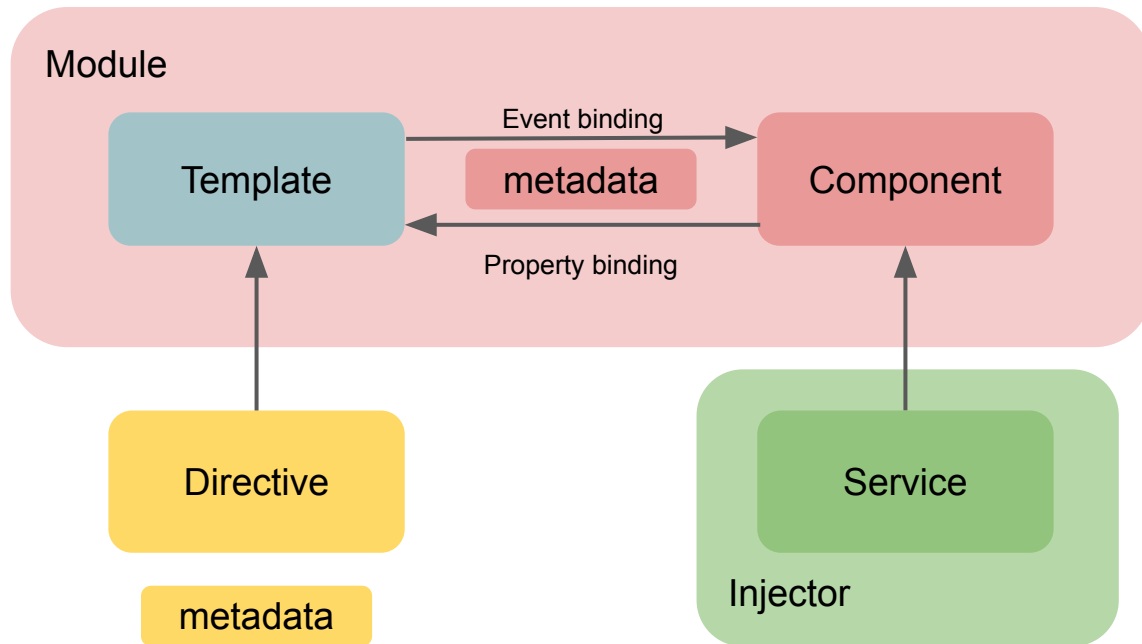




AngularJS

Архитектура приложения Angular



Module

*.module.ts

```
import { NgModule } from '@angular/core';  
import { BrowserModule } from '@angular/platform-browser';  
  
@NgModule({  
  imports: [ BrowserModule ],  
  providers: [ Logger ],  
  declarations: [ AppComponent ],  
  bootstrap: [ AppComponent ]  
})  
export class AppModule { }
```

decorator

service

main view

Component & Metadata

`*.component.ts`

```
import { Component } from '@angular/core';  
  
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})
```

metadata



```
export class AppComponent {  
  title = 'app works!';  
}
```

Component



Template

`*.component.html`

```
<h2>Hero List</h2>
```

```
<p><i>Pick a hero from the list</i></p>
```

```
<ul>
```

```
<li *ngFor="let hero of heroes" (click)="selectHero(hero)">
```

```
  {{hero.name}}
```

```
</li>
```

```
</ul>
```

```
<hero-detail *ngIf="selectedHero" [hero]="selectedHero"></hero-detail>
```

