

TYPO3.Flow - Task #3398

Are methods in CLI\Request still used?

2009-05-20 19:16 - Robert Lemke

Status:	Resolved	Start date:	2009-05-20
Priority:	Should have	Due date:	
Assignee:	Karsten Dambekalns	% Done:	100%
Category:	MVC	Estimated time:	0.00 hour
Target version:	1.0 alpha 1	Has patch:	
Sprint:		Complexity:	
PHP Version:			

Description

In the CLI Request I currently find these functions:

```
/**
 * Sets the arguments given to this request.
 *
 * @param array $arguments
 * @return void
 * @author Karsten Dambekalns <karsten@typo3.org>
 */
public function setCLIArguments(array $arguments) {

/**
 * Returns the arguments given to the request.
 *
 * @return array
 * @author Karsten Dambekalns <karsten@typo3.org>
 */
public function getCLIArguments() {
```

Are they still in use? Why do we need special arguments for CLI?

Associated revisions

Revision d45434c4 - 2009-05-25 15:04 - Karsten Dambekalns

FLOW3:

- CLI request builder now has a more friendly error message for calls with 2 command parts
- renamed CLIArguments to CommandLineArguments in CLI\Request, refs #3398
- removed some superfluous code in CLI\RequestBuilder

Revision 2366 - 2009-05-25 15:04 - Karsten Dambekalns

FLOW3:

- CLI request builder now has a more friendly error message for calls with 2 command parts
- renamed CLIArguments to CommandLineArguments in CLI\Request, refs #3398
- removed some superfluous code in CLI\RequestBuilder

History

#1 - 2009-05-20 21:17 - Karsten Dambekalns

Hehe, from my OmniFocus:

- rename to avoid confusion with arguments in request object
- make it mandatory to register use of CLI arguments to get them

#2 - 2009-05-23 23:00 - Karsten Dambekalns

As for the "Why do we need special arguments for CLI?" see this explanation from my mail "Re: [TYPO3-team-core-v5] CLI support for controllers in sub-packages" from 2008-09-25 11:55:34 CEST

```
> you were too fast for me at least.. I didn't really get the difference
> between options and arguments. Do we need to separate them at all?
```

Yes, we do.

Consider some command that accepts a bunch of files as input. You do probably not want to add one option per file. And adding more than one value to one option is also bad style IMHO.

So, let's use a lame example: Assume we have `ls` (dir for you ;) as part of some FLOW3 shell-emulator. How do you write this in a FLOW3 call:

```
ls -G *.jpg
```

(Keep in mind the wildcard is expanded by the shell, so FLOW3 receives a list of arguments, not the literal `"*.jpg"`!)

I'd expect this:

```
php index.php shell file ls -G -- *.jpg
```

(I admit the `--` is ugly here, but we don't know if `-G` accepts a value)

The alternative would be to add some option, say `"subject"`:

```
php index.php shell file ls -G --subject *.jpg
```

Now you'd need to `explode(' ', $subject)`. Bah. Or, even worse:

```
php index.php shell file ls -G --subject one.jpg --subject2 two.jpg ...
```

Argh, right?

Anyway, I hope at least I could explain the difference between options and arguments.[1]

Karsten

[1] Another way: options is what you specify like `"\b"` in DOS, arguments is what you simply give to your command.

[2] Yet another way: options is `"how"`, arguments is `"with what"` - more or less.

#3 - 2009-05-25 15:30 - Karsten Dambekalns

- *Status changed from Needs Feedback to Resolved*

- *% Done changed from 0 to 100*

The methods stay (see above for reasons), explicit registration is nonsense (you need to explicitly fetch them anyway) and some renaming has been done in r2366.