



# Smart and Dumb components





# Архитектура

**Smart компонент** - управляет другими компонентами, предоставляет данные

**Dumb компонент** - получает данные и отображает их, обрабатывает действия пользователей и отправляет их на обработку в Smart компонент

Одна из задач - правильно разбить приложение на Smart и Dumb компоненты

Избегаем "каши" из компонентов, которые взаимодействуют не формализовано

Четкое понимание API - как взаимодействуют компоненты

--

Один из способов решения: архитектура "**Smart and Dumb components**":

- Взаимодействие с компонентами преимущественно через декораторы @Input и @Output
- @Input не должны изменяться внутри **dumb** компонента (это делает **smart**)
- Разделить компоненты-представления (Presentational, Dumb, Pure) и компоненты-менеджеры (Smart, Container)

Сервисы

AppComponent

Вся бизнес-логика

**TodosComponent**  
(отображение)

@Input  
@Output

**CategoriesComponent**  
(отображение)

@Input  
@Output

Другие Dumb компоненты

**AppComponent** - smart, container:

- Собирает всю страницу из других компонентов
- Управляет данными дочерних компонентов
- Реагирует на действия дочерних компонентов

**Минусы:** при сложном дереве компонентов - трудно передавать параметры и отслеживать события (решение - использовать библиотеку для хранения состояний, например

**TodosComponent, CategoriesComponent** - dumb, presentational, pure

- Получают данные через @Input и отображают
- Не знаю откуда данные берутся, их задача - просто отобразить
- При необходимости - отправляют действия пользователя с помощью @Output в Smart-компонент
- Напрямую не обращаются к данным

Сервисы

AppComponent

Вся бизнес-логика

**TodosComponent**

(отображение)

@Input

@Output

**CategoriesComponent**

(отображение)

@Input

@Output

Другие Dumb компоненты

Каждый компонент должен заниматься только своим делом

**Smart** компоненты нужно разделять логически, в зависимости от круга задач - в нашем случае достаточно одного: **AppComponent**

Содержит все бизнес-процессы, может запрашивать данные

---

**Dumb** - все компоненты для отображения

Можно изменять внутреннюю реализацию, но это не отразится на работу приложения (т.к. API из набора @Input и @Output не меняется)

Легче тестировать



# Домашнее задание

- Прочитать PDF  
<https://drive.google.com/open?id=1O133XQCAMb3xcWtogtTMMuBsAC7PPDgX>