Data binding

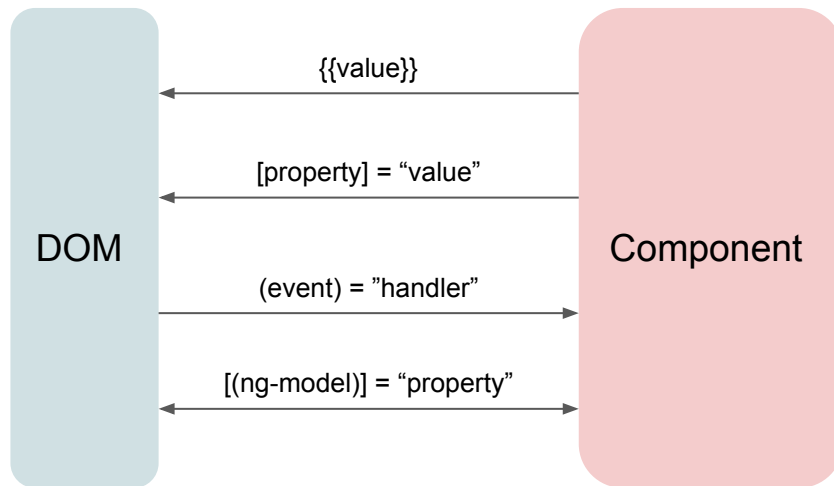
```
<li>{{hero.name}}</li>
```

```
<hero-detail [hero]="selectedHero"></hero-detail>
```

```
<li (click)="selectHero(hero)"></li>
```

Two-way data binding

```
<input [(ngModel)]="hero.name">
```



Directive

```
import { Component } from '@angular/core';  
  
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})
```

app/*.component.html (structural)

```
<li *ngFor="let hero of heroes"></li>  
<hero-detail *ngIf="selectedHero"></hero-detail>
```

Service

@Injectable

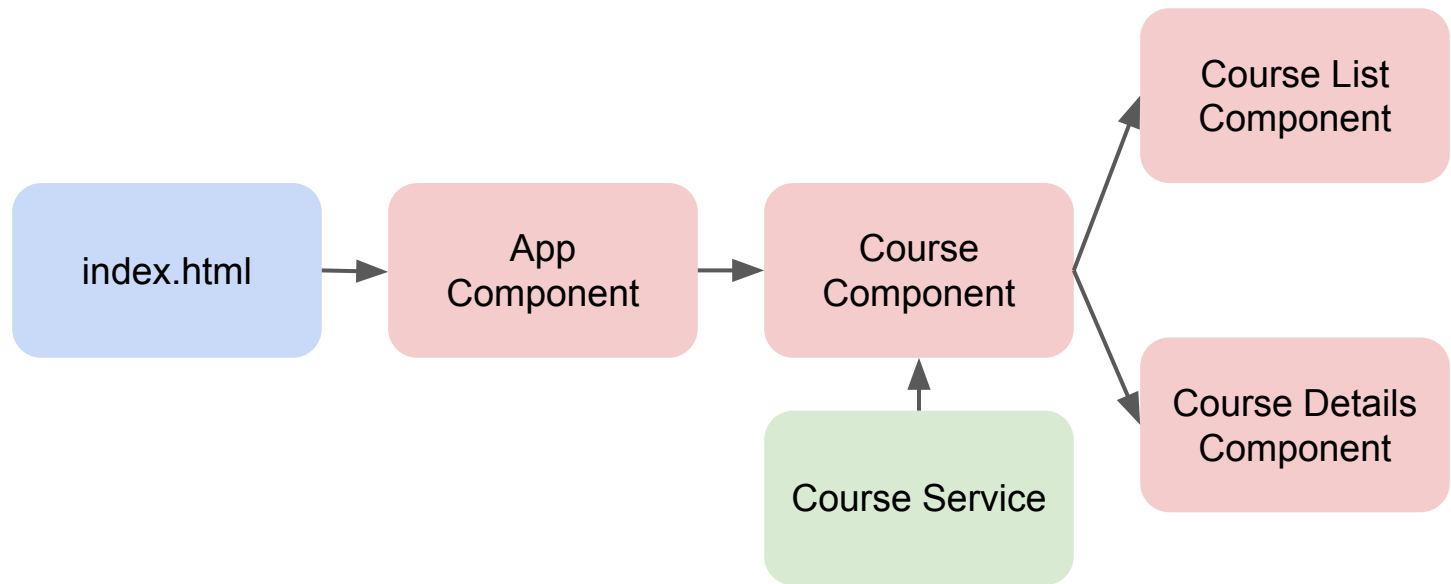
```
export class CourseService {  
  constructor(private backend: BackendService) { }  
  
  getCourses() {  
    return this.backend.getAll(Hero);  
  }  
}
```

