

## TYPO3.Flow - Bug # 35720

<b>Status:</b>	New	<b>Priority:</b>	Must have
<b>Author:</b>	Johannes K	<b>Category:</b>	Security
<b>Created:</b>	2012-04-05	<b>Assigned To:</b>	
<b>Updated:</b>	2012-10-25	<b>Due date:</b>	
<b>PHP Version:</b>			
<b>Has patch:</b>	No		
<b>Complexity:</b>			
<b>Affected Flow version:</b> Git master			
<b>Subject:</b>	Access denied Exception for widget links to actions with a policy		
<b>Description</b>			
<p>In TYPO3\FLOW3\Security\Aspect\CsrfProtectionAspect::addCsrfTokenToUri() the detection for the target classname fails, if a link is generated via &lt;f:link.widget /&gt;, so the link is missing the __csrfToken and you get an AccessDeniedException:</p> <p>#1216919280: You are not allowed to perform this action. (More information)</p> <p>TYPO3\FLOW3\Security\Exception\AccessDeniedException thrown in file .../Data/Temporary/Development/Cache/Code/FLOW3_Object_Classes/TYPO3_FLOW3_Security_Authorization_Interceptor_Accesseny_Original.php in line 30.</p> <p>Example to reproduce: Use the paginate widget for an action with a policy</p>			
<b>Related issues:</b>			
duplicates TYPO3.Flow - Bug # 27798: CSRF protection not working for forms in...		<b>Accepted</b>	<b>2011-07-01</b>

### History

#### #1 - 2012-04-05 15:58 - Johannes K

The problem is that the CsrfProtectionAspect is woven into UriBuilder::build() (via "method(TYPO3\FLOW3\MVC\Web\Routing\UriBuilder->build())"), but the arguments to determine the target Controller/Action are not yet merged into \$arguments, so the CsrfProtectionAspect has no way to check the action.

Possible solution: Use the following code and weave the CsrfProtectionAspect into UriBuilder::doBuild(), but maybe you guys have a better solution ;)

```
class UriBuilder {
...
/**
 * Builds the URI
 *
 * @param array $arguments optional URI arguments. Will be merged with $this->arguments with precedence to $arguments
 * @return string The URI
 * @api
 */
public function build(array $arguments = array()) {
    $arguments = \TYPO3\FLOW3\Utility\Arrays::arrayMergeRecursiveOverrule($this->arguments, $arguments);
    $this->mergeArgumentsWithRequestArguments($arguments);

    $this->doBuild($arguments);
}
```

```
}  
  
protected function doBuild(array $arguments = array()) {  
    // rest of the build function  
}  
...  
}
```