TYPO3.Flow - Feature #33371

HTTP 1.1 Support

2012-01-22 17:15 - Robert Lemke

Status:	Resolved		**Start date:**	2012-05-17
Priority:	Must have		**Due date:**	
Assignee:	Robert Lemke		**% Done:**	100%
Category:	Http		**Estimated time:**	0.00 hour
Target version:	1.1 RC1			
PHP Version:			**Complexity:**	nightmare
Has patch:	No			

Description
Implement proper HTTP 1.1 support. This issue is a stub for the corresponding FLOW3 1.1 key feature (until I come up with some better description)

Subtasks:	
Task # 37259: Improve spec coverage of Request / Response classes	**Resolved**

Related issues:		
Related to TYPO3.Flow - Feature #35243: HTTP 1.1 Support – Foundations	**Resolved**	**2012-03-26**
Related to TYPO3.Flow - Feature #37212: Edge Side Includes (ESI)	**Accepted**	**2012-05-16**

Associated revisions

Revision 331a78e8 - 2012-04-11 14:09 - Robert Lemke

[!!!][TASK] Sub package name case and location adjustments

This change set contains (mostly) renames and relocations of FLOW3 sub
namespaces. These changes are necessary to meet the FLOW3 coding
guidelines and naming conventions and to support the new CLI / HTTP
and MVC structure.

\TYPO3\FLOW3\AOP -> \TYPO3\FLOW3\Aop
\TYPO3\FLOW3\MVC -> \TYPO3\FLOW3\Mvc
\TYPO3\FLOW3\MVC\CLI -> \TYPO3\FLOW3\Cli
\TYPO3\FLOW3\MVC\Web\Routing -> \TYPO3\FLOW3\Mvc\Routing
\TYPO3\FLOW3\MVC\Web\Request -> \TYPO3\FLOW3\Mvc\ActionRequest
\TYPO3\FLOW3\MVC\Web\Response -> \TYPO3\FLOW3\Http\Response
\TYPO3\FLOW3\MVC\Web\SubRequest -> \TYPO3\FLOW3\Mvc\ActionRequest
TYPO3\FLOW3\MVC\Web\SubResponse -> TYPO3\FLOW3\Http\Response
\TYPO3\FLOW3\MVC\Controller\CommandController ->
\TYPO3\FLOW3\Cli\CommandController
\TYPO3\FLOW3\Property\DataType\Uri -> \TYPO3\FLOW3\Http\Uri

\TYPO3\FLOW3\MVC\Web\RequestBuilder doesn't exist anymore. If
you need to create requests, do "new ActionRequest($parentRequest)"
\TYPO3\FLOW3\MVC\Web\SubRequestBuilder doesn't exist anymore.
If you need to create sub requests, do "new ActionRequest($parentRequest)""

Note that this change set contains most but not every single adjustment
to the new locations as it is part of a bigger change for the respective
feature.

Change-Id: Idd28eb709d9543aff97b333dcbf6fa4d544b5746
Resolves: #33705
Related: #33371
Releases: 1.1

Revision eb813f96 - 2012-04-11 14:10 - Sebastian Kurfuerst

[TASK] Add missing getParentRequest() method in HTTP Response

Furthermore, some documentation cleanup.

Related: #35243

Related: #33371
Releases: 1.1
Change-Id: I310dad852b8ac0053bd5c2a6c4b03e8f32c1c6d3

Revision bbc5e334 - 2012-04-11 14:10 - Robert Lemke

[FEATURE] Provide method getContent() for HTTP request

This implements a method for retrieving the HTTP request body.

Change-Id: I6c11fc7a72ee3da76b73c87a48ab0c2dc8fb53e6
Related: #33371
Releases: 1.1

Revision 0e05ccc5 - 2012-04-11 14:10 - Robert Lemke

[!!!][FEATURE] HTTP 1.1 Support – Foundations

This commit contains the foundations for a better HTTP support in
FLOW3. It extracts request and response handling into a dedicated
sub package "Http" and clearly distinguishes between HTTP requests
and "action requests".

While this change set does contain a few new features which are
related to HTTP support, it is by far not complete and rather serves
as a foundation for further features which are following.

Change-Id: I557e6c45cdf20970f9410374aa03ee115e7789cd
Resolves: #35243
Related: #33371
Releases: 1.1

Revision 238efb06 - 2012-05-23 13:57 - Robert Lemke

[TASK] Temporarily remove nested response code

This temporarily removes the code in HTTP Response which
tackles support of nested responses. This feature needs
to be implemented only when most of the other features,
including Cache Control, are in place.

