IT doupě

# Jak se tvoří web – Balíčky, bundling a publikace

Dnešní díl se bude od těch předchozích trochu lišit. Poprvé totiž nebudeme dělat nic se stránkou samotnou, ale zprovozníme technologie, bez kterých by se v budoucnosti jen těžko obešla.

Začněme balíčky. K jejich správě se používá příkaz npm a díky němu je můžeme ze stejnojmenné stránky (https://www.npmjs.com/) importovat do projektu. Tam pro jeho fungování stačí zadat příkaz npm init, čímž vygenerujete package.json obsahující info a knihovny použité v projektu – ty se instalují pomocí npm install <balíček>. Mezi nejpopulárnější z nich patří například moment (práce s časem), lodash (práce s objekty), React (již brzy), nebo třeba Webpack, který stáhneme jako první, a hned vysvětlím proč.

Pokud půjdete na téměř jakoukoliv moderní stránku (například Instagram) a pomocí F12 si rozbalíte její kód, celkem rychle přijdete na to, že je téměř nečitelný. A pokud ho nějaký šílenec netvořil ručně, je na vině bundler, mezi které Webpack spadá. Ten totiž dokáže vzít soubory, které v projektu používáte, podle jejich typu je zpracovat, osekat nepotřebné věci jako mezery, nové řádky, nebo jména funkcí, a následně vyprodukovat jeden bundle.js soubor obsahující celou stránku, který pro správné fungování stačí zmínit někde ve vašem HTML souboru.

Vše, co Webpack dělá, se dá nakonfigurovat ve webpack.config.js souboru. Potřebujete zadat vstup, ve kterém si pak najde importy a pomocí stromu závislostí je rekurzivně zpracuje. Mezi pravidla jsem zadal, aby pro styly používal css-loader a díky HtmlPluginu můžu společně s bundlem exportovat i HTML soubory, ke kterým se automaticky připojí. Po spuštění příkazu webpack se všechny soubory „zkompilují“ do složky dist. To, co je v ní, následně může jít na produkci.

No jo, ale jak? Teoreticky můžete stránku přímo otevřít v průzkumníku souborů a bude fungovat. Tím ale ztrácíte možnost tvořit vlastní cesty a vykonávat akce na serveru. Proto je lepší využít některou z volně dostupných knihoven na již zmíněném npm. V budoucnosti vše zařídí Next.js, prozatím jsem si vybral express.js, protože je poměrně jednoduchý na konfiguraci a umí vše co budeme potřebovat. Například reagovat na neplatné cesty nebo možnost spustit server na jakémkoliv portu. Nezapomeňte na některou z cest nastavit i bundle.js, jinak se ke klientovi vůbec nedostane.

A to bude vše, zdroják k dnešnímu dílu najdete jako vždy u mě na GitHubu, a příští rok zase Informatice Zdar!