

О популярных фреймворках, использующих компонентный подход для разработки сложных веб-интерфейсов

Н. Н. Вилейто, И. М. Пости

Научный руководитель: Н. М. Воронова

Владимирский государственный университет им. А. Г. и Н. Г. Столетовых, г. Владимир

Постановка проблемы

Интернет технологии постоянно развиваются. Сейчас очень много веб-сайтов являются достаточно сложными веб-приложениями, разрабатывать и поддерживать которые гораздо сложнее, чем сайты периода начала становления интернета. Из-за масштабов и сложности современных веб-приложений, использование только базовых веб-технологий (HTML, CSS JavaScript) является недостаточным.

Основные требования к разработке современных веб-интерфейсов можно сформулировать так:

- код должен быть хорошо структурирован, быть модульным и масштабируемым;
- в условиях динамического контента должно быть грамотно организовано управление состоянием приложения – изменение данных как правило, сопровождается обновлением интерфейса, и в приложении должен быть реализован механизм такого обновления;
- должен быть реализован механизм, решающий проблему смешения стилей: при большом количестве css-классов сложно обеспечить уникальность их имён и избежать коллизии стилей.

Все эти требования выполняются при использовании компонентного подхода: веб-приложение представляется как совокупность отдельных независимых частей – компонентов, которые включают в себя структуру элемента интерфейса и логику его работы. Взаимодействие компонентов друг с другом реализуется с помощью событий, при этом взаимодействие приложения с сервером выносится на отдельный уровень.

Реализовать компонентный подход, можно напрямую используя HTML templates и Shadow DOM или на базе веб-фреймворков. Использование фреймворков предпочтительней, так как они предоставляют уже готовую компонентную структуру приложения.

Цель работы

Целью работы является анализ наиболее популярных веб-фреймворков для разработки веб-приложений, использующих компонентный подход. В результате анализа должны быть выявлены особенности использования этих фреймворков, предоставляемые ими возможности, их преимущества и недостатки.

Базовые вопросы для исследования

1. Обзор типичных проблем, возникающих при разработке сложных веб-приложений.
2. Принципы компонентного подхода, модель взаимодействия данных и компонентов в веб-приложении.
3. Обзор задач, решаемых веб-фреймворками, использующими компонентный подход.
4. Анализ веб-фреймворков Angular, React и Vue.

Результаты проведённого исследования

1. Рассмотрены принципы компонентного подхода. Обоснована целесообразность использования веб-фреймворков, поддерживающих компонентный подход для разработки веб-приложений.
2. Проведён анализ популярных фреймворков Angular, React и Vue. Сравнение проводилось по критериям:

- простота реализации и порог вхождения для разработчиков;
- масштабируемость;
- удобство и эффективность управления данными и состоянием приложения;
- удобство и эффективность управление CSS-стилями;
- простота использования средств разработчика, предоставляемых рассматриваемым фреймворком;
- сложность управления конфигурацией проекта, его сборкой и развёртыванием.

В результате анализа выявлены отличия в использовании каждого из исследуемых фреймворков, их преимущества и недостатки.