Npm v6 .hooks deprecation and npm v7 workspaces

Let's navigate into the npm history to review what's relevant change or features we found between npm v6 and npm v7. We will focusing our attention in the npm v6 .hooks that become deprecated in npm v7 and see and simple example for npm workspaces.

From hooks scripts to bin scripts

To provide a better context, let's see a summary of the .hooks scripts:

Hook scripts: npm v6 provide this executable scripts located by default in `node_modules/.hooks/{eventname}` this hooks works as the same way of the package.json scripts. if you need more information read the hook scripts documentation.

The hooks scripts were removed as part of the npm v7 updates. That's because is a horrible way to configure and work with. So, the correct way to work with additional scripts is the bin scripts.

Bin scripts: These scripts are executable files that will be installed into the npm PATH. So, with this way npm make this very easy. See more information regarding in the npm bin documentation. To finalize with this first topic lets do a quick practice with the npm bin scripts:

Create an empty directory(bin-scripts-example) using the npm initializer:

```
npm init -y
```

now Create new file into bin/hello.js

```
#!/usr/bin/env node


// eslint-disable-next-line no-console
console.log('Hello World');
```

then update your `package.json` file with the bin field:

```
 "bin": {
   "hello-script": "bin/hello.js"
 },
```

Now in any place that you install your package you will have available the script:

```
$ npm run hello-script
Hello World
```

Npm workspaces:

Npm v7 introduce the workspaces feature. That Help to share information between internal packages in a monorepo. Let's do a quick practice following the steps bellow:

Configure the basic project:

Create an empty directory(nice-workspace-example) using the npm initializer:

```
npm init -y
```

now Create the packages

```
npm init -y --scope @nice-workspace-example -w packages/back
npm init -y --scope @nice-workspace-example -w packages/front
npm init -y --scope @nice-workspace-example -w packages/controllers
```

you will see the files extructure as following:

```
- [**packages**](packages)
    - [**controllers**](packages/controllers)
    - [**back**](packages/back)
    - [**front**](packages/front)
- [nice-workspace-example](readme.md)
```

Share the dependencies needed between the packages

```
npm install @nice-workspace-example/back -w @nice-workspace-example/front
npm install @nice-workspace-example/back -w @nice-workspace-example/controllers
```

Analize the workspace workflows between the packages:

lets add some simple code to the share package to test it:

update the `packages/back/index.js` as following:

```
console.log("Hello from back!");
```

update the `packages/front/index.js` as well:

```
require("@nice-workspace-example/back");
console.log("Hello from front!");
```

finally add this code snipped to the `packages/controllers/index.js` file:

```
require("@nice-workspace-example/back");
console.log("Hello from controllers!");
```

and now run your file with node:

```
node packages/controllers/index.js
```

you will see the following output in your console:

```
Hello from back!
Hello from controllers!
```