Change-Id: Idd85fcb53be412d3df5c603da37d4015f71c36c0
Releases: 1.1, 1.2
Related: #33371

Revision 26f83dd3 - 2012-05-23 16:21 - Robert Lemke

[FEATURE] Support for HTTP PUT/DELETE arguments

This adds transparent argument support for arguments
in PUT and DELETE requests which are passed through
the request body. Depending on the content type of
the request, the message body is parsed and then
mapped into arguments for further use in controllers.

The following content content types are currently
supported:

- application/x-www-form-urlencoded
- application/json
- xml

Change-Id: Ifa6712e38d68c94cd9ec79a499d25ee577fc2d57
Resolves: #36913
Related: #37402
Related: #33371
Releases: 1.1, 1.2

Revision 5151598f - 2012-05-23 16:25 - Robert Lemke

[TASK] Temporarily remove nested response code

This temporarily removes the code in HTTP Response which
tackles support of nested responses. This feature needs
to be implemented only when most of the other features,

including Cache Control, are in place.

Change-Id: Icfbf4186c08997c4ce69838bbe4145a8af1e4b0b
Releases: 1.1, 1.2
Related: #33371

Revision 8ce2f8da - 2012-05-23 16:37 - Robert Lemke

[FEATURE] Support for HTTP PUT/DELETE arguments

This adds transparent argument support for arguments
in PUT and DELETE requests which are passed through
the request body. Depending on the content type of
the request, the message body is parsed and then
mapped into arguments for further use in controllers.

The following content content types are currently
supported:

- application/x-www-form-urlencoded
- application/json
- xml

Change-Id: Id938025a7359b6ec3b29cec5fd7259db97703a16
Resolves: #36913
Related: #37402
Related: #33371
Releases: 1.1, 1.2

Revision 149a012e - 2012-05-25 15:48 - Robert Lemke

[FEATURE] Make Message / Response setters chainable

This makes the setter methods of the Message and the
Response class chainable.

Change-Id: I826ad686a6d0649c42767a950de2e7e169e312b6
Related: #33371
Releases: 1.1, 1.2

Revision b18a9a51 - 2012-05-25 15:48 - Robert Lemke

[FEATURE] More convenience methods in Response

This adds a few more convenience methods for setting
headers and cache control directives in the Response
object:

- setDate() / getDate()
- setLastmodified() / getLastModified()
- setMaximumAge() / getMaximumAge()
- setSharedMaximumAge() / getSharedMaximumAge()

Change-Id: I8c253574bfb6be5ca42ce17d16bcf49d4d268933
Related: #33371
Releases: 1.1, 1.2

Revision fb8b0647 - 2012-05-25 15:48 - Robert Lemke

[FEATURE] Support for HTTP Cache-Control headers

This adds support for the Cache-Control header.
The "Headers" object now handles the Cache-Control
header in a special way by keeping track of the
various cache directives separately. Those can
be set through the method setCacheControlDirective()
and get('Cache-Control') returns a rendered version
based on the previously set directives.

Alternatively, a whole Cache-Control header can also
be set through set() / setHeader(). The given field
value will then be parsed and the individual cache
directives are stored internally.

This patch also corrects the behavior of set() and

get() to convert date / time values always to GMT.
Also added a related note in the comments of the
respective API methods.

This patch also adds the new API methods setPrivate()
and setPublic() to the Response class.

Change-Id: I7d7bae82e40ff24b176be6fae2b4a69c040c992b
Related: #33371
Releases: 1.1, 1.2

Revision 60c22ce6 - 2012-05-25 17:34 - Robert Lemke

[FEATURE] Response setExpires() / getExpires()

This adds two new convenience methods to the Response
API which allow for setting and retrieving the Expires
HTTP header.

Change-Id: Iba33b8f34c702ddecb9e6ed51bdc071270c27ef9
Related: #33371
Releases: 1.1, 1.2

Revision 27f16226 - 2012-05-25 22:09 - Robert Lemke

[FEATURE] Support for HTTP Cache-Control headers

This adds support for the Cache-Control header.
The "Headers" object now handles the Cache-Control
header in a special way by keeping track of the
various cache directives separately. Those can
be set through the method setCacheControlDirective()
and get('Cache-Control') returns a rendered version
based on the previously set directives.

Alternatively, a whole Cache-Control header can also
be set through set() / setHeader(). The given field
value will then be parsed and the individual cache
directives are stored internally.

This patch also corrects the behavior of set() and
get() to convert date / time values always to GMT.
Also added a related note in the comments of the
respective API methods.

This patch also adds the new API methods setPrivate()
and setPublic() to the Response class.

