**COMPILING - TRANSPILING**

Contents

[1. **WEBPACK** 2](#_Toc77355049)

[ **Entry:** 2](#_Toc77355050)

[ **Output:** 2](#_Toc77355051)

[ **Loaders:** 2](#_Toc77355052)

[ **Plugins:** 2](#_Toc77355053)

[ **Mode:** 2](#_Toc77355054)

1. **WEBPACK**

* **Entry:**
* the entry point is the module that webpack uses to start building its internal dependency graph.
* Thông thường sẽ trỏ đến src file.
* By default: ./src/index.js
* **Output:**
* the output property instructs webpack where to emit the bundle(s) and what name to use for the file(s).
* Đầu ra của webpack. Nơi để nó emit cái file bundle ra.
* By default: ./dist/main.js for the main bundle
* and ./dist for other generated files — such as images
* **Loaders:**
* Là cái sẽ đc đưa vào trước khi compile 1 file (Pre-process).
* **Loaders** work at the individual file level **during** or **before** the bundle is generated
* Thường là sẽ cần loaders cho những file non-js. Và chuyển nó thành dạng module vd: Typescript language to JavaScript, inline images, import CSS files directly from our JavaScript
* Theo webpack: mún process thì tất cả các filé file là dạng module và phải là ngôn ngữ js
* **Plugins:**
* **plugins** are used for any other task that **loaders** can’t do (mạnh hơn loaders).
* **deeply integrate** into webpack because they can register hooks and access (and modify) the compiler
* **Plugins** có thể tích hợp sâu vào bất cứ quá trình nào khi bundle
* Vd: asset management, bundle minimization, optimization
* **Mode:**
* Thường có 3 môi trường: **development, staging, production**.
* Trong webpack thì thông thường chỉ có 2: development, production
* This allows webpack to use built-in optimizations corresponding to each environment. Khi có define mode thì webpack sẽ dùng những cái có sẵn theo từng môi trường để tối ưu bundle.