

Лекция №5

Фреймворки для модульного тестирования JavaScript

- JavaScript – это язык программирования, который определяется как динамический и интерпретируемый язык высокого уровня, используемый в веб-приложениях HTML.
- JavaScript также используется не для веб-документов, таких как PDF-файлы и виджеты на рабочем столе, и стал популярным для серверных веб-приложений.
- JavaScript является объектно-ориентированным скриптом и следует прототипу

Модульное тестирование JavaScript

Юнит-тесты JavaScript обычно запускаются в браузере или на веб-интерфейсе. Тестовый код JavaScript написан для страницы сайта или отдельного модуля приложения, а затем этот код объединяется с HTML в качестве встроенного обработчика событий. Эти модульные тесты организованы один за другим в комплекте. Каждый набор содержит количество тестов, предназначенных для выполнения для отдельного модуля. Наиболее важно, что они не конфликтуют с любым другим модулем и работают с меньшим количеством зависимостей друг от друга (некоторая критическая ситуация может вызвать зависимости).

Проблемы в модульном тестировании JavaScript

Есть определенные проблемы, которые можно найти при выполнении модульного тестирования JavaScript, такие как;

1. Многие другие языки поддерживают модульное тестирование в браузерах, как в стабильной, так и в среде выполнения, но JavaScript не может
2. Вы можете понять некоторые системные действия с другими языками, но это не так с JavaScript
3. Некоторые JavaScript, написанные для веб-приложения, могут иметь несколько зависимостей
4. JavaScript хорошо использовать в сочетании с HTML и CSS, а не в Интернете

5. Трудности с рендерингом страниц и манипулированием DOM

Чтобы избежать таких проблем, вы можете это сделать;

1. Не используйте глобальные переменные
2. Не манипулируйте предопределенными объектами
3. Разработка основных функций на основе библиотеки
4. Попробуйте создать небольшие части функциональности с меньшими зависимостями

Фреймворки для модульного тестирования JavaScript

Иногда на экране появляется сообщение об ошибке, например «Невозможно загрузить example.js» или любая другая ошибка JavaScript, связанная с контролем версий, эти уязвимости подпадают под юнит-тестирование JavaScript.

Мы рассмотрим некоторые инструменты и платформы, которые используются для выполнения модульного тестирования JavaScript.

1. [Unit.js](#) : он известен как библиотека утверждений с открытым исходным кодом, работающая в браузере и Node.js. Он чрезвычайно совместим с другими средами модульного тестирования JavaScript, такими как Mocha, Karma, Jasmine, QUnit, Protractor и т. Д. Предоставляет полный документированный API списка утверждений.

2. [QUnit](#) : используется как для клиентского, так и для серверного модульного тестирования JavaScript. Этот бесплатный фреймворк используется для проектов jQuery. Он следует общей спецификации JS модульного тестирования для модульного тестирования. Он поддерживает График долгосрочной поддержки узла.

3. [Jasmine](#) : Jasmine – это основанная на поведении среда разработки для модульного тестирования JavaScript. Он используется для тестирования как синхронного,

так и асинхронного кода JavaScript. Он не требует DOM и поставляется с простым синтаксисом, который может быть написан для любого теста.

4. [Карма](#) : Карма – это продуктивная среда тестирования с открытым исходным кодом. Простое управление рабочим процессом Запуск в командной строке. Предоставляет свободу написания тестов с помощью Jasmine, Mocha и QUnit. Вы можете запустить тест на реальных устройствах с простой отладкой.

5. [Мокко](#) : Мокко работает на Node.js и в браузере. Mocha выполняет асинхронное тестирование более простым способом. Обеспечивает точность и гибкость в отчетности. Обеспечивает огромную поддержку разнообразных функций, таких как время ожидания для конкретного теста, API-интерфейсы JavaScript и т. Д.

6. [Jest](#) : Jest до сих пор используется Facebook для тестирования всего кода JavaScript. Он обеспечивает опыт тестирования «нулевой конфигурации». Поддерживает независимый и непрерывный тест без каких-либо конфликтов. Не требуется никаких других настроек конфигурации и библиотек.

7. [AVA](#) : AVA – это простая среда модульного тестирования JavaScript. Тесты проводятся параллельно и последовательно. Параллельные тесты запускаются без перерыва друг друга. AVA также поддерживает асинхронное тестирование. AVA использует подпроцессы для запуска теста.

Резюме

- Модульное тестирование JavaScript может иногда становиться утомительным и хитрым, поскольку оно выполняется в основном для внешнего интерфейса. Можно использовать библиотеки JS, чтобы добавить немного легкости. Задача может стать больше, поскольку JavaScript включается в Node.js и TypeScript.

- При выполнении теста вы должны помнить три вещи: Функция, которая должна быть проверена, окончательный результат и ожидаемый результат. Некоторые инструменты и рамки могут помочь вам в выполнении этой задачи. Вышеупомянутые списки инструментов упоминаются с большинством популярных и полезных сред, используемых для модульного тестирования JavaScript.

- Более того, в связи с предстоящими проблемами в проведении тестирования, в будущем могут появиться более мощные платформы и инструменты.