Dependency injection

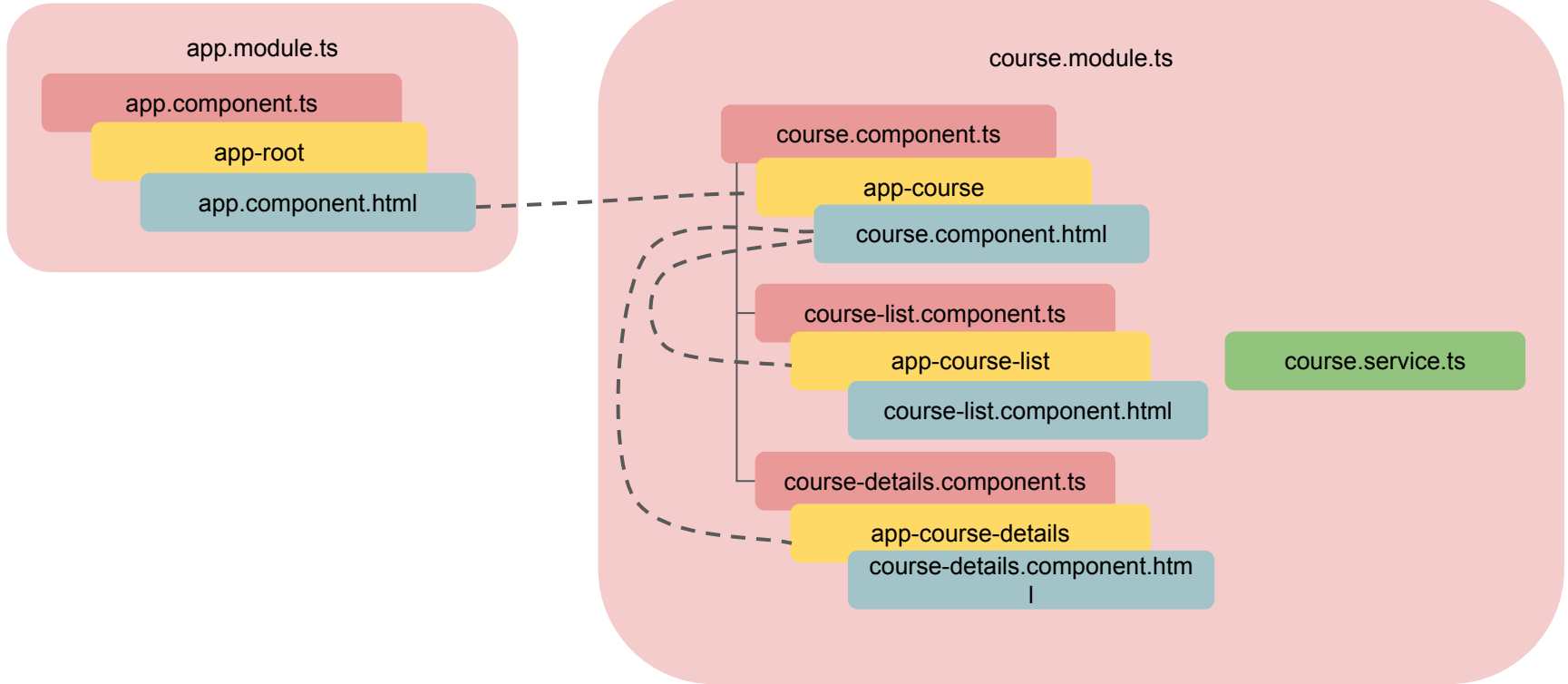
```
graph TD; DI[Dependency injection] --> CS[CourseService]; DI --> CC[CourseComponent];
```

```
@Component({  
  providers: [CourseService]  
})  
export class CourseComponent {  
  constructor(private courseService: CourseService) {  
  }  
}
```

