TYPO3 Extbase Cheat Sheet 1

GIT: git clone http://git.typo3.org/TYPO3v4/CoreProjects/MVC/extbase.git
Forge: http://forge.typo3.org/projects/show/typo3v4-mvc
Issue-Tracker: http://forge.typo3.org/projects/typo3v4-mvc/issues

v 3.01 / 03.03.2014
Patrick Lobacher / www.lobacher.de
compatible with TYPO3 CMS >= 6.2

LOBACHER.

Domain Driven Design (terms by wikipedia)

Domain	A sphere of knowledge, influence, or activity.
Model	A system of abstractions that describes selected aspects of a domain
Ubiquitous Language (UL)	A language structured around the domain model & used by all team members to connect all the activities of the team with the software. The words of the UL are used throughout all artefacts.
Entity	An object that is not defined by its attributes, but rather by a thread of continuity and its identity. Derives from class \TYPO3\CMS\Extbase\DomainObject\AbstractEntity
Value Object (VO)	An object that contains attributes but has no conceptual identity. They should be treated as immutable. Derives from class \TYPO3\CMS\Extbase\DomainObject\AbstractValueObject
Aggregate	A collection of objects that are bound together by a root entity, known as an aggregate root.
Aggregate root	The aggregrate root guarantees the consistency of changes being made within the aggregate by forbidding external objects from holding references to its members.
Repository	Methods for retrieving domain objects should delegate to a specialized Repository object such that alternative storage implementations may be easily interchanged. Derives from class \TYPO3\CMS\Extbase\Persistence\Repository

Naming Conventions

UpperCamelCase	Directories, Classes, Files, ExtensionName (not extension_key!)
lowerCamelCase	Actions, Methods, Properties
Namespaces	namespace [Vendor]\[ExtensionName]\Dir1\Dir2; ([Vendor] of TYPO3 is: TYPO3\CMS) Location is: [ExtBaseDir]/[extension_key]/Classes/Dir1/Dir2 [ExtensionName] is [extension_key] without _ and in UpperCamelCase [ExtBaseDir]: for own extensions it is typo3conf/ext/, for system extensions it is typo3/sysext/
FQCN	Full Qualified Class Name e.g. \TYPO3\CMS\Extbase\Utility\ExtensionUtility Path: typo3/sysext/extbase/Classes/Utility Classname: ExtensionUtility (namespace TYPO3\CMS\Extbase\Utility); Class filename: ExtensionUtility.php

MISC

FlashMessages ($this->addFlashMessage() OR $this->controllerContext->getFlashMessageQueue())

addFlashMessage($message) enqueue($message)	Adds a FlashMessage in the queue $message = new \TYPO3\CMS\Core\Messaging\FlashMessage($message, $title = '', $severity = \TYPO3\CMS\Core\Messaging\FlashMessage::OK, $storeInSession = FALSE) Severity: NOTICE, INFO, OK, WARNING, ERROR
dequeue()	Removes last FlashMessage from the queue
getAllMessages()	Gets all FlashMessages
getAllMessagesAndFlush()	Get all FlashMessages and removes them from the session

Debugging

```
\TYPO3\CMS\Extbase\Utility\DebuggerUtility::var_dump(...)
```

Exceptions

```
throw new \RuntimeException($text, 1363300072); // Unix-TS because of uniqueness
```

Dependency Injection (DI)

```
/**
 * @var FQCN (Fully qualified class name)
 * @inject
 */
protected $varName;
```

DI Persistence Manager (persist all objects) - FQCN: \TYPO3\CMS\Extbase\Persistence\Generic\PersistenceManager

```
$this->persistenceManager->persistAll();
```

DI Object Manager (gets objects via DI) - FQCN: \TYPO3\CMS\Extbase\Object\ObjectManagerInterface

```
$this->objectManager->get('[Vendor]\\[ExtensionName]\\[Path]\\[ClassName]')->...
$this->objectManager->getScope('[Vendor]\\[ExtensionName]\\[Path]\\[ClassName]')
```

DI Configuration Manager - FQCN: \TYPO3\CMS\Extbase\Configuration\ConfigurationManagerInterface

```
$this->configurationManager->getConfiguration(\TYPO3\CMS\Extbase\Configuration\ConfigurationManagerInterface::XXX,$extensionName=NULL, $pluginName=NULL) [XXX is CONFIGURATION_TYPE_FRAMEWORK, CONFIGURATION_TYPE_SETTINGS, CONFIGURATION_TYPE_FULL_TYPOSCRIPT] | getContentObject()
```

Translation

```
\TYPO3\CMS\Extbase\Utility\LocalizationUtility::translate($key, $extensionName, $arguments = NULL)
```

