

Лекция №7: JavaScript-фреймворки

Web-программирование / ПГНИУ

Введение

Клиентскую часть web-приложения в части javascript разрабатывать просто, пока она работает с небольшими частями страницы.

JS отвечает за анимации, интерактивные элементы (слайдеры, дата-пикеры), небольшие части страницы с AJAX.

Но иногда клиентская часть становится сложной, и весь (или большая часть) GUI разрабатывается на JS.

Архитектура - организация системы

- Гибкость (расширяемость)
- Поддерживаемость
- Масштабируемость процесса разработки на множество разработчиков (команд)
- Тестируемость
- Переиспользование

Модули

- Сложно работать с системой как с большим монолитным целым
- Система декомпозируется на структурные единицы
- Декомпозиция иерархическая
- Пакеты, модули, компоненты
- Взаимодействие модулей? Управление модулями?

MV*

У нас есть:

- Данные
- Графический интерфейс - представление (с которым взаимодействует пользователь!)
- Какая-то "логика", которая может менять представление и данные

Требуется как-то всё это связать.

С этим помогают паттерны: **MVC, MVP, MVVM**

MVP

- Model-View-Presenter
- Модель - бизнес-логика, слой данных
- Представление - отображение данных и предоставление функций и событий для взаимодействия с пользователем (реализация UI)
- Презентёр - абстракция представления
- Презентёр: взаимодействует с UI, подписывается на события UI, взаимодействует с моделью в обе стороны.

MVC

- Model-View-Controller
- Модель - бизнес-логика, слой данных
- Представление - отображение данных (+БЛ для этого отображения)
- Контроллер - связь пользователя с системой, определяет, какое сейчас используется представление
- Модель сообщает об изменениях представлению (представление запрашивает данные или подписывается на них)
- Представление использует контроллер для изменения состояния (вызывает методы контроллера, или контроллер подписывается на события представления)

MVVM

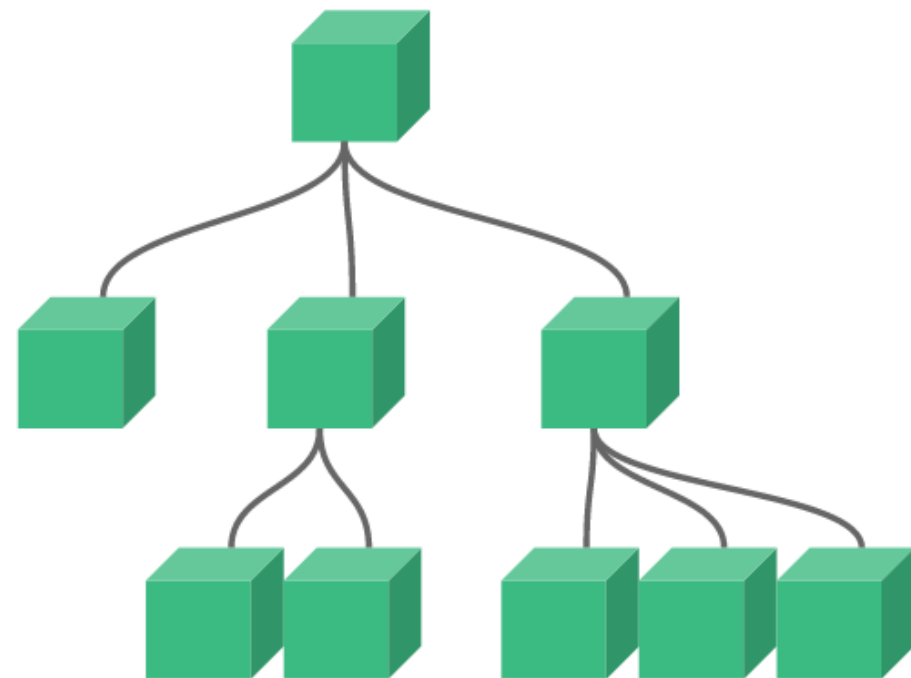
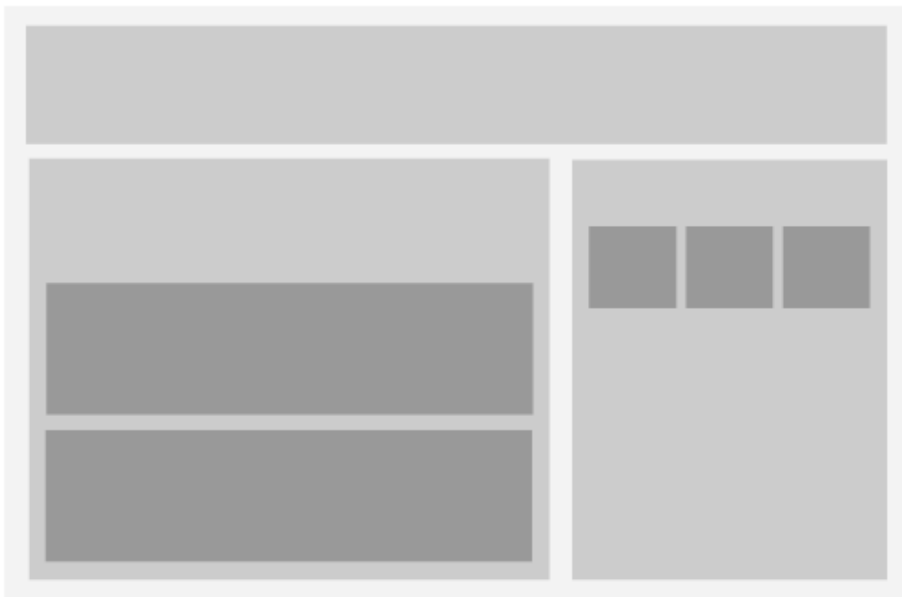
- Model-ViewModel-View
- Модель - бизнес-логика, слой данных
- Представление - отображение данных
- ViewModel - связывание представление и модели, абстракция представления + обёртка над моделью
- Изменение ViewModel **автоматически** изменяет представление
- ViewModel получает данные от модели, обновляет данные в модели и может подписываться на их изменение

JavaScript фреймворки

- Современные JS фреймворки — это инструменты разработки динамических веб-страниц
- Минимум - реализуют разработку переиспользуемых и расширяемых "UI виджетов"
- Максимум - определяют архитектуру всего приложения
- React.js, Vue.js, Angular, Ember

Компоненты

- Переиспользуемые части приложения; функционально независимая часть системы
- Во фреймворках чаще кастомные элементы интерфейса
- Могут быть как простыми и глупыми (кнопки, иконки), так и большими (форма авторизации, целая страница)
- Иногда компоненты не связаны с UI, но также встраиваются в дерево компонентов
- Компоненты выстраиваются в иерархию



Vue.js

- JS фреймворк с открытым исходным кодом, появившийся в 2014 году
- Текущая версия 2 (почти вышла 3)
- Позволяет просто и эффективно разрабатывать компоненты на MVVM

Ссылки

- <https://ru.wikipedia.org/wiki/Model-View-Controller>
- <https://ru.wikipedia.org/wiki/Model-View-Presenter>
- <https://ru.wikipedia.org/wiki/Model-View-ViewModel>
- <https://vuejs.org>
- <https://reactjs.org>
- <https://angular.io>
- <https://emberjs.com>