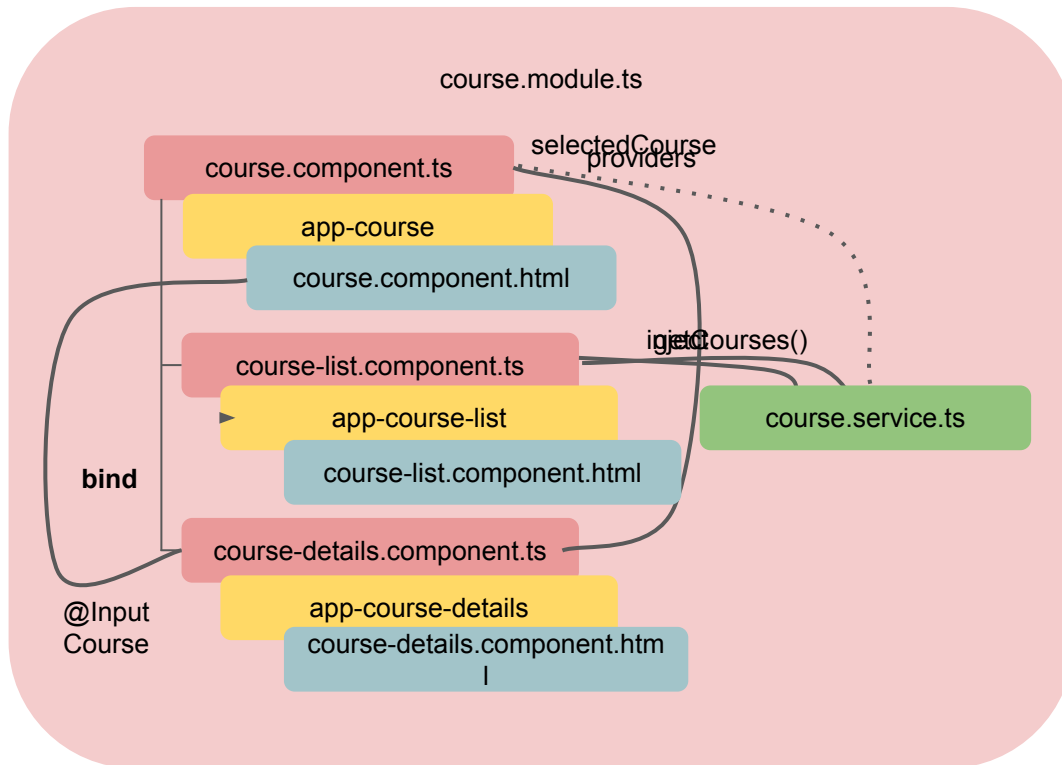
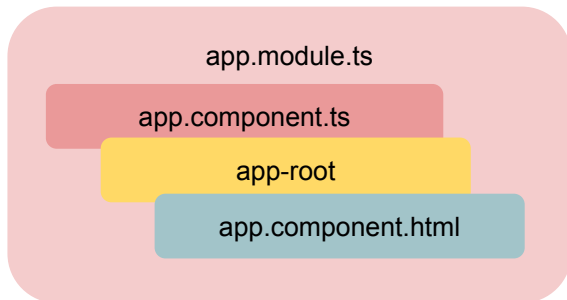