Folder structure inside extension dir typo3conf/ext/[extension_key]/

ext_autoload.php	Mapping classname to classfile location
ext_emconf.php	Extension manager configuration
ext_icon.gif	Extension icon
ext_localconf.php	Frontend configuration
ext_tables.php	Backend configuration
ext_tables.sql	SQL statements for the databse structure
composer.json /ExtensionBuilder.json	ExtensionBuilder configuration
Classes/	All PHP classes reside here
Classes/Controller/[DomainObjectName]Controller.php	Controller of model [DomainObjectName] (rec.)
Classes/Domain/Model/[DomainObjectName].php	Specific model class for [DomainObjectName]
Classes/Domain/Repository/[DomainObjectName]Repository.php	Repository of model [DomainObjectName]
Classes/Validation/Validator/[DomainObjectName]Validator.php	Validator of model [DomainObjectName]
Classes/ViewHelpers/[VHName]ViewHelper.php	View helper with name VHName
Configuration/	All configuration (structure is a suggestion)
Configuration/TCA/	Table configuration array (TCA)
Configuration/FlexForms/	Flexforms used for backend forms
Configuration/TypoScript/	TypoScript constants and setup
Documentation/	All documentation reside here
Documentation/Manual/	Extension manual, subfolder [format]/[lang]/
Resources/	All resources reside here
Resources/Private/Backend/Layouts/	Layout files for backend modules
Resources/Private/Backend/Templates/[ControllerName]/	All templates of a specific controller (BE)
Resources/Private/Backend/Templates/[ControllerName]/[action].[format]	Template of [action] from [Controller] (BE)
Resources/Private/Language/locallang.xlf	Main language file - use key w. translate viewhelper
Resources/Private/Layouts/	Layout files for frontend plugins
Resources/Private/Partials/	Partials files for frontend plugins
Resources/Private/Templates/[Controller]/[Action].[format]	Template of [Action] from [Controller] (FE)
Resources/Public/	Additional resources (own dirs if needed, like „Icons", ...)
Tests/	All tests reside here

Flexform (example option maxPosts and switchableControllerActions)

```
<T3DataStructure>
  <sheets>
    <sDEF>
      <ROOT>
        <TCEforms>
          <sheetTitle>LLL:EXT:my_extension/Resources/Private/Language/locallang_db.xml:ff.sheetTitle</sheetTitle>
        </TCEforms>
        <type>array</type>
        <el>

...ELEMENTS...

        </el>
      </ROOT>
    </sDEF>
  </sheets>
</T3DataStructure>
```

```
<settings.maxPosts>
  <TCEforms>
    <label>Max. number of posts</label>
    <config>
      <type>input</type>
      <size>2</size>
      <eval>int</eval>
      <default>10</default>
    </config>
  </TCEforms>
</settings.maxPosts>
```

locallang.xlf and [iso].locallang.xlf (in Resources/Private/Language)

```
<switchableControllerActions>
  <TCEforms>
    <label>Some label</label>
    <config>
      <type>select</type>
      <items type="array">
        <numIndex index="0" type="array">
          <numIndex index="0">Default</numIndex>
          <numIndex index="1">Controller1->action1;
                              Controller2->action2;...</numIndex>
        </numIndex>
      </items>
      <maxitems>1</maxitems>
      <size>1</size>
    </config>
  </TCEforms>
</switchableControllerActions>
```

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<xliff version="1.0">
  <file source-language="en" datatype="plaintext" original="-messages" date="date" product-name="[extensionkey]">
    <header/>
    <body>
      <trans-unit id="key">
        <source>Original Text</source>
        <target state="translated">Translated Text</target>
      </trans-unit>
</body></file></xliff>
```

ExtensionUtility-API

Frontend Plugin

ext_tables.php

```
\TYPO3\CMS\Extbase\Utility\ExtensionUtility::registerPlugin(
  '[Vendor].' . $_EXTKEY,                                  // vendor + ext key
  'PluginName',                                            // plugin name
  'Plugin title',                                          // plugin title
  \TYPO3\CMS\Core\Utility\ExtensionManagementUtility::extPath($extensionKey).
        'Resources/Public/Icons/someIcon.gif'              // icon path (optional)
);
```

ext_localconf.php

```
\TYPO3\CMS\Extbase\Utility\ExtensionUtility::configurePlugin(
  '[Vendor].' . $_EXTKEY,                                  // vendor + ext key
  'PluginName',                                            // plugin name
  array(      'Controller1' => 'action1, action2, ...',    // controller action
              'Controller2' => 'action3, action4, ...',    // combinations
  ...),
  array(      'Controller1' => 'action1, action2, ...',    // controller action
              'Controller2' => 'action3, action4, ...',    // combinations uncached
  ...),                                                    // must be in 1st arr too
  \TYPO3\CMS\Extbase\Utility\ExtensionUtility::TYPE_PLUGIN // or TYPE_CONTENT_ELEMENT
);
```

Backend Module

ext_tables.php

```
Tx_Extbase_Utility_Extension::registerModule(
  '[Vendor].' . $_EXTKEY,                                  // vendor + ext key
  'web',                                                   // main module
  'tx_extkey_m1',                                          // sub module
  $position,                                               // see below
  array(      'Controller1' => 'action1, action2, ...',    // controller action
              'Controller2' => 'action3, action4, ...',    // combinations
  ...),
  array(                                                   // configuration array
              'acces'  => 'user,group',
              'icon'   => 'EXT:extkey/ext_icon.gif',
              'labels' => 'LLL:EXT:extkey/Resources/Private/Language/...'
  )
);
```

$position has this syntax: [cmd]:[submodule-key]. cmd can be „after", „before" or „top" (or blank - default). If „after"/"before" then submod will be inserted after/before the existing submod with [submodule-key] if found. If not, the bottom of list. If „top" the module is inserted in the top of the submodule list.

General functions

ext_tables.php

