

СУЧАСНІ ІНСТРУМЕНТИ JS-РОЗРОБНИКА

К. О. Тимошенко, студент групи КМ-17-Зск

О.Є. Тесленко, асистент кафедри ПКСМ

Центральноукраїнський національний технічний університет

Популярність JavaScript привела до створення динамічної екосистеми технологій, структур і бібліотек. З кожним роком збільшується кількість допоміжних програмних продуктів для більш зручної розробки програмного продукту. Які ж технології будуть дійсно корисними сучасному JS-розробнику?

Почнемо з деяких інструментів командного рядка.

Команда *np* являє собою гарну альтернативу *npm*. Якщо необхідна публікація розробок в *npm*, можливе застосування *np*, так як це засіб командного рядка значно спрощує процес управління версіями, допомагає працювати з тегами *git*-релізів і публікувати пакети в *npm*, особливо якщо трапляється так, що ви займаєтесь підтримкою більш ніж пари модулів. Крім того, якщо широко застосовується утиліта *np*, буде корисним використанн я інструменту *release*, створений *Zeit*. Він призначений для автоматизації публікації релізів *GitHub*-проектів.

Yarn - це менеджер пакетів, сумісний з *npm* і відрізняється поліпшеними можливостями.

Хоча *npm* v5 набагато швидше, ніж його попередні версії, але *yarn* переважає за швидкістю і стабільністю. У будь-якому випадку, *yarn* працює з тими ж базами даних модулів, що і *npm*, при цьому не можна з упевненістю сказати - що краще - *yarn* або *npm*. Виходячи з цього тому, хто займається JS-розробкою, варто освоїти і *npm*, і *yarn*.

Prettier - це засіб для форматування коду.

Prettier дозволяє підтримувати однаковий стиль коду, приводячи тексти програм до виду, заданому його правилами, які, зокрема, беруть до уваги максимально допустиму довжину рядка, переносячи код на новий рядок там, де це потрібно.

Розробники витрачають занадто багато сил, розмірковуючи про подання і стилізації коду, в той час як *prettier* усуває необхідність подібних роздумів. Автоматичне форматування коду дозволяє програмістам зосередитися на тому, що вони пишуть, і не думати про те, як це має виглядати.

Веб-скрапінг

Існує безліч відмінних утиліт для веб-скрапінга, деякі з них, наприклад, *cheerio*, працюють на рівні вихідного HTML-коду, а деякі, такі як *puppeteer*, дозволяють обробляти сторінки засобами повноцінного браузера. Що саме вибрати - залежить від цілей конкретного проекту, так як робота з HTML набагато швидша і вимагає менше системних ресурсів, в той час як автоматизація браузера без користувальницького інтерфейсу складніша і вимагає більш серйозної попередньої підготовки.

Cheerio - це швидка, гнучка і мініатюрна реалізація основних можливостей *jQuery*, створена спеціально для сервера і призначена для роботи з HTML.

Cheerio - це відмінний інструмент для швидкого чорнового веб-скрапінга, відповідний для розбору HTML-коду сторінок. Це засіб відрізняється зручним синтаксисом, схожим на *jQuery* і призначене для обходу HTML-документів і виконання маніпуляцій з ними. *Cheerio* добре поєднується з бібліотекою *request-promise-native*, призначеної для завантаження віддалених HTML-документів.

Puppeteer - це Node-API для керування браузером *Chrome* без користувальницького інтерфейсу.

На відміну від *cheerio*, *puppeteer* - це обгортка для автоматизації *Chrome* без користувальницького інтерфейсу, яка дуже корисна для роботи з сучасними веб-додатками, в

тому числі - лише однієї сторінки, заснованими на JavaScript. Так як робота ведеться з браузером Chrome, puppeteer характеризується високим рівнем підтримки сучасних веб-Стандарт. Це впливає на можливість правильного формування сторінок і виконання скриптів. Браузер Chrome без користувальницького інтерфейсу - інструмент порівняно новий, але він, мабуть, поступово витіснить більш старі засоби, такі як PhantomJS.

Бібліотеки для Node.js

Dotenv-safe дозволяє завантажувати змінні оточення з файлу `.env` і забезпечує наявність всіх необхідних змінних.

Цей модуль розширює вельми популярний модуль `dotenv`, підтримуючи вказівку очікуваних змінних оточення за допомогою `.env.example`. Як і базова бібліотека, `dotenv-safe` забезпечує швидку, безпечну і надійну роботу зі змінними оточення в середовищі Node.js.

Request і request-promise-native

Бібліотека *request* і заснована на ній бібліотека `request-promise-native`, оснащується її підтримкою *ES6-Проміс*, являють собою клієнти, які спрощують виконання HTTP-запитів.

Виконання HTTP-запитів - це операція, яка потрібна практично завжди і скрізь. Я для виконання цієї операції користуюся модулем `request-promise-native`. Відбувається це в 95% випадків, коли потрібно організувати очікування результату HTTP-запиту, обгорненого в проміс. В інших 5% випадків, коли потрібно працювати з потоком відповіді безпосередньо, я користуюся модулем `request`, який Проміс не підтримує.

Consolidate - це бібліотека, що забезпечує однакову роботу з різними шаблонизатор для Node.js.

`Consolidate` відмінно підходить для роботи з будь-якими серверними шаблонами. Наприклад - це шаблони електронних листів, твітів, HTML-сторінок.[1]