***Webpack presentation***

Hi guys. Now I will tell you about the webpack.  
  
Webpack - is a tool, which allows to compile, for example, JavaScript modules into a single JS-file. Webpack is also known as **a module builder**.

When there are a lot of files, it creates one large file to run your application.

**CHANGE SLIDE**

It is also capable of performing many other operations:

**POINTING TO THE SLIDE**

* helps bring together your resources
* monitors changes and reruns tasks
* can transpile next generation JavaScript to older JavaScript standard ( ES 5)
* can transpile CoffeeScript to JavaScript
* can convert inline images to data : URI
* allows you to use require for CSS files
* can split output file into several files, to avoid slow loading pages from -  large size of JS-file.

Webpack is a more targeted tool. You just need to specify the entry point to your application, and webpack will analyze files and combine them into a single output JavaScript-file, containing everything you need to run the application.

**CHANGE SLIDE**

Webpack can be installed globally or locally for each project.

***Global installation***

To install, we can use yarn and whether npm

To install globally using Yarn, run this command:

**POINTING TO THE SLIDE**

If you are using npm, for install you need to run this command:

**POINTING TO THE SLIDE**

**CHANGE SLIDE**

***Local installation***

Now let's talk about local installation. It is recommendable, since Webpack can be updated every project and you have a less of a problem, when you use the latest features for a small project, rather than continuous update of all projects, which use Webpack.

To install locally using Yarn, run this command:

**POINTING TO THE SLIDE**

Accordingly, for npm we use this command:

**POINTING TO THE SLIDE**

**CHANGE SLIDE**

***Setting up Webpack***

**CHANGE SLIDE**

 After installing the webpack, you need to configure it. To do this, a webpack.config.js, which exports the object. This object contains webpack settings.

The main task of  webpack is to analyze modules, transform them optionally and intelligently combine them into one or more bundles, so webpack needs to know three things:

**POINTING TO THE SLIDE**

1. Application entry point
2. Conversion, you need to perform
3. Place, where you want to place shaped bundle

**CHANGE SLIDE**

***Point of entry***

No matter how many modules an application contains, there is always a single entry point. This module includes rest. Typically, this file is index.js. It might look like this:

**POINTING TO THE SLIDE**

If we tell webpack path to this file, it uses it to create application's dependency graph.

**CHANGE SLIDE**

To do this, you need to add the entry property to the webpack settings with path value to main file:

**POINTING TO THE SLIDE**

***loaders***

After adding an entry point, you need to inform webpack about the changes, you need to complete before generating the bundle. For this, loaders are used.

**CHANGE SLIDE**  
By default, when generating a dependency graph based on import / require () statements, the webpack can only process JavaScript and JSON files.

Almost always, application uses styles,  SVG , images. Loaders are used to transform them.

**CHANGE SLIDE**

Since we want to load SVG, using this command

**POINTING TO THE SLIDE**

Next, add it to the webpack settings. All loaders are included in array of module objects.rules:

**POINTING TO THE SLIDE**

Information on the loader consists of two parts.  First is type of files being processed

**POINTING TO THE SLIDE**

Second - the loader, used for treatment of this type of file.

**POINTING TO THE SLIDE**

Now we can import  SVG-files.

**CHANGE SLIDE**

For our styles to work correctly, we need to use two loaders.

To import css-files use css-loader.

**POINTING TO THE SLIDE**

This is the command for installation,

**POINTING TO THE SLIDE**

but here is how to register loader in config

**POINTING TO THE SLIDE**

**CHANGE SLIDE**

In order to put styles in < style > tag and for them to be applied to DOM elements, you need a style-loader.

Here is the command to install it

**POINTING TO THE SLIDE**

And here we add it to the config

**POINTING TO THE SLIDE**

Note that since there are two loaders used to process CSS files, the value of use property is an array. Also pay attention to order of loaders, first style-loader, then css-loader. It is important. The webpack will apply them in reverse order. First it uses css-loader to import css, then style-loader to inject styles into DOM.

**CHANGE SLIDE**

Also, our application will need images formats jpg, png, gif. For this, we can install file-loader using this command.

**POINTING TO THE SLIDE**

And register it in config

**POINTING TO THE SLIDE**

**CHANGE SLIDE**

Loaders can be used not only to import files, but also to convert them. For example Babel is used to transform modern JavaScript to ES5. And as always, this is the command for installation,

**POINTING TO THE SLIDE**

and here we write loader

**POINTING TO THE SLIDE**

There are loaders for almost any type of file.

**CHANGE SLIDE**

**Exit point**

Now webpack knows about the entry point and loaders. The next step is to specify directory for the bundle. To do this, add output property to webpack settings. Here we indicate the folder, where you want to place the finished bundle,

**POINTING TO THE SLIDE**

and here name of the bundle itself.

**POINTING TO THE SLIDE**

The whole process looks something like this:

**POINTING TO THE SLIDE**

1. Webpack receives the entry point, which is in  ./app/index.js
2. It parses import / require statements and creates a dependency graph
3. The webpack starts building bundle by

converting code using the appropriate loaders

1. It collects bundle and puts it in dist/index\_bundle.js

***Launching a webpack***

At the moment we know how webpack works and how to configure it, it remains to run it.

**CHANGE SLIDE**  
  
We have a package.json, where we can create a script to run webpack.

**POINTING TO THE SLIDE**

Now when you run “npm run build” in the terminal

**POINTING TO THE SLIDE**

will be launched webpack, which will create a bundle  and put it in dist.

**CHANGE SLIDE**

That is all for me. Now you know how to install and configure Webpack for your project. Thank you all for your attention, link to presentation under video. Bye everyone!