```
// Static TypoScript
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::addStaticFile($_EXTKEY, 'Configuration/TypoScript', 'TS-Template Label');
// Include flexforms
$pluginSignature = '[extkey]_[pluginName]';
$TCA['tt_content']['types']['list']['subtypes_addlist'][$pluginSignature] = 'pi_flexform';
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::addPiFlexFormValue($pluginSignature,'FILE:EXT:[extkey]/Configuration/FlexForms/flexform_bloglisting.xml');
// Allow data entries on standard pages (paramter is table name like tx_simpleblog...)
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::allowTableOnStandardPages('...');
```

PHPDoc annotations (always use FQCN - „use" statement is not enough!)

@api	Declares that the following class/method is part of the official API
@cascade remove	Delete child(s) if parent is removed (use at property in domain model)
@deprecated	Declares that the following class/method should not be used anymore
@dontverifyrequesthash	Disable request hash checking (just used with old property mapper)
@ignorevalidation $var	Action: No validation for $var (use @dontvalidate if old property mapper active)
@inject	Executes the dependency injection (DI) of th class named in @var
@lazy	Lazy loading in domain model (load child objects only when needed)
@param [Type] $var	Action: Parameter. $var validates to [Type]
@return [Type]	Return value is of type [Type]
@validate [$var] [Validator]	Model & Action: Validation for $var. In model without $var. It's possible to use shorthand notation: [ExtensionName]:[ValidatorName]
@var [Type]	Model: Type of var in Domain Model - either simple type, class or ObjectStorage: \TYPO3\CMS\Extbase\Persistence\ObjectStorage <\[Vendor]\[ExtensionName]\Domain\Model\[Model]> delivers methods: count(), attach(), attachAll(), detach(), detachAll(), contains(), ...

TYPO3 Extbase Cheat Sheet 2

GIT: git clone http://git.typo3.org/TYPO3v4/CoreProjects/MVC/extbase.git
Forge: http://forge.typo3.org/projects/show/typo3v4-mvc
Issue-Tracker: http://forge.typo3.org/projects/typo3v4-mvc/issues

v 3.01 / 03.03.2014
Patrick Lobacher / www.lobacher.de
compatible with TYPO3 CMS >= 6.2
LOBACHER.

TypoScript

CONSTANTS: plugin.tx_[lowercasedextensionname] & module.tx_[lowercasedextensionname]			

view	# cat=plugin.tx_myextension/file; type=string; label=Path to template root (FE)
	templateRootPath = EXT:my_extension/Resources/Private/Templates/		

persistence	# cat=plugin.tx_myextension//a; type=int+; label=Default storage PID storagePid =		
SETUP: plugin.tx_[lowercasedextensionname] & module.tx_[lowercasedextensionname] & config.tx_extbase			
settings			Access in controller: $this->settings Access in Fluid: {settings}
features	rewrittenPropertyMapper skipDefaultArguments ignoreAllEnableFieldsInBe		Turn on new property mapper (TRUE) Skip default arguments (FALSE) Ignore enable fields in BE (FALSE)
persistence	enableAutomaticCacheClearing updateReferenceIndex storagePid		Clear Cache at write operations (FALSE) Update refrence index (FALSE) PID list where records are read from
persistence. classes. [fullClassName].	newRecordStoragePid mapping.tableName mapping.recordType		PID where new records will be stored Which table is mapped Record type (needs TCA field 'type' => 'record_type'),
	mapping.columns.	field_name. mapOnProperty = propertyName	Which column in table is mapped field_name = field in table propertyName = property in model
	subclasses.		put one entry for every subclass in superclass definition [Identifier] = [fullClassName]
view	templateRootPath partialRootPath layoutRootPath		Template path (templateRootPaths.ARRAY = fallbackpath) Partial path (partialRootPaths.ARRAY = fallbackpath) Layout path (layoutRootPaths.ARRAY = fallbackpath)
_LOCAL_LANG	[ISOlang].key [default].key		Localization (key corresponds to the key in file: Resources/Private/Language/locallang.xlf)
_CSS_DEFAULT_STYLE			Inline stylesheets
SETUP: config.tx_extbase			Include plugin via TS

objects.
[sourceClass].
className =
[targetClass]
	At all places where the code refers to [sourceClass], an object of [targetClass] should be instantiated.		lib.foo = USER
			lib.foo {
			userFunc = TYPO3\CMS\Extbase\Core\Bootstrap->run
			extensionName = [ExtensionName]
			pluginName = [PluginName]
			vendorName = [VendorName]
			controller = [ControllerName]
			action = [actionName]
			switchableControllerActions {
			[ControllerName] {
			1 = [actionName]
			2 = [anotherActionName]

mvc. requestHandlers	Configures the 3 request handlers for FE, BE, CLI		

View API (\TYPO3\CMS\Extbase\Mvc\View\ViewInterface)

$this->view->assign($key,$value)	Assign $value to key $key in view
$this->view->assignMultiple(array $values)	Assign array $values view - use key of array element for key in view
$this->view->initializeView()	Initializing the concrete view implementation
$this->view->render()	Returns the rendered view

Request API (\TYPO3\CMS\Extbase\Mvc\RequestInterface) $this->request

