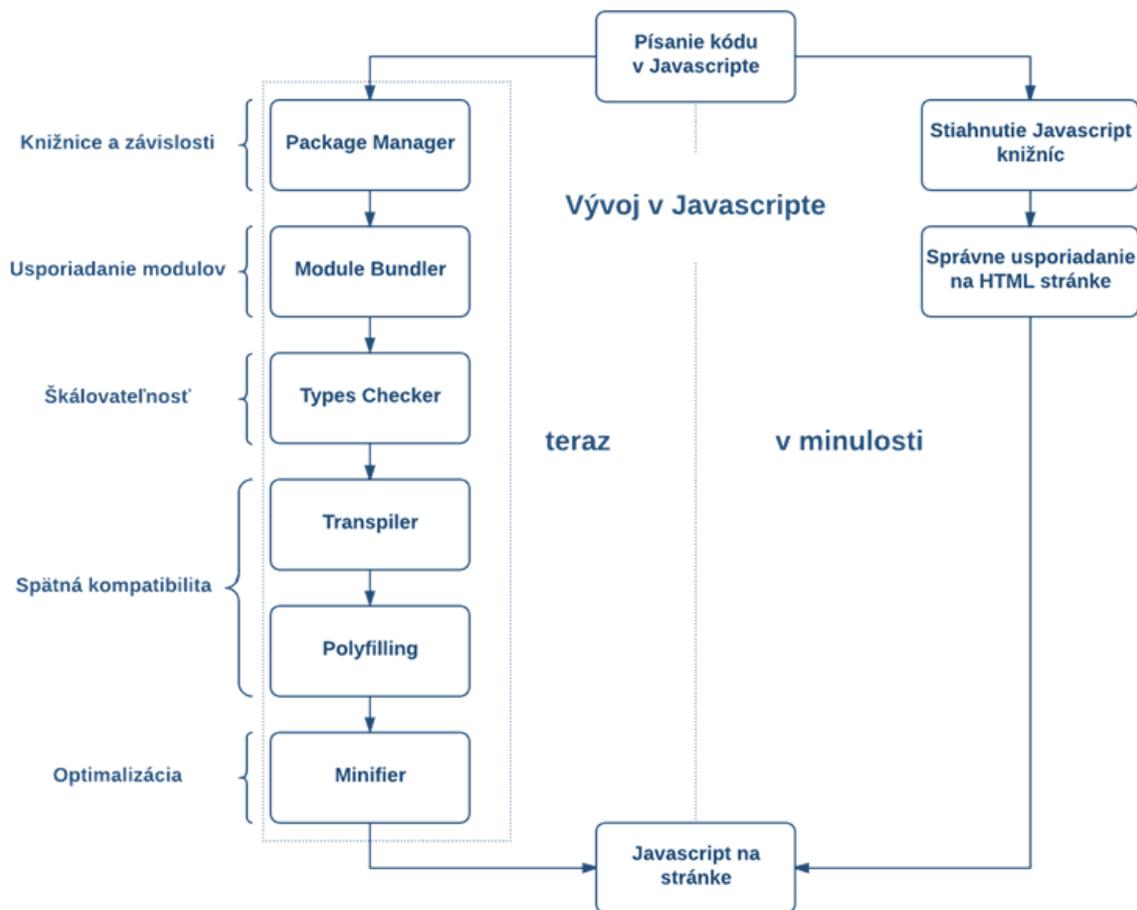


Support of the Javascript Applications Development on the SmartWeb Platform

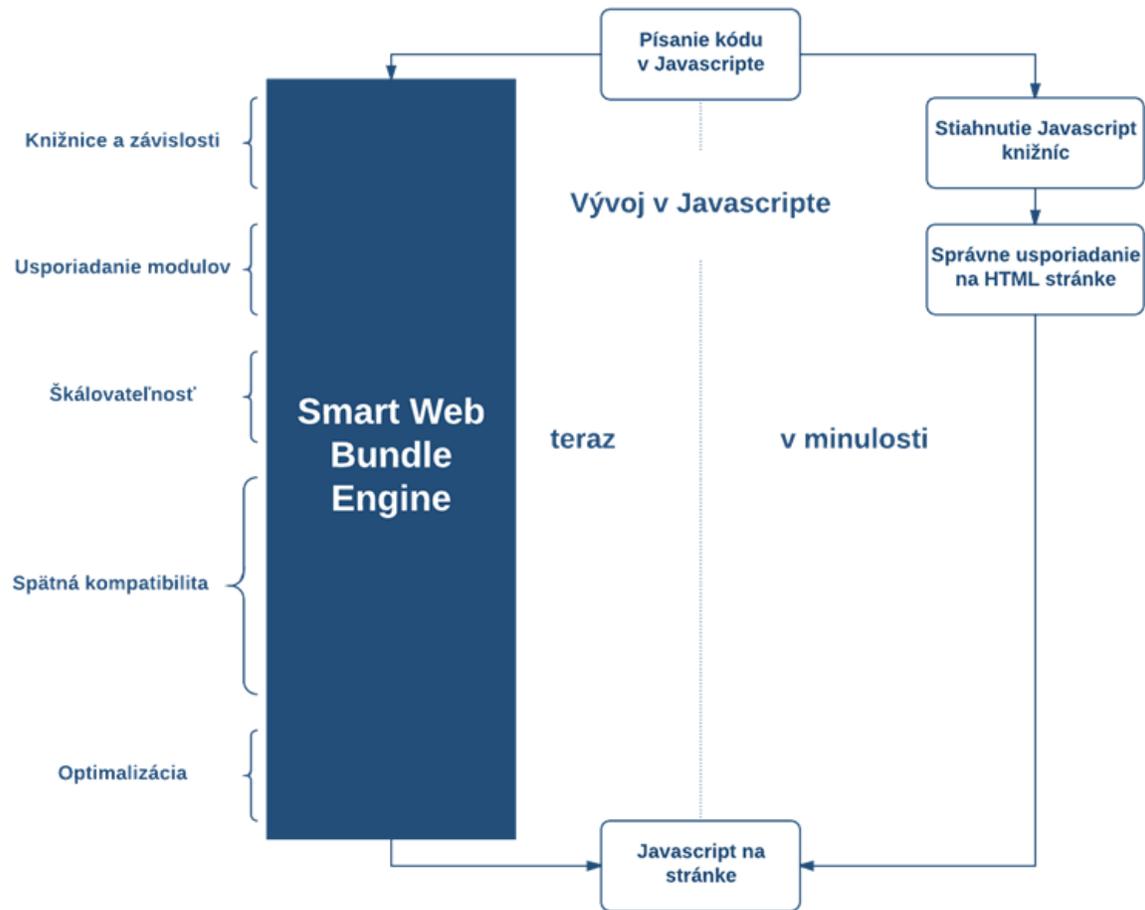
Development of Javascript applications has changed dramatically over the past years. The reason is mainly that the code capacity on web sites has expanded exponentially and Javascript itself has gone through major changes as a language. While in the past, it was enough to manually add own code or download library to a web site, this method is not practical when using modern applications nowadays. Reasons are the following:

- At present, it is normal that a Javascript application uses indirectly hundreds of libraries of third parties (used libraries need for their running other libraries and those libraries need another ones and so on). Organizing, solving conflicts of versions and correctly integrating such amount of libraries is not the task of a programmer anymore but an automated tool (*Package Manager/Module Bundler*).
- It is more advantageous to write an application with type extension over Javascript which is a dynamic language when having large code capacity. Before the code distribution on the web site, it is necessary for these type annotations to be checked by an automated tool (*Type Checker*) for unintentional type errors which should be removed from the code because of the browser that does not know them.
- For applications, it is necessary to have the support of older browsers (Internet Explorer 11) which are still used in company sphere. For the application written in a modern Javascript version to work also on older browsers, it is necessary for modern constructions of the language to be translated into older browser versions (*Transpiler*). At the same time, it is vital for the used API functions unsupported by an older browser to be emulated by a special code attached to the application code (*Polyfilling*).
- Since the capacity of the final code with the necessary libraries can be significantly large, it is desirable to reduce the code or else remove by a special tool (*Minifier*) the parts (e.g. functions) that are not used in the application.



That is why the development of modern web applications is realized in cooperation with more automated tools. Setting and mutual connecting of these tools, and thus the whole "development pipeline", from source code to production files is not trivial at all. It is not trivial also because of the large number of alternative tools for each of the individual steps, for example. Setting and configuration require vast experience with individual Javascript tools, their mutual integration, possibilities, advantages, and disadvantages.

One of the main aims of the SmartWeb platform is also the support of the Javascript applications' development. That is why over this platform, a programmer does not have to configure, install nor integrate any tool that transforms a source Javascript code into a final form interpreted by a browser. This is automatically realized by a special module of the SmartWeb platform - **Bundle Engine**.



