Webpack

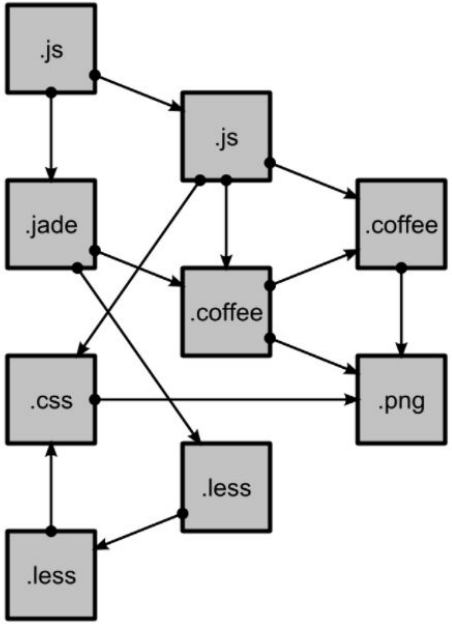
https://nathankrasney.com/

מה זה Webpack

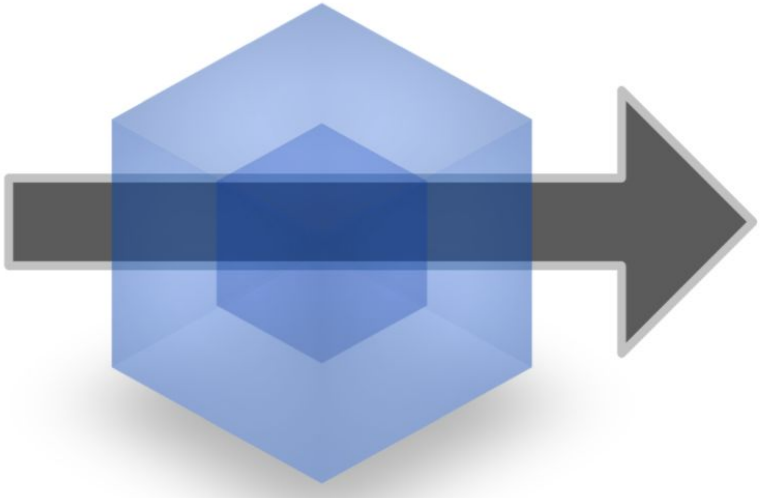
- Webpack הוא module bundler אשר נחשב לכלי De Facto בתעשיה
- webpack בעצמו הוא מודול ראו כאן
- Webpack לוקח מודולים עם תלויות והופך אותם לקבצים סטטיים
- webpack מבצע את האיחוד בצורה אופטימלית מבחינת ביצועים לדוגמא קבצים קטנים כך שזמן הטעינה יהיה מהיר
- לפעמים סדר טעינה של קבצי js , css חשוב - webpack יכול לעזור גם במקרה זה
- אפשר לכתוב קוד גרסת js מתקדמת כמו es6 כמו שימוש ב const , import,export,arrow function והוא מתרגם אותו לגרסה מובנת על ידי דפדפנים (ז"א ל es5 ??)

https://nathankrasney.com/

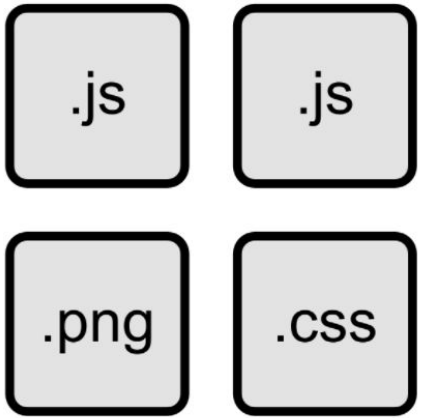
תיאור סכמטי של Webpack



modules
with dependencies



webpack
MODULE BUNDLER



static
assets

https://nathankrasney.com/

איך זה עובד

Webpack חייב לפחות נקודת אחיזה אחת להתחיל ממנה לדוגמא App.js ב React

Webpack משתמש ב loaders אשר מבצעים המרות של קבצים. לדוגמא
babel-loader מתרגם גרסה js מתקדמת לגרסה פחות מתקדמת ומתאימה לדפדפנים

