

WEB10 - MANAGING DEPENDENCIES USING REQUIREJS



Fabio Franzini

Microsoft SharePoint MVP

fabio@fabiofranzini.com - @franzinifabio

<http://www.fabiofranzini.com/>

Grazie a



Sponsor



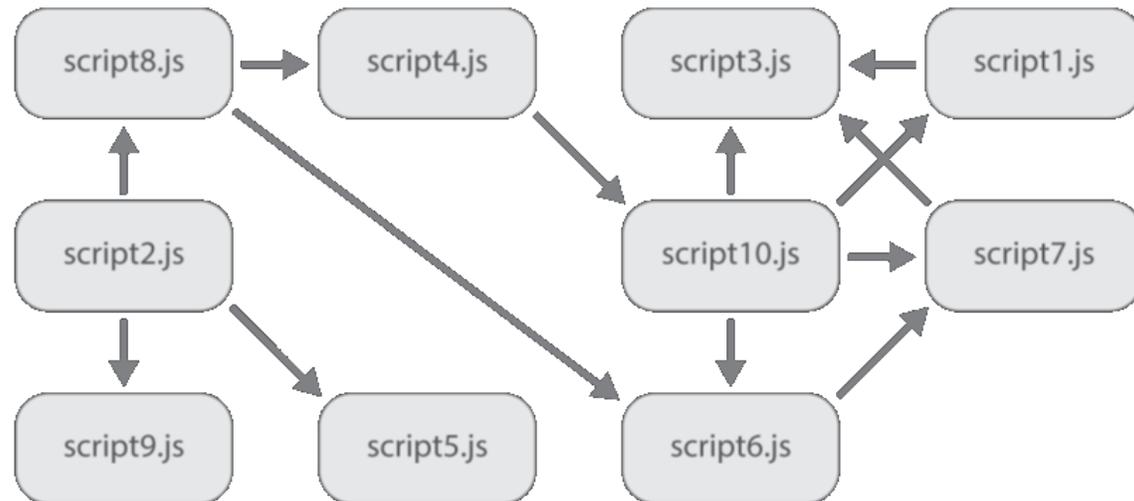
Agenda

- Introduction
- RequireJs
- Defining and Using a RequireJs module
- Optimization

Introduction

Dependency Management is:

Loading the right code in the right order at the right time



First Problem: Ordering

What are the dependencies here?

```
<script src="script3.js"></script>  
<script src="script1.js"></script>  
<script src="script7.js"></script>  
<script src="script6.js"></script>  
<script src="script4.js"></script>  
<script src="script5.js"></script>  
<script src="script9.js"></script>  
<script src="script8.js"></script>  
<script src="script2.js"></script>
```

Second Problems: Global Variables

- No control of access
- No garbage collection
- Naming collision
- Etc..

Partial Solution: Module

Module is a structure used to encapsulate methods and attributes to avoid polluting the global namespace.

```
var revealingModule = (function () {  
  
    var privateVar = "Ben Cherry", publicVar = "Hey there!";  
  
    function publicSetName(strName) { privateVar = strName; }  
    function publicGetName() { privateFunction(); }  
    function privateFunction() { console.log("Name:" + privateVar); }  
  
    return {  
        setName: publicSetName,  
        greeting: publicVar,  
        getName: publicGetName  
    };  
  
})();  
  
revealingModule.setName("Paul Kinlan");
```

RequireJS

- JavaScript files and modules loader, capable of loading asynchronously the required modules and their dependencies
- Open Source (BSD and MIT licensed)



Defining and Using a RequireJs module 1/2

define(id?, dependencies?, callback);

- The id argument is an optional string representing the name of the module (the name of the file becomes the id of the module and will be added into the module when running the r.js optimization tool)
- The dependencies argument is an optional array of other modules, which this module is dependent on

require(dependencies, callback?);

- The dependencies argument is an array of other modules, which this module is dependent on

Defining and Using a RequireJs module 2/2

```
// /js/car.js
define({
  model: "A4 2.0 TDI",
  vendor: "Audi"
});

// /js/cars.js
define(['car'], function (car) {
  return {
    showCar: function () {
      alert(car.vendor + " - " + car.model);
    }
  };
});

// /js/main.js
require(['cars'], function (cars) {
  cars.showCar();
});
```

Using RequireJs in page

```
<!DOCTYPE html>
<html>
<head>
  <title>Example</title>
  <script data-main="js/main" src="js/libs/require.js"></script>
</head>
<body>
</body>
</html>
```

define() vs require()

define() complete three steps:

1. Loads the specified dependencies
2. Calls the callback function
3. Registers the return value from the callback function

require() only performs #1 and #2

demo

define and require demo



Optimization using r.js

NodeJs building tool for RequireJs

```
node ..\scripts\rjs -o ..\_build\build.js
```

```
{
  baseUrl: "../",
  name: "js/main",
  include: ["Scripts/require"],
  exclude: [],
  optimize: "uglify2",
  out: "../js/main-built.js",
  insertRequire: ["js/main"]
}
```

demo

Using r.js



Recap

- Using RequireJs for modularize your javascript code
 - <http://requirejs.org/>
- Using r.js for optimize downloading speed of your RequireJs code
 - <http://requirejs.org/docs/optimization.html>

Q&A

Tutto il materiale di questa sessione su

<http://www.communitydays.it/>

Lascia il feedback su questa sessione,
potrai essere estratto per i nostri premi!

Seguici su

Twitter @CommunityDaysIT

Facebook <http://facebook.com/cdaysit>

#CDays14