get	set	Argument($argumentName, $value)	Gets/Sets the value of the specified argument
get	set	Arguments(array $arguments)	Gets/Sets the whole arguments array
is	set	Dispatched	Get/Sets the dispached flag for the request
get	set	ControllerActionName($actionName)	Gets/Sets the name of the action
get		ControllerExtensionKey()	Returns the extension name of the specified controller
get	set	ControllerExtensionName($extensionName)	Gets/Sets the extension name of the controller
get	set	ControllerName($controllerName)	Gets/Sets the name of the controller
get	set	ControllerObjectName($controllerName)	Gets/Sets the object name of the controller
get	set	ControllerSubpackageKey($subpackageKey)	Gets/Sets the subpackage key of the controller
get	set	ControllerVendorName($vendorName)	Gets/Sets the vendor name of the controller
		hasArgument($argumentName)	Checks if an argument of the given name exists (is set)
get	set	Format($format)	Gets/Sets requested representation format
get	set	PluginName($pluginName)	Gets/Sets the plugin name of the controller

ActionController API (\TYPO3\CMS\Extbase\Mvc\Controller\ActionController)

$this->actionMethodName	Name of current action (Default: indexAction)
$this->defaultViewObjectName	Default view: TYPO3\\CMS\\Fluid\\View\\TemplateView
$this->errorMethodName	Name of error action (Default: errorAction)
$this->request	Request object (of type \TYPO3\CMS\Extbase\Mvc\RequestInterface)
$this->response	Response object (of type \TYPO3\CMS\Extbase\Mvc\ResponseInterface)
$this->settings	Domain specific settings from TypoScript (array)
$this->view	current view (of type \TYPO3\CMS\Extbase\Mvc\View\ViewInterface)
$this->namespacesViewObjectNamePattern	@vendor\@extension\View\@controller\@action@format
function initializeView(\TYPO3\CMS\Extbase\Mvc\View\ViewInterface $view)	Initialize method for the commited view
function initializeAction()	Initialize method for all actions
function initialize[ActionName]Action()	Initialize method for specific action [ActionName]
function [actionName]Action()	Specific method for action [actionName]
function errorAction()	Called at arguments validation error.
function resolveView()	Prepares a view for the current action and stores it in $this->view
function resolveViewObjectName()	Determines the fully qualified view object name
$this->forward($actionName, $controllerName, $extensionName, array $arguments)	Internal redirect of request to another controller
$this->redirect($actionName, $controllerName, $extensionName, array $arguments, $pageUid, $delay = 0, $statusCode = 303)	External HTTP redirect to another controller
$this->redirectToURI($uri, $delay=0, $statusCode=303)	Redirect to URI
$this->throwStatus($statusCode, $statusMessage, $content)	Send HTTP status code

Repository API (\TYPO3\CMS\Extbase\Persistence\Repository)

$defaultOrderings = array ('pro'=>sorting)	Default order. Property and sorting (see custom query) is needed.
$defaultQuerySettings	Repository wide settings ($this->createQuery()->getQuerySettings())
$query->getQuerySettings()-> ->setRespectStoragePage ->setStoragePageIds ->setRespectSysLanguage ->setLanguageOverlayMode ->setLanguageMode	(TYPO3\CMS\Extbase\Persistence\Generic\QuerySettingsInterface) ->setLanguageUid ->setIgnoreEnableFields ->setEnableFieldsToBeIgnored ->setIncludeDeleted
add($object)	Add object to repository
countAll()	Returns the total number objects of this repository
countBy[PropertyName]($propertyValue)	Returns the number objects with [PropertyName] == $propertyValue
createQuery()	Creates a query (see custom queries)
findByUid($uid)	Finds an object matching the given identifier
findAll()	Find all objects of given type
findBy[PropertyName]($propertyValue)	Find all objects where property [PropertyName] == $propertyValue
findOneBy[PropertyName]($propertyValue)	Same as above, just find one (the first found) object (type object!)
remove($object)	Remove object from repository
removeAll()	Removes all objects of this repository
setDefaultOrderings(array $defOrderings)	Sets the default orderings
setDefaultQuerySettings(...)	Sets the default query settings: type of ...\QuerySettingsInterface
update($object)	Update stored object with $object

UriBuilder API (TYPO3\CMS\Extbase\Mvc\Web\Routing\UriBuilder)

Access in controller via $this->uriBuilder->...

section = '' | format = '' | createAbsoluteUri = FALSE | addQueryString = FALSE | addQueryStringMethod = '' | linkAccessRestrictedPages = FALSE | argumentsToBeExcludedFromQueryString = array() | targetPageUid = NULL | targetPageType = 0 | noCache = FALSE | useCacheHash = TRUE	UriBuilder Options there are Getter and Setter for all of them
setRequest(\TYPO3\CMS\Extbase\Mvc\RequestInterface $request) getRequest()	Sets/Gets request
reset()	Resets all UriBuilder options to default
uriFor($actionName = NULL, $controllerArguments = array(), $controllerName = NULL, $extensionName = NULL, $pluginName = NULL)	Creates an URI used for linking to an Extbase action
build()	Builds the URI

Custom queries (use in own method inside repository)