Webpack משתמש בplugin אשר מאפשרים להזריק לוגיקה מבחוץ

Webpack יוצר את הפלט , עבור js זה יהיה קובץ js

https://nathankrasney.com/

דוגמא

package.json :

```
"build": "webpack --mode development --entry ./src/js/app.js --output ./dist/bundle.js",
```

זה הקובץ איתו webpack מתחיל

זו תוצאת השילוב של כל קובצי הjs ועם זה עובדים בקובץ html

app.js :

```
import {sum2} from './sum2'
export const sum3 = (num1,num2,num3) => {
    return sum2(num1,num2)+num3;
}
```

יש שימוש ב import/export על מנת לנהל טוב יותר תלויות בין קבצים ושימוש במודולים אבל זה ES6 webpackו מתרגם ל ES6 ל dist

Sample here

https://nathankrasney.com/

המשך דוגמא - יצירת גרסה ל production

package.json :

```
"build:prod": "webpack --mode production --entry ./src/js/app.js --output ./dist/bundle.js",
```

אפשר להשתמש באיזה טקסט שרוצים

ייצור קובץ minified מתאים ל production לדוגמא מבחינת גודלו

Sample here

שרת לוקאלי

כל מה שנאמר על webpack בשקף הקודם נכון אבל בנוסף הוא גם מאפשר לנו להרים שרת node לוקאלי ראה כאן וזה חשוב :

- בגלל בעיות CORS
- כדי לראות איך האפליקציה מתפקדת עם שרת (אפשר לבדוק ב chrome dev tool network tab)

package.json

```
"build:server": "webpack-dev-server --mode development --entry ./src/js/app.js --output ./dist/bundle.js",
```

אפשר לפתוח בדפדפן http://localhost:8080/
שקובץ ה html נטען לדפדפן על ידי השרת המקומי. איך ידע לטעון את קובץ הhtml ??