Change-Id: I7d7bae82e40ff24b176be6fae2b4a69c040c992b
Related: #33371
Releases: 1.1, 1.2

Revision c4034394 - 2012-05-25 22:09 - Robert Lemke

[FEATURE] More convenience methods in Response

This adds a few more convenience methods for setting
headers and cache control directives in the Response
object:

- setDate() / getDate()
- setLastmodified() / getLastModified()
- setMaximumAge() / getMaximumAge()
- setSharedMaximumAge() / getSharedMaximumAge()

Change-Id: I8c253574bfb6be5ca42ce17d16bcf49d4d268933
Related: #33371
Releases: 1.1, 1.2

Revision 11f8eb8e - 2012-05-25 22:09 - Robert Lemke

[FEATURE] Make Message / Response setters chainable

This makes the setter methods of the Message and the
Response class chainable.

Change-Id: I826ad686a6d0649c42767a950de2e7e169e312b6
Related: #33371
Releases: 1.1, 1.2

Revision 3e5836fa - 2012-05-29 10:18 - Robert Lemke

[FEATURE] Response setExpires() / getExpires()

This adds two new convenience methods to the Response
API which allow for setting and retrieving the Expires
HTTP header.

Change-Id: Iba33b8f34c702ddecb9e6ed51bdc071270c27ef9
Related: #33371
Releases: 1.1, 1.2

Revision 63283d92 - 2012-05-29 10:25 - Robert Lemke

[FEATURE] Make HTTP responses standards compliant

This adds a new API method to the HTTP response which
is called by a request handler to assure conformity
with RFC 2616 and other related specifications.

While there are a lot of rules to consider, this patch
only introduces a first set of important checks which
are all explained by a corresponding test.

Change-Id: I8b7a9bf53061cfc371c22bdec018c1918c63a4fe
Related: #33371
Releases: 1.1, 1.2

Revision 462fec11 - 2012-05-29 10:31 - Robert Lemke

[FEATURE] Make HTTP responses standards compliant

This adds a new API method to the HTTP response which
is called by a request handler to assure conformity
with RFC 2616 and other related specifications.

While there are a lot of rules to consider, this patch
only introduces a first set of important checks which
are all explained by a corresponding test.

Change-Id: I8b7a9bf53061cfc371c22bdec018c1918c63a4fe
Related: #33371
Releases: 1.1, 1.2

Revision ef3e8fbb - 2012-05-29 22:41 - Robert Lemke

[!!!][TASK] Switch from MIME to Media Types

The term MIME type is outdated, at least if used in a web context. The correct
term is "Internet Media Type". Furthermore, our list of MIME types (or media
types) was not up to date.

This patch introduces a new utility class "MediaTypes" which replaces
"FileTypes". Along with the new class comes a script which allows the core team
to conveniently update the list of Media Types and filename extensions.

This is a breaking change as the FileTypes class is deprecated with it. The old
methods are still available but should not be used anymore. A code migration to
use the new ones instead is shipped with the change.

Change-Id: I8f0997f79f09c828dbce5c7a34b4487a522aab0c
Related: #37402
Related: #33371
Releases: 1.1, 1.2

Revision fc4c4ab5 - 2012-05-30 14:24 - Robert Lemke

[!!!][TASK] Switch from MIME to Media Types

The term MIME type is outdated, at least if used in a web context. The correct
term is "Internet Media Type". Furthermore, our list of MIME types (or media

types) was not up to date.

This patch introduces a new utility class "MediaTypes" which replaces "FileTypes". Along with the new class comes a script which allows the core team to conveniently update the list of Media Types and filename extensions.

This is a breaking change as the FileTypes class is deprecated with it. The old methods are still available but should not be used anymore. A code migration to use the new ones instead is shipped with the change.

Change-Id: l8f0997f79f09c828dbce5c7a34b4487a522aab0c
Related: #37402
Related: #33371
Releases: 1.1, 1.2

History

#1 - 2012-02-19 00:10 - Jacob Floyd

Robert Lemke - I've seen/heard your comments in the scrum protocols/meeetings about this HTTP work you're doing.

I've been browsing the ietf specs to understand what implementing HTTP might entail, and I must admit: I am ecstatic. Whatever you're doing will be awesome.

So, in my browsing, I've found 2616, and 6265. I also looked at 3229 which looks pretty sweet for sending page changes via AJAX.

And then there are related specs about URIs and URLs: 3986 which updates 1738 and obsoletes 1808, 2396, and 2732.

Which specs are you handling in FLOW3? Are you only working on 2616 and 6265? Or are you also doing 3229, and 3986?

#2 - 2012-05-25 09:40 - Karsten Dambekalns

- Target version changed from 1.1 to 1.1 RC1

#3 - 2012-06-20 21:16 - Karsten Dambekalns

- Status changed from Accepted to Resolved