```
$query = $this->createQuery();
$query = $query->matching(
  $query->logicalAnd(
    $query->equals('blog', $blog),
    $query->equals('tags.name', $tag)
  )
)->setOrderings(array('date' =>\TYPO3\CMS\Extbase\Persistence\QueryInterface::ORDER_DESCENDING))->setLimit((integer)$limit)
return $query->execute();
```

$query = $this->createQuery();	Initializes a query. Access to below methods with $query->...
logicalAnd($constraints) logicalOr($constraints) logicalNot($constraint)	Performs a logical conjunction of the given constraints Performs a logical disjunction of the given constraints Performs a logical negation of the given constraint
statement($constraint)	SQL-Statement
get | setOrderings(array $orderings)	\TYPO3\CMS\Extbase\Persistence\QueryInterface::ORDER_ASCENDING \TYPO3\CMS\Extbase\Persistence\QueryInterface::ORDER_DESCENDING
get | setLimit($limit)	Sets the maximum size of the result set to limit (integer!)
get | setOffset($offset)	Sets the start offset of the result set to offset (integer!)
getConstraint()	Get the constraints back (if there are any)
execute(FALSE)	Executes the query against the backend and returns the result, TRUE = raw
count()	Get the first result back / $count = $query->execute()->count()
getFirst()	Get the first result back / $first = $query->execute()->getFirst()
matching($constraints)	The constraint used to limit the result set. Use methods below...
equals($propertyName, $operand, $caseSensitive = TRUE)	Returns an equals criterion used for matching objects against a query
like($propertyName, $operand)	Returns an equals criterion used for matching objects against a query
contains($propertyName, $operand)	It matches if the multivalued property contains the given operand
in($propertyName, $operand)	It matches if the property's value is contained in the multivalued operand
lessThan($propertyName, $operand)	Returns a less than criterion used for matching objects against a query
lessThanOrEqual ($propertyName, $operand)	Returns a less or equal than criterion used for matching objects against a query
greaterThan($propertyName, $operand)	Returns a greater than criterion used for matching objects against a query
greaterThanOrEqual($propertyName,$operand)	Returns a greater than or equal criterion used for matching objects against a query

Validator API (TYPO3\CMS\Extbase\Validation\Validator\AbstractValidator)

In annotation of Domain Model: @validate Validator1, Validator2(operand1 = value1, ...), ...
Validator class for Domain Model: class [Vendor]\[ExtensionName]\Validation\Validator\[DomainModelName]Validator extends \TYPO3\CMS\Extbase\Validation\Validator\AbstractValidator
Validation of controller arguments: @validate $variableName Validator1, Validator2, ...

Alphanumeric	TRUE, if the given property is a valid alphanumeric string [a-zA-Z0-9]*
Boolean(is=true|false)	Checks if the given value is true or false
Conjunction / Disjunction	AND/OR: Con/Disjunction(0=Validator1,1=Validator2, ...)
DateTime	Checks if the given value is a valid DateTime object
EmailAddress	Checks if the given value is a valid email address
Float	Checks if the given value is a valid float
Integer	Checks if the given value is a valid integer
NotEmpty	Checks if the given value is not empty (NULL or empty string)
NumberRange(startRange,endRange)	Returns TRUE, if the given value is a valid number in the given range
Number	Checks if the given value is a valid number
RegularExpression(regularExpression)	Returns TRUE, if the given value matches the given regular expression
StringLength(minimum,maximum)	Returns TRUE, if the given property ($value) is a valid string and its length
String	Returns TRUE, if the given property ($value) is a valid string
Text	Returns TRUE, if the given property ($propertyValue) is a valid text

Prop Validator (@validate \[VendorName]\[ExtensionName]\Validation\Validator\[ValName]Validator(option=value))

