1.8. Дерево компонентов. Однофайловые компоненты

Экземпляр Vue — это самая важная для понимания вещь, потому что он буквально контролирует раздел приложения. Чем тщательнее вы разберётесь в экземпляре, тем лучше поймёте более сложные вещи, например, как создаётся новый проект или как работают SFC (single-file component) — однофайловые компоненты.

Каждое приложение начинается с создания нового экземпляра Vue с помощью функции createApp:

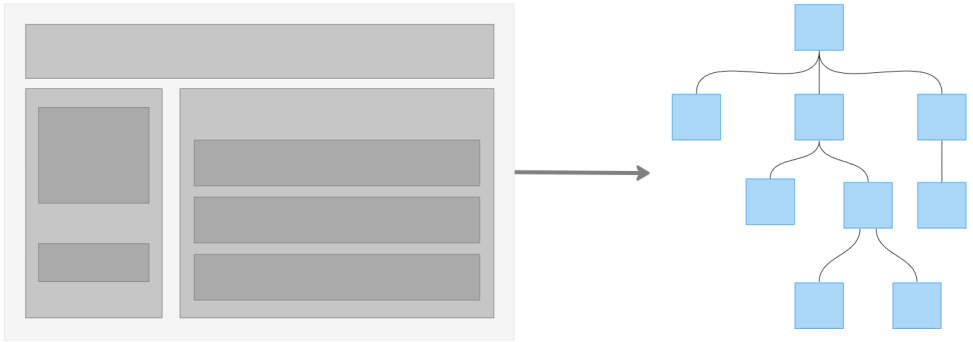
```
import { createApp } from 'vue'

const app = createApp(...)
```

Этот экземпляр Vue мало что делает. Фактически он вообще ничего видимого не делает: если бы вы добавили это на веб-страницу, то ничего бы не увидели. Однако Vue уже инициализирован и готов к использованию.

Хотя Vue не реализует паттерн MVVM в полной мере, архитектура фреймворка им во многом вдохновлена. Поэтому переменную с экземпляром Vue традиционно именуют vm — сокращённо от ViewModel.

Приложение Vue часто реализует паттерн Component Decomposition и состоит из корневого экземпляра Vue, создаваемого с помощью `createApp`, опционально организованного в дерево вложенных, повторно используемых компонентов.



Дерево компонентов

Например, у вас могут быть компоненты для заголовка, боковой панели, зоны контента, каждый из которых может содержать другие компоненты для навигационных ссылок, постов блога и так далее.

Каждый компонент хранится в файле с расширением `.vue`. У него есть три ключевые составляющие:

- template (шаблон);
- script (скрипт);
- style (стили).

Вместе они реализуют подход Vue.js, который называется **Single File Component** (SFC) — однофайловый компонент. То есть мы не разделяем код по типу: отдельно HTML, отдельно JS-скрипты и отдельно стили. Мы разделяем код по ответственности. В файле находится всё необходимое для изолированного компонента: вёрстка — *шаблон*, JS-логика — *script*, стили — *style*. Это очень удобно.

Внутри компонента его шаблон, логика и стили неразрывно связаны между собой, что позволяет сделать компонент более сплочённым и удобным в поддержке.

Vue 3 принёс несколько изменений в однофайловые компоненты по сравнению со второй версией.

Множественные корневые элементы компонента

Представим, что компонент содержит два блока `<div></div>`.

Если бы мы использовали Vue 2, то нам пришлось бы обернуть эти блоки в ещё один `<div></div>`, поскольку расположить множество элементов на верхнем уровне компонента было бы невозможно из-за ограничений самого фреймворка.

```
<template>
  <div>
    <div>
      Блок 1
    </div>

    <div>
      Блок 2
    </div>
  </div>
</template>
```

Однако у Vue 3 нет такого ограничения, он легко позволят реализовать нашу идею.

```
<template>
  <div>
    Блок 1
  </div>

  <div>
```

```
    Блок 2
  </div>
</template>
```

Composition API из коробки

Версии Vue 2 до 2.7 не включали Composition API и требовали установки плагина.

Script setup

Блок `<script>` теперь имеет директиву `setup`, которая является не более чем синтаксическим сахаром для использования Composition API. Однако использование `<script setup>` позволяет писать компоненты в формате, наиболее приближенным к натуральному JavaScript.

Обо всех изменениях мы подробно расскажем в последующих статьях.

Песочницы

Можно практиковаться или экспериментировать с компонентами Vue без дополнительной установки нового проекта. Для этого есть разные песочницы, среди которых мы рекомендуем:

- Vue SFC Playground,
- CodeSandbox.

Гид по написанию компонентов

Обязательно прочитайте и держите «под рукой» официальный гид по написанию компонентов.