Архитектура приложения



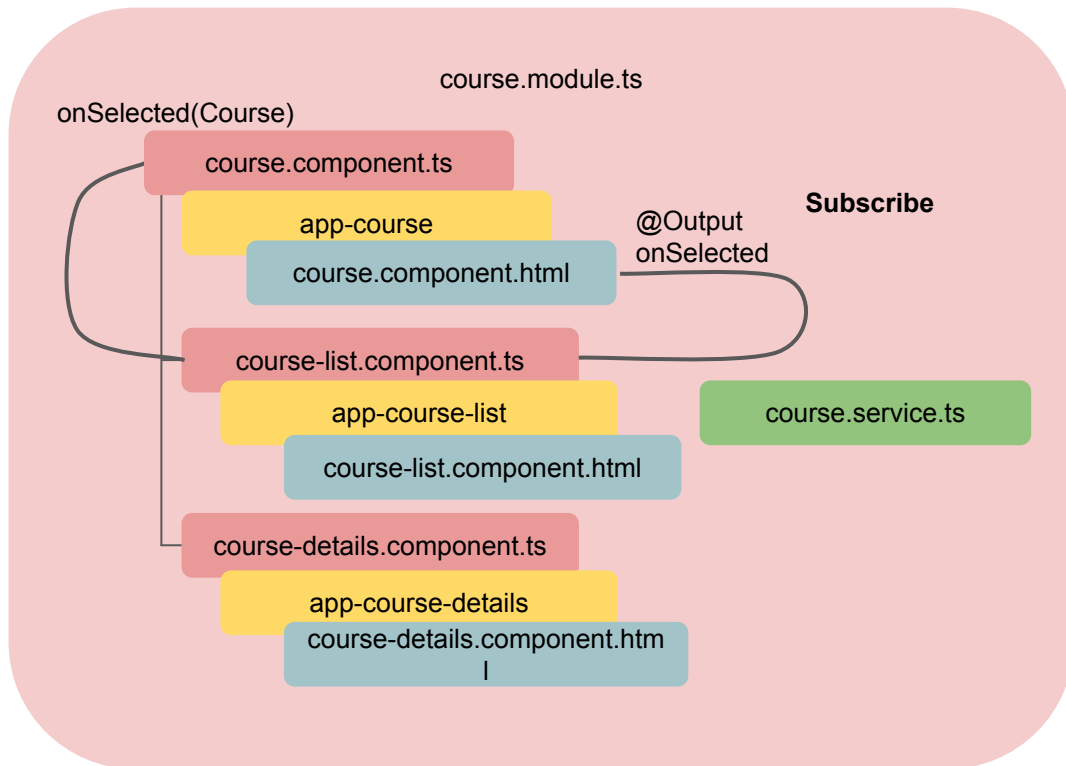
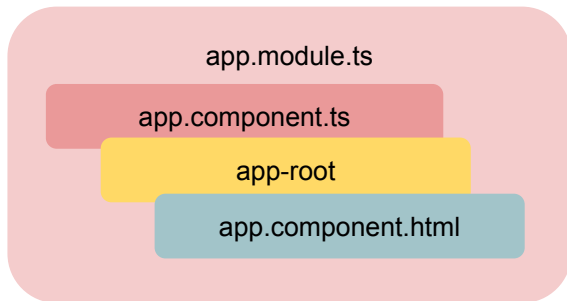
Templates Flow



Data Flow



Events Flow



Задание

По задаче из backlog'а сделать:

1. Список необходимых компонент
2. Описать назначение компонент
3. Нарисовать эскиз шаблона для компонентов

Пример

Задача: Список пользователей

Компоненты:

1. Список пользователей

Отображает список ФИО пользователей, реагирует на нажатие на пользователя изменением информации в компоненте детализации

1. Детализация пользователя

Отображает детальную информацию о выбранном из списка пользователе: ФИО, дату регистрации, почту ...

Шаблоны:

1	Петров Иван
	Иванов Петр
	Сидоров Андрей
	...
2	
	Иванов Петр
	29.01.2016
	ipetr@pmail.ru