```
namespace [VendorName]\[ExtensionName]\Validation\Validator;
class ValNameValidator extends \TYPO3\CMS\Extbase\Validation\Validator\AbstractValidator {
  public function isValid($property) {
      $option1 = $this->options['option1'];                    // options access
      $this->addError('ErrorString', 1262341470,$arguments,$title);  // type \TYPO3\CMS\Extbase\Validation\Error
      return TRUE; // or FALSE                                 // validates if TRUE
  }
}
```

DO Validator ([VendorName]\[ExtensionName]\Validation\Validator\[DomainModelName]Validator) - e.g. for prop. title

```
$error = $this->objectManager->get('TYPO3\\CMS\\Extbase\\Validation\\Error', $apiValidationResult['title'], time());
$this->result->forProperty('title')->addError($error);
```

TYPO3 Fluid Cheat Sheet 1

GIT: git clone http://git.typo3.org/TYPO3v4/CoreProjects/MVC/fluid.git
Forge: http://forge.typo3.org/projects/package-typo3-fluid
Issue-Tracker: http://forge.typo3.org/projects/package-typo3-fluid/issues

v 3.01 / 03.03.2014
Patrick Lobacher / www.lobacher.de
compatible with TYPO3 CMS >= 6.2

LOBACHER.

FLUIDTEMPLATE (TypoScript cObject) {data} contains page properties

template	cObject	extbase.pluginName	Plugin name: string/stdWrap
file	string/strWrap	extbase.controllerExtensionName	Extension name: string/stdWrap
layoutRootPath	filepath/stdWrap	extbase.controllerName	Controller name: string/stdWrap
partialRootPath	filepath/stdWrap	extbase.controllerActionName	Action name: string/stdWrap
format	string/stdWrap	variables	(array of cObjects) -> Access: {variable}
stdWrap	stdWrap	settings	(array of keys) -> Access: {settings.key}

templateRootPaths, layoutRootPaths, partialRootPaths => filepath array (Fallback paths)

Addressing the view in action (controller class)

$this->view->assign('key', $value)	Makes Skey (which is of kind string, array or object)
$this->view->assignMultiple(array('key'=>$value))	available as {identifier} in fluid. Multiple assign is possible.
$this->view->render()	Forces rendering of template (default is at actions end)

Templates, Layouts, Partials (in directory Resources/Private/...)

Template (Templates/[Controller]/[Action].html)	<f:layout name="default" /> <f:section name="content"> <f:render partial="partName" arguments="..." /> <f:render partial="..." arguments="{_all}" /> </f:section>	Layout (Layouts/Default.html)	<f:render section="content" optional="true" />
		Partial (Partials/PartName.html)	...

Object accessor syntax

{name.property}	Object accessor: Result of getProperty() in model [name]
{name.key} / {name.number}	Associative / Numeric array: Element in array [name] with [key] / at position [number]

ViewHelper syntax

{namespace f=TYPO3\Fluid\ViewHelpers}	Declares the abbreviation f as namespace for TYPO3\Fluid\ViewHelpers
<f:vhname ATTRIBUTES> </f:vhname>	ViewHelper with the name [vhname]. Corresponding class is found at: typo3/sysext/fluid/Classes/ViewHelpers/VhnameViewHelper.php
ATTRIBUTES: attr="value" or attr="FLUID-ARRAY"	FLUID-ARRAYS will be listed like: {key1:value1,key2:value2} e.g. each="{0:1,1:2,...}" or values="{0:'odd',1:'even',...}" or arguments="{name:object}". Fluid arrays allowed in attributes only!

Inline syntax:
<f:vhname argument="value"... /> could be written as {f:vhname(argument:value, ...)} and
<f:format.nl2br><f:format.crop maxCharacters="20">{some.value}</f:format.crop></f:format.nl2br>
could be written as {some.value -> f.format.crop(maxCharacters:20) -> f:format.nl2br()}

Boolean expressions - like: condition="XXX operator YYY"	XXX,YYY is of type number, object accessor, array or ViewHelper inline syntax operator is one of: == != % >= > <= <

TagBasedViewHelper (general arguments for tag based viewhelper)

used with viewhelper: form (and all sub viewhelpers), image, link, renderFlashMessages

additionalAttributes	Associative array of additional tag-attributes	style	Individual CSS style
class	CSS classes for this tag	title	Tooltip text for this element
dir	Sets dir attribute - ltr or rtl	accesskey	Defines access key
id	Unique id	tabindex	Tab order for this element
lang	Language of this element (RFC 1766)	onclick	JavaScript for onclick event

alias ViewHelper - define alias of variables

<f:alias map="{x: foo.bar.baz, y: foo.bar.baz.name}">{x.name} or {y}</f:alias>

map	Mapping of alias - array - just valid inside alias tags

base ViewHelper - returns a base href tag

cObject ViewHelper - display TypoScript object (access with data & current)

<f:cObject typoscriptObjectPath="lib.customHeader" data="{article}" currentValueKey="{article.title}" />

typoscriptObjectPath	TypoScript object path which should be used
data	Data which is used for the rendering - same as <f:cObject>data</f:cObject>
currentValueKey	Key which sets the stdWrap property current

comment ViewHelper - comment out

count ViewHelper - counts elements

<f:count subject="{0:1, 1:2, 2:3, 3:4}" />

subject	The array or ObjectStorage to iterated over

cycle ViewHelper - iterates through given values

<f:for each="{0:1, 1:2, 2:3, 3:4}" as="foo">
<f:cycle values="{0: 'odd', 1: 'even'}" as="zebraClass"><li class="{zebraClass}">{foo}
</f:cycle>
</f:for>

values	Array of values which is used for iteration	as	Name of iteration variable

debug ViewHelper

<f:debug>{testVariables.array}</f:debug>

ansiColors	Add color to output (FALSE)	maxDepth	Sets the max recursion depth (8)
blacklistedClassNames	An array of class names (RegEx) to be filtered (NULL)	plainText	Plaintext output or HTML (FALSE)
blacklistedPropertyNames	An array of property names and/or array keys (RegEx) to be filtered (NULL)	title	Optional custom title for the debug output
inline	Inline rendering (FALSE)		

flashMessages ViewHelper - renders the flash messages (if there are any)

<f:flashMessages renderMode="div" />

renderMode	One of 'div' or 'ul' (ul is default)

for ViewHelper - foreach function

<f:for each="{fruit1: 'apple', fruit2: 'pear', fruit3: 'banana', fruit4: 'cherry'}" as="fruit" key="label">
{label}: {fruit}
</f:for>
<f:for each="{0:1, 1:2, 2:3, 3:4}" as="foo" iteration="fooIterator">
Index: {fooIterator.index} Cycle: {fooIterator.cycle} Total: {fooIterator.total}
{f:if(condition: fooIterator.isEven, then: 'Even')} {f:if(condition: fooIterator.isOdd, then: 'Odd')}{f:if(condition: fooIterator.isFirst, then: 'First')} {f:if(condition: fooIterator.isLast, then: 'Last')}
</f:for>

each	Array of values or objects which is used for iteration	key	Key of iteration variable
as	Name of iteration variable	reverse	If TRUE, direction will be reversed
iteration	The name of the variable to store iteration information index, cycle, isFirst, isLast, isEven, isOdd, total		

