



Базовые понятия






Component (компонент)

Одна из главных задач - правильное
разделение приложения на компоненты

Template
(ссылка на HTML)

Метаданные
(служебные данные движка
Angular)

Class на TypeScript
(свойства, методы - обработка
действий пользователя)



Service (сервис)



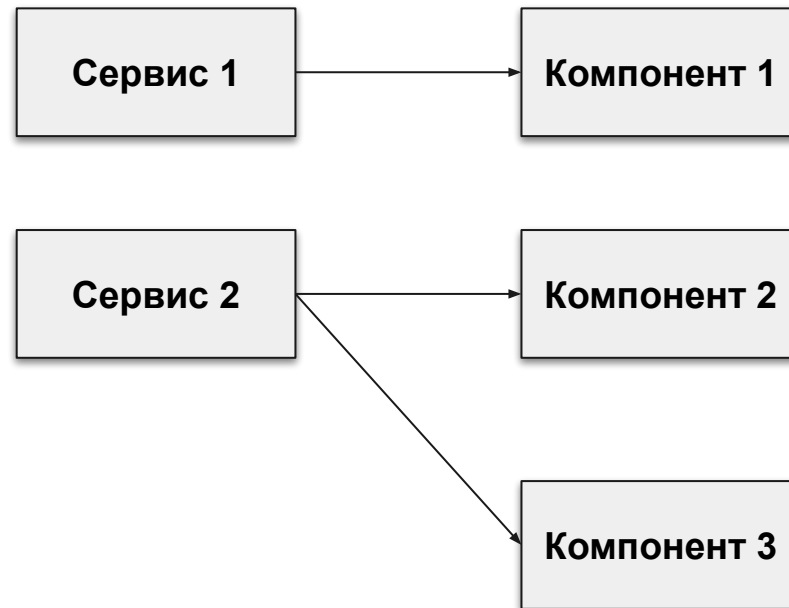
Класс на TypeScript - невизуальная часть приложения, которая реализовывает логику работы и взаимодействует с компонентами

Сервисов (как и компонентов) может быть любое количество, главное - не создавать лишние

Компонент напрямую не работает с данными - только через специальный сервис (разделение обязанностей)



Service (сервис)



Связи могут быть любыми. Главное, чтобы они были логичными и избыточными



Module (модуль)

Самый верхний элемент, которые может объединять множество компонентов, сервисов или любой код

Имеет свой API, который "открывает" разработчик и его можно использовать в других модулях

Модуль

Компоненты

Сервисы

Любой код

Все, что угодно



Принципы

Каждый модуль, компонент или сервис должен “заниматься” только своими делами - должен быть логичным и понятным

Правильное разделение приложения на компоненты и сервисы - залог расширяемости

Принцип “бритвы Оккама” - не нужно плодить лишнее без необходимости



RxJS

Реактивное программирование

Всегда используйте когда получаете
какие-либо данные

Позволяет выполнять асинхронные
вызовы данных

Автоматическое обновление значений
на странице



Домашнее задание

- <https://angular.io/guide/architecture-modules>
- <https://angular.io/guide/architecture-components>
- <https://angular.io/guide/architecture-services>
-
- DI в Angular:
 - <https://angular.io/guide/architecture-services>
 - <https://angular.io/guide/dependency-injection>
- Директивы:
 - <https://dzone.com/articles/angular-directives-1>

Не все может быть сразу понятно - главное пытаться уловить суть - впоследствии будете постоянно возвращаться к разделам и перечитывать