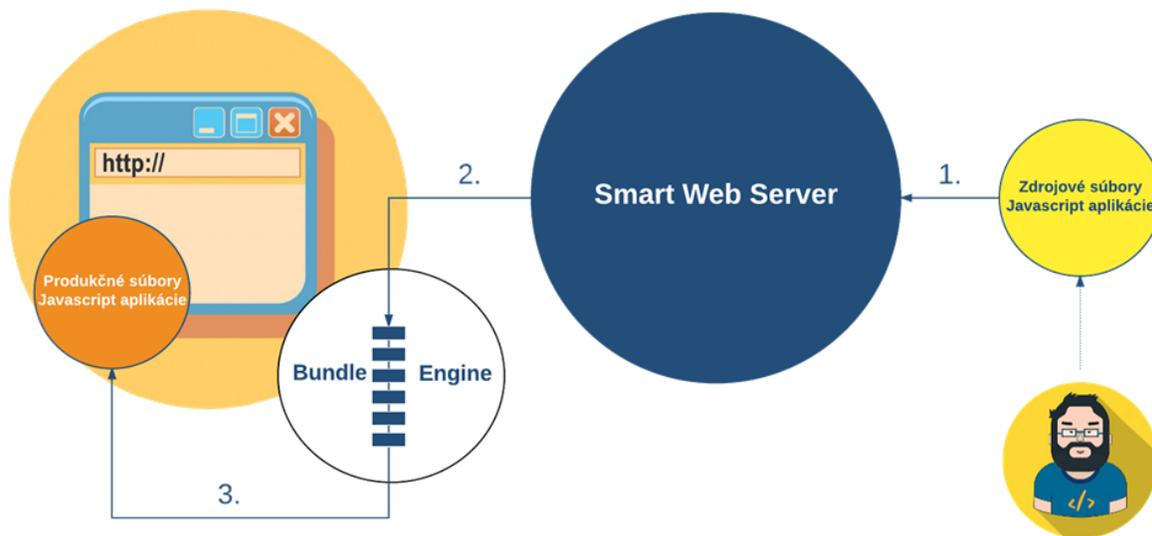
The Bundle Engine is an independent Javascript runtime that runs the mentioned tools. It is administrated, installed and upgraded automatically by the SmartWeb server from the Internet, that is why a programmer does not have to take care of it. In the case of an offline installation, it is also possible to install the Bundle Engine automatically. The Bundle Engine is multi-platform and supports the newest technologies in the web development area. Among the key features, there is:

- support of the newest standard of the Javascript ES 2017o language
- support of backward compatibility of the application to Internet Explorer 11
- automatic download of Javascript libraries from the public repository npmjs.com (more than 475 000 libraries available)
- support of quick debugging (code change is enough + refresh of a browser) by a unique combination of tools

Situations of the Bundle Engine Module Usage

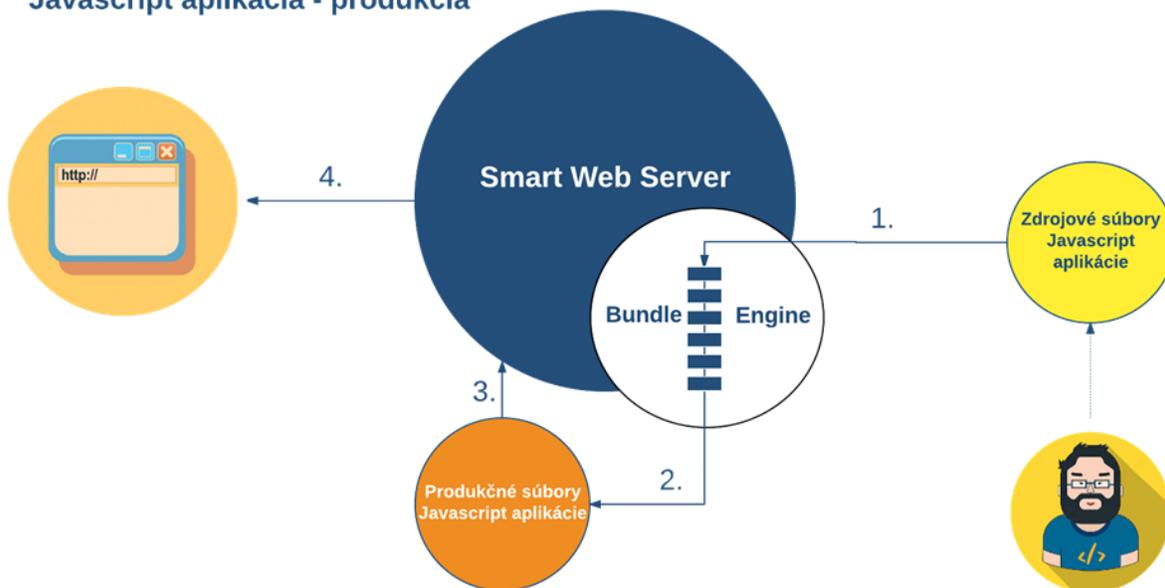
Since every tool used for transformation of source codes to productive in the Bundle Engine are implemented in Javascript, it is possible to execute this transformation also directly in the browser schematically displayed on the following picture. This way of running the application site is advantageous for debugging directly in a browser when we also have source code at disposal.

Javascript aplikácia - ladenie



We can use the Bundle Engine module not only in development and debugging but also in a situation when we want to support the possibility of ad hoc changes in production directly on the client's side. The following picture illustrates the possibilities of the Bundle Engine module usage for ad hoc transformation of source application files to production files which are distributed into a browser. A developer runs this transformation by so-called REBUNDLE command.

Javascript aplikácia - produkcia



There is an alternative possibility of application version distribution only with production files (meaning without source files). These files can be distributed in one archive together with the SmartWeb application (smartWeb.war).

Javascript aplikácia - produkcia, alternatíva