groupedFor ViewHelper - grouping of results

<f:groupedFor each="{0: {name: 'cherry', color: 'red'}, 1: {name: 'banana', color: 'yellow'}, 2: {name: 'strawberry', color: 'red'}}" as="fruitsOfThisColor" groupBy="color" groupKey="color">
{color} fruits:
<f:for each="{fruitsOfThisColor}" as="fruit" key="label">
{label}: {fruit.name}
</f:for>

</f:groupedFor>

as	Name of iteration variable	groupBy	Group by this property (property path possible)
each	Array or object which is used for iteration	groupKey	Name of variable which stores the grouping

if/then/else ViewHelper - if-then-else (w/o then if there is no else)

<f:if condition="somecondition"><f:then>...</f:then><f:else>...</f:else></f:if>
Shorthand-Syntax: {f:if(condition: '{rank} > 100', then: 'rank is > 100', else: 'rank is <= 100')}
<f:if condition="{rank} == {k:bar()}"> / <f:if condition="{foo.bar} == 'stringToCompare'">

condition	XX Comparator YY (e.g. <f:if condition="{rank} > 100">) Comparator is one of: ==, !=, <, <=, >, >= and % The % operator converts the result of the % operation to boolean. XX and YY can be one of: number / Object Accessor / Array / a ViewHelper / a String

image ViewHelper - displays an image (uri.image VH for link to image)

<f:image src="EXT:myext/Resources/Public/typo3_logo.png" alt="alt text" />
{f:image(src: 'EXT:myext/Resources/Public/logo.png', alt: 'alt text', minWidth: 30, maxWidth: 40)}

alt	Specifies an alternate text for an image	width	Width of the image ('m' / 'c' possible)
ismap	Specifies an image as a server-side image-map.	height	Height of the image ('m' / 'c' possible)
longdesc	Specifies the URL to a document	minWidth	Minimum width of the image
usemap	Specifies an image as a client-side image-map	minHeight	Minimum height of the image
src	Source of the image	maxWidth	Maximum width of the image
treatIdAsReference	Given src argument is a sys_file_reference record	maxHeight	Maximum height of the image

form ViewHelper <f:form ...> - form generation

absolute	Render absolute action URI (FALSE)	name	Name of form
action	Form action (default is current url)	noCache	Disable caching (FALSE)
actionUri	Overwrite the "action" attribute	noCacheHash	Supress the cHash (FALSE)
additionalParams	Add. action URI query params (Array)	pluginName	Target plugin (default is current)
addQueryString	Query params will be kept in the URI	pageUid	UID of target page
arguments	Additonal arguments (Array)	pageType	Type of target page (default is 0)
controller	Target controller (default is current)	object	Object bound to the form->property
enctype	MIME type for transmission	objectName	Name of the object bound to the form
extensionName	Target extension name (def. is current)	onreset/onsubmit	JavaScript handler
format	The requested format like "html"	section	The anchor to be added to the URI
fieldNamePrefix	Prefix to be added to all field names. If not set: tx_yourExtension_plugin	argumentsToBeExcludedFromQueryString	Arguments to be removed from the action URI
method	GET or POST (default)	hiddenFieldClassName	Class name of hidden fields

GENERAL ATTRIBUTES FOR ELEMENTS

errorClass	CSS class to set if there are errors	value	Value of element
name	Name of element	disabled	Displays the element disabled
property	Property of object bound through form	autofocus	for button, textfield & textarea

form.button - displays a button

<button type="submit" name="" value="">Send Mail</button>

form	1 or more forms the button belongs to	formnovalidate	No validation of form data
formaction	Form action	formtarget	Target (e.g. _blank, _self)
formenctype	Encoding	type	Type (e.g. button, reset, submit)
formmethod	Method (POST or GET)		

form.checkbox - displays a checkbox

<f:form.checkbox name="myCheckBox" value="someValue" checked="{object.value} == 5" />

checked	If TRUE then checkbox is checked	property="myProperty" value="myValue" if {object.myProperty} == myValue => checked	

form.hidden - displays a hidden field

<f:form.hidden name="myHiddenValue" value="42" />

form.password - displays a password input field

<f:form.password name="myPassword" />

maxlength	Maximum length of field	size	Length of input field
readonly	Readonly attribute of field		

form.radio - displays a radio button

<f:form.radio name="myRadioButton" value="someValue" checked="{object.value} == 5" />

checked	If TRUE then radiobutton is checked	see form.checkbox above (property="...)	

form.select - displays a selector box

<f:form.select name="users" options="{userArray}" optionValueField="id" optionLabelField="firstName" />

selectAllByDefault	Select all options	size	Length of selector box
optionLabelField	Property label field	multiple	Display a multi-select box
optionValueField	Property value field (uid default)	prepend OptionLabel / An option at first position - label	
options	Assoc array or object used for options	prepend OptionValue / An option at first position - value	
sortByOptionLabel	List will be sorted by label		

form.submit - displays a submit button

<f:form.submit value="Send Mail" />

form.textarea - displays a text area

<f:form.textarea name="myTextArea" value="This is shown inside the textarea" rows="5" cols="30" />

cols	Number of cols	rows	Number of rows
placeholder	Placeholder text		

form.textfield - displays an input field

<f:form.textfield name="myTextBox" value="default value" />

maxlength	Maximum length of field	required	Required attribute of field
placeholder	Placerholder attribute of field	size	Length of input field
readonly	Readonly attribute of field	type	Type (e.g. text, email, url, ...)

form.upload - displays an upload field (just works with enctype="multipart/form-data" in form tag)

<f:form.upload name="file" />

form.validationResults (use attribute „for" to restrict it to a property, e.g. for="blog.title" and „as" as error name)

<f:form.validationResults><f:if condition="{validationResults.flattenedErrors}"><ul class="errors"><f:for each="{validationResults.flattenedErrors}" as="errors" key="propertyPath">
{propertyPath}
<f:for each="{errors}" as="error">
{error.code}: {error}
</f:for>
</f:for>
</f:if></f:form.validationResults>

TYPO3 Fluid Cheat Sheet 2

GIT: git clone http://git.typo3.org/TYPO3v4/CoreProjects/MVC/fluid.git
Forge: http://forge.typo3.org/projects/package-typo3-fluid
Issue-Tracker: http://forge.typo3.org/projects/package-typo3-fluid/issues

v 3.01 / 03.03.2014
Patrick Lobacher / www.lobacher.de
compatible with TYPO3 CMS >= 6.2
LOBACHER.

format ViewHelper - formats in different ways (use value attr or child VH)

format.bytes - Formats an integer with a byte count into human-readable form

