Core - Task # 48383

Epic # 55575 (Accepted): Streamline JavaScript Code in the TYPO3 Backend

Status:	Accepted	**Priority:**	Should have
Author:	Oliver Hader	**Category:**	Default
Created:	2013-05-19	**Assigned To:**	
Updated:	2015-07-28	**Due date:**	
TYPO3 Version:	6.1		
PHP Version:	5.3		
Complexity:	hard		
Sprint Focus:			
Subject:	RequireJS does not consider correct namespace prefix		

Description

My extension "mapping" uses these namespace prefixes in PHP "OliverHader\Mapping" (vendor, package).
However, the requireJS implementation in the PageRenderer ignores that and adds "TYPO3\CMS" as prefix.

The behaviour shall be the same as for PHP class files - the "\TYPO3\CMS" prefix is a special case for the Core only.

Related issues:

related to Core - Bug # 55264: requireJS for frontend in PageRenderer and Typ...	**Under Review**	**2014-01-23**

History

#1 - 2014-02-02 11:20 - Daniel Siepmann

- Parent task set to #55575

- Is Regression set to No

`

#2 - 2014-03-10 16:31 - Daniel Siepmann

The only thing I can implement at the moment is to remove the TYPO3 vender for external extensions. But I don't have any clue how to add the vendor for JS Files as it just exists inside PHP Files.

IF someone has an idea code it, or answer, so I can get into it.

#3 - 2015-07-19 19:50 - Benjamin Mack

- Assigned To set to Benjamin Mack

- Target version set to 7.4 (Backend)

- Sprint Focus set to Stabilization Sprint

#4 - 2015-07-23 11:51 - Mathias Brodala

You can manually add your prefix with the PageRenderer hook render-preProcess like this:

```
$pageRenderer->addRequireJsConfiguration([
  'paths' => [
    'Vendor/Package' => PathUtility::getRelativePath(PATH_typo3,
GeneralUtility::getFileAbsFileName('EXT:package/Resources/Public/JavaScript/')),
    ],
]);
```

#5 - 2015-07-27 20:20 - Markus Klein

- Status changed from New to Rejected

- Assigned To deleted (Benjamin Mack)

- Target version deleted (7.4 (Backend))

- Complexity changed from medium to hard

- Sprint Focus deleted (Stabilization Sprint)

This is still private API of Core. To add you own modules you need to do it like this:

```
$fullJsPath = 'EXT:' . $this->extKey . '/Resources/Public/JavaScript/';
$fullJsPath = GeneralUtility::getFileAbsFileName($fullJsPath);
$fullJsPath = PathUtility::getRelativePath(PATH_site, $fullJsPath);
$fullJsPath = $tsfe->absRefPrefix . rtrim($fullJsPath, '/');

$pageRenderer->addRequireJsConfiguration([
    'paths' => [
        'Reelworx/MyExt' => $fullJsPath
    ]
]);

$pageRenderer->loadRequireJsModule('Reelworx/MyExt/somemodule');
```

Making this API really versatile is a load of work and for sure no bug!

#6 - 2015-07-28 12:41 - Benjamin Mack

- Tracker changed from Bug to Task

- Status changed from Rejected to Accepted

- Target version set to 7.5

ok. but we should "do it right" for 7.5 then.