TYPO3.Flow - Feature # 342

Status:	Resolved	**Priority:**	Must have
Author:	Robert Lemke	**Category:**	MVC
Created:	2008-02-20	**Assigned To:**	Robert Lemke
Updated:	2010-10-20	**Due date:**	
PHP Version:			
Has patch:			
Complexity:			
Subject:	Support for file uploads		

Description

The MVC framework needs to support file / multipart uploads. It should be implemented in a way that uploads are only accepted if they have been announced on beforehand. Could be implemented with a token which is passed along.

Related issues:

related to TYPO3.Flow - Feature # 5660: Implement support for Object Converters	**Resolved**	**2009-12-07**
related to TYPO3.Flow - Feature # 6008: Automatically purge files of removed ...	**Resolved**	**2010-01-13**

Associated revisions

Revision 027a4016 - 2009-12-07 19:09 - Robert Lemke

[~TASK] FLOW3 (AOP): Removed the "Resource" sub package from the blacklisted sub packages because it now contains a class (Resource) which needs to be persistable.
[-FEATURE][!!!] FLOW3 (MVC): For now removed the request hash feature (HMAC) because it mocks a level of security for incoming data which it doesn't provide. The current mechanism effectively puts control over content security into Fluid templates and it doesn't belong there. Although there might be a need for a request hash, the content security must be implemented by other means. Relates to #4960 and relates to #5659.
[+FEATURE] FLOW3 (MVC): Implemented support for file uploads. Uploading files is cooperation between the Web Request Builder, the Property Mapper and the Resource sub package. The solution included in this commit provides handling of incoming files (including nested arguments) and transparent conversion into Resource objects. Resources (files) are only stored once, no matter how often they are uploaded or what original filename they carried. Still missing: view helper, documentation and automatic purging of unused resource files. Addresses #342.
[~API][!!!] FLOW3 (Property): Renamed the property mapper class to "PropertyMapper" (was just "Mapper" before). Relates to #5658
[+FEATURE] FLOW3 (Property): The Property Mapper now supports a mechanism called Object Converters. These convertes enable the mapper to convert strings, arrays or numbers to certain objects, for example a unix time stamp to a DateTime object. Resolves #5660.
[+FEATURE] FLOW3 (Reflection): Implemented the methods "isPropertySettable" and "isPropertyGettable" for the ObjectAccess class.
[~TASK] FLOW3 (Resource): Renamed the StreamWrapper class to StreamWrapperAdapter
[+FEATURE] FLOW3 (Resource): Implemented a ResourceObjectConverter which is capable of converting arrays or strings to Resource objects.
[+FEATURE] FLOW3 (Utility): Implemented a setValueByPath() method for the Array utilities class.
[+FEATURE] FLOW3 (Utility): Added support for the _FILES super global to the Environment class. The array of information about uploaded files can be obtained in a much cleaner way than PHP provides it by the new getUploadedFiles() method.

History

#1 - 2008-07-01 15:36 - Robert Lemke

- Target version deleted (1)

#2 - 2009-05-19 12:34 - Robert Lemke

- Target version set to 283

#3 - 2009-07-13 11:24 - Robert Lemke

- Target version deleted (283)

#4 - 2009-10-27 15:21 - Karsten Dambekalns

- Status changed from New to Accepted

- Assigned To set to Karsten Dambekalns

#5 - 2009-11-23 15:33 - Karsten Dambekalns

What we need (to consider):

- a form.file viewhelper
- a file validator of some sort
  - applicable to traversable, checks each element
  - checks
    - size
    - extension
    - mimetype
  - subclasses as needed (for images, ...)
- a way to "use" files in the code
  - filename (better not so directly)
  - file instance

If a file is to be bound to a model and should be persisted then the file resource is simply referenced in the model. If you want to manipulate an uploaded file before persisting it, you simply declare the file instance as an action argument and do whatever you need with the file.

Files will be stored in the CR (up to that to decide on filesystem vs DB storage) and most probably also cached in the local filesystem for faster access (data mapper must grok that).

#6 - 2009-12-07 18:53 - Robert Lemke

- Assigned To changed from Karsten Dambekalns to Robert Lemke

- Priority changed from Should have to Must have

- Target version set to 1.0 alpha 7

- % Done changed from 0 to 80

#7 - 2010-01-13 17:03 - Robert Lemke

- Status changed from Accepted to Resolved

- % Done changed from 80 to 100