```
{fileSize -> f:format.bytes(decimals: 2, decimalSeparator: ',', thousandsSeparator: '.')}
```

decimals	Number of digits after decimal point	thousandsSeparator	Character for thousands
decimalSeparator	Decimal point character		

format.cdata - Outputs an argument/value without any escaping and wraps it with CDATA tags.

```
<f:format.cdata>{string}</f:format.cdata>  | <f:format.cdata value="{string}" />  | {string -> f:format.cdata()}
```

format.crop - Use this view helper to crop the text between its opening and closing tags

```
<f:format.crop maxCharacters="17" append=" [more]">This is some very long text</f:format.crop>
```

append	String, which is append at crop position	respectHtml	Cropped string will respect HTML
maxCharacters	Max. characters which are displayed	respectWordBoundaries	Crops only at word boundaries

format.currency - displays currency conversions

```
{someNumber -> f:format.currency(thousandsSeparator: ',', currencySign: '€')}
```

currencySign	The currency sign (like $, €, ...)	prependCurrency	Prepend currency sign (FALSE)
decimals	Set decimals places	separateCurrency	Space beetween sign (FALSE)
decimalSeparator	Character for decimal separation	thousandsSeparator	Character for thousands sep.

format.date - displays dates (use @ if you handle with unix timestamps) - default is ["SYS"]["ddmmyy"]

```
<f:format.date format="H:i">{dateObj}</f:format.date><f:format.date format="d.m.Y - H:i:s">@{timestamp}</f:format.date>
```

format	Format in date() syntax	date	DateTime object or string

format.htmlentitiesDecode - Applies html_entity_decode() to a value

```
{text -> f:format.htmlentitiesDecode(encoding: 'ISO-8859-1')}
```

encoding	Encoding	keepQuotes	Keep Quotes (FALSE)

format.htmlentities - Applies htmlentities() escaping to a value

```
<f:format.htmlentities>{text}</f:format.htmlentities>  | {text -> f:format.htmlentities(encoding: 'ISO-8859-1')}
```

encoding	Encoding	keepQuotes	Keep Quotes (FALSE)

format.htmlspecialchars - Applies htmlspecialchars() escaping to a value

```
<f:format.htmlspecialchars>{text}</f:format.htmlspecialchars>  | {text -> f:format.htmlspecialchars(encoding: 'ISO-8859-1')}
```

encoding	Encoding	keepQuotes	Keep Quotes (FALSE)
doubleEncode	If FALSE html entities won't be encoded		

format.html - Renders strings with TYPO3 parseFunc

```
<f:format.html parseFuncTSPath="lib.parseFunc">foo <b>bar</b>. Some <LINK 1>link</LINK>.</f:format.html>
```

parseFuncTSPath	Path to own parseFunc. Default is "lib.parseFunc_RTE"

format.nl2br - Wrapper for PHP function nl2br

```
<f:format.nl2br>{text_with_linebreaks}</f:format.nl2br> / {text_with_linebreaks -> f:format.nl2br()}
```

format.number - Formats numbers

```
<f:format.number decimals="1" decimalSeparator="," thousandsSeparator=".">423423.234</f:format.number>
```

decimals	Numbers after comma. Default is "2"	thousandsSeparator	Character for thousands sep.
decimalSeparator	Character for decimal separation		

format.padding - Wrapper for PHP function str_pad()

```
<f:format.padding padLength="10" padString="-=">TYPO3</f:format.padding>
```

padLength	Length of outputted string	padType	Values: right (default), left, both
padString	String which is used for filling up		

format.printf - Wrapper for PHP function printf()

```
<f:format.printf arguments="{number: 362525200}">%.3e</f:format.printf>
<f:format.printf arguments="{0: 3, 1: 'Kasper'}">%2$s is great, TYPO%1$d too.</f:format.printf>
```

arguments	Arguments for printf as array

format.raw - Outputs an argument/value without any escaping

```
<f:format.raw>{string}</f:format.raw> | <f:format.raw value="{string}" /> | {string -> f:format.raw()}
```

format.stripTags - Removes tags from the given string (applying PHPs strip_tags() function)

```
<f:format.stripTags>Some Text with <b>Tags</b> and an &Uuml;mlaut.</f:format.stripTags>
```

format.urlencode - Encodes the given string (applying PHPs rawurlencode() function)

```
<f:format.urlencode>foo @+%/</f:format.rawurlencode> | {text -> f:format.urlencode()}
```

translate ViewHelper - (from Resources/Private/Language/locallang.xlf)

```
<f:translate key="LLL:EXT:myext/Resources/Private/Language/locallang.xml:key1" />
{f:translate(key: 'argumentsKey', arguments: {0: 'dog', 1: 'fox'}, default: 'default value')}
```

arguments	Arguments	htmlEscape	If FALSE, the output will not be escaped (TRUE)
default	Default key if 'key' is not found	id	Key in locallang file (has precedence over key)
extensionName	UpperCamelCased extension key	key	Key in locallang file

link/uri ViewHelper - generates URLs (link.x with tag / uri.x without tag)

name	Specifies the name of an anchor	target	Target parameter
rel	Rel: current => linked document	rev	Rel: linked => current document

link.action / uri.action - generates extbase action links

```
<f:link.action action="show">action link</f:link.action>
```

absolute	Render absolute URI	format	The format, e.g. ".html"
action	Target action	linkAccessRestrictedPages	Show even access