ლაბორატორიული 10

1. **Webpack**

Webpack არის თანამედროვე module bundler-ი. მას გაცილებით ფართო შესაძლებლობები აქვს ვიდრე browserify-ს.

დაყენება:

npm install --save-dev webpack webpack-cli

ბილდის გაშვების გასამარტივებლად package.json-ში ვამატებთ შემდეგ კოდს:

"scripts": {

"build": "webpack --config webpack.config.js"

}

webpack-ის გამოყენებისთვის სასურველია შევქმნათ კონფიგურაციის ფაილი webpack.config.js. მისი მინიმალური ვერსია შეიძლება გამოყურებოდეს შემდეგნაირად:

const path = require('path');

module.exports = {

entry: './src/index.js',

output: {

filename: 'main.js',

path: path.resolve(\_\_dirname, 'dist')

}

};

იმისთვის რათა შესაძლებელი იყოს სტილების იმპორტი javascript მოდულების მსგავსად საჭიროა შემდეგი მოდულების დაყენება:

npm install --save-dev style-loader css-loader

და webpack-ის კონფიგში შემდეგი კოდის დამატება:

module: {

rules: [

{

test: /\.css$/,

use: [

'style-loader',

'css-loader'

]

}

]

}

ამის შემდეგ ვებ აპლიკაციაში js ფაილებში შესაძლებელია სტილების იმპორტი შემდეგი ფორმით:

import './style.css';

es6 ფუნქციონალის გამოსაყენებლად საჭიროა babel-loader-ის დაყენება:

npm install -D babel-loader @babel/core @babel/preset-env

და კონფიგში ჩამატება:

module: {

rules: [

{

test: /\.m?js$/,

exclude: /(node\_modules|bower\_components)/,

use: {

loader: 'babel-loader',

options: {

presets: ['@babel/preset-env'],

plugins: ['@babel/plugin-proposal-object-rest-spread']

}

}

}

]

}

**დავალება L1001:**

წინა ლაბორატორიულში შესრულებული დავალება გადააკეთეთ ისე რომ browserify-ს ნაცვლად იყენებდეს webpack-ს.