

УДК 004.4'242

*Васильчук Д. П., студент, гр. КІ-1,  
Оринчак І. А., старш. викладач кафедри  
Житомирський державний технологічний університет*

## **ВИКОРИСТАННЯ WEBPACK ДЛЯ СТВОРЕННЯ СУЧАСНОГО FRONTEND-У**

Webpack – це статичний модульний збирач для додатків на JavaScript. Офіційний сайт проекту: <https://webpack.js.org>. Програми, написані на JavaScript, постійно ускладнюються, тому виходом з цієї ситуації є використання збирача (або Бандлера). Подібні інструменти дозволяють розробникам упаковувати, компілювати і в цілому організувати всі ресурси, необхідні для проекту. Можна використовувати не тільки сторонні бібліотеки, а й власні файли. Подібна модульна система дозволяє домогтися кращої організації проекту, так як виходить, що він розбитий на невеликі модулі.

Webpack на даний момент є одним з найпотужніших подібних інструментів. Він має відкритий вихідний код і дозволяє вирішувати безліч різних задач. Як і інші інструменти розробника, Webpack має свої плюси і мінуси.

До плюсів можна віднести те, що він відмінно підходить для роботи з односторінковими додатками (SPA – Single Page Application). Також Webpack може здійснювати поділ коду. Через ці та інші плюси зараз він є одним з найбільш популярних інструментів JS-розробки.

З мінусів можна відмітити, що спочатку трохи складно розібратися в його роботі, частина документації застаріла через велику кількість змін у вихідному коді.

Webpack можна встановити за допомогою менеджера пакетів, наприклад npm:

```
npm install --save-dev webpack
```

Webpack дуже гнучкий в налаштуванні інструмент. Для того, щоб почати працювати з ним, необхідно познайомитися з чотирма базовими поняттями.

- Entry – вхід
- Output – вихід
- Loaders – завантажувачі
- Plugins – плагіни

Під Entry (вхід) мається на увазі точка входу (entry point), яку Webpack буде використовувати для побудови внутрішнього графа залежностей. Після введення точки входу Webpack зможе зрозуміти, які модулі і бібліотеки безпосередньо і не безпосередньо зв'язуються.

В результаті кожна залежність перетворюється в файли, які називаються бандл (bundles – перекладається як пакети або вузли).

Output (вихід) вказує, де Webpack повинен розміщувати створені бандли, і як він буде називати ці файли (за замовчуванням це ./dist). Налаштувати цю частину процесу можна в полі output в конфігурації. За ім'я файлу відповідає output.filename, а output.path – за місце, директорію, де буде знаходитися зібраний файл.

Завантажники (Loaders) дозволяють Webpack'у обробляти не тільки файли JavaScript, тому що сам по собі Webpack розуміє тільки мову JS.

Завантажники трансформують всі типи файлів в модулі, які потім можна додати в граф залежностей вашого додатку (а значить, і в бандл).

Використання завантажувачів має дві мети:

- властивість test визначає, який файл/файли повинні бути трансформовані;
- властивість use вказує, який завантажувач повинен використовуватися для виконання трансформації.

Правила (rules) визначаються для певного модуля і використовуються властивості test і use. Компілятор Webpack'а зрозуміє, що коли він дійде до файлу який вказаний у блоці „rules”, після властивості „test:” перед додаванням в бандл йому потрібно буде використовувати те, що буде записано у кодї після властивості „use:”.

Плагіни (Plugins) – дозволяють виконувати досить широкий спектр функцій. Якщо завантажувачі використовуються для трансформації певних типів модулів, то плагіни можуть бути використані для виконання набагато більш широкого списку задач.

Для того, щоб використовувати плагін, необхідно використовувати require() і додати його в масив плагінів plugins: []. Більшість плагінів можна налаштувати. Так як один плагін може використовуватися кілька разів для різних цілей, необхідно створити кілька окремих екземплярів, використавши оператор new.

Багато плагінів поставляються відразу з Webpack'ом з коробки.

Враховуючи все вищевказане, можна з впевненістю говорити, що Webpack – корисний перспективний програмний продукт, гнучкий до різноманітних налаштувань, який суттєво спрощує розробку сучасного frontend'у.