Sample here

https://nathankrasney.com/

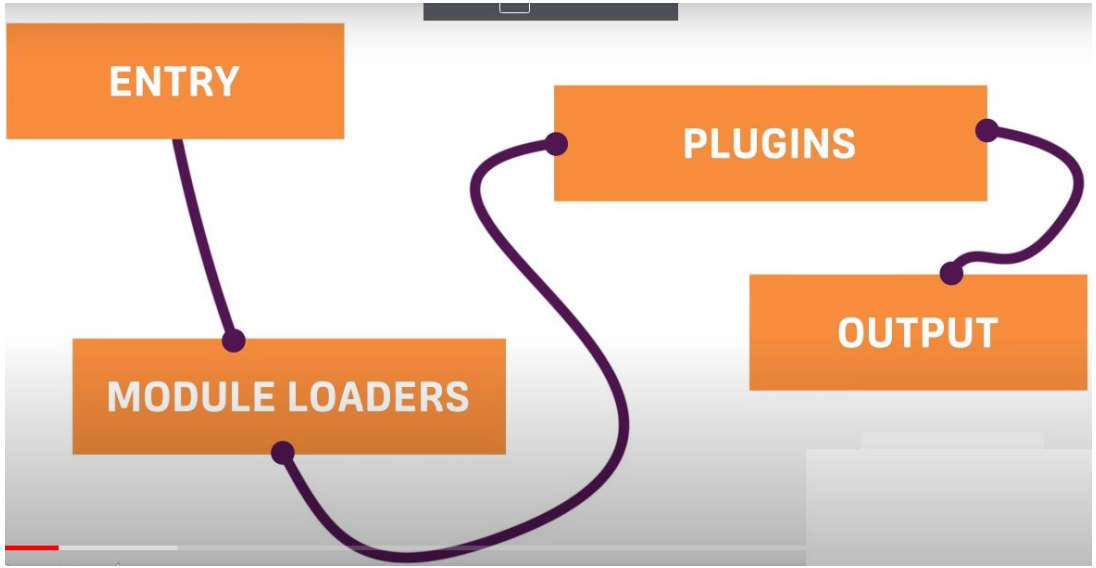
עקרונות מרכזיים - webpack

נקודת כניסה ז"א מאיפה להתחיל - ממומש על ידי entry--

נקודת סיום ז"א איפה לשים את התוצאה - ממומש על ידי output--

module loaders - מאפשר לשנות את הקוד

plugin -



https://nathankrasney.com/

קובץ קונפיגורציה

במקרים פשוטים אפשר להסתדר כמו שעשינו דרך scripts ב package.json אבל במקרים מורכבים יותר משתמשים בקובץ קונפיגורציה בשם webpack.config.js אשר ממוקם ב root.
הקובץ הזה מייצא אובייקט והיצוא נעשה בעזרת התחביר של node.js

```
const path = require('path')
module.exports = {
    entry: "./src/js/app.js",
    output : {
        // here the path must be absolute and filename can be any
        path:path.resolve(__dirname,'dist'),
        filename : 'bundle.js'
    }
}
```

https://nathankrasney.com/

קובץ הקונפיגורציה
webpack.config.js וזה מאפשר
לקצר ב package.json ראה בדוגמא

path יחסי לקובץ הקונפיגורציה

Max has added publicPath
- is it needed ??

דוגמת קוד כאן

Module and loaders

רשימה של loaders נמצאת כאן . פועלים per file

רשימה של plugin נמצאת כאן. פועלים על ה bundle הסופי

לדוגמא style-loader , css-loader שהם מודולים אותם מתקינים

npm i css-loader style-loader --save-dev

https://nathankrasney.com/

דוגמא - טעינת קבצי css לקובץ js

עד עכשיו היה לנו ב index.html שימוש בקבצי css בצורה הסטנדרטית בתוך אלמנט head

```
<link rel="stylesheet" href="./src/css/orange.css">
<link rel="stylesheet" href="./src/css/red.css">
```

אבל אפשר לטעון אותם ישירות לקובץ js באמצעות import (נעשה הרבה ב create-react-app)

וזאת נעשה בעזרת style-loader css-loader אותם נתקין ובעזרת הכנסת הקוד הבא ל app.js

```
import '../css/orange.css'
import '../css/red.css'
```

ועדכון webpack.config.js כפי שמופיע בדוגמא

דוגמת קוד כאן

https://nathankrasney.com/

המשך דוגמא - טעינת קבצי css לקובץ js

```
const path = require("path");
module.exports = {
  entry: "./src/js/app.js",
  output: {
    // here the path must be absolute
    path: path.resolve(__dirname, "dist"),
    filename: "bundle.js",
  },
  module: {
    rules: [{
        test: /\.css$/,
        use: ["style-loader" , "css-loader"],
      },],},
};
```

regular expression אשר קובע לאיזה קבצים להתייחס. במקרה זה - קובצי css

סדר טעינת מודולים. הסדר הוא מסוף המערך להתחלה

https://nathankrasney.com/

references

WHAT IS WEBPACK, HOW DOES IT WORK? | Webpack 2 Basics Tutorial - 2017 Max part1

USING THE WEBPACK DEV SERVER | Webpack 2 Basics Tutorial - 2017 max part 2

THE WEBPACK CORE CONCEPTS | Webpack 2 Basics Tutorial - 2017 max part 3

BASIC BABEL + SCSS WORKFLOW | Webpack 2 Basics Tutorial - 2017 max part 4

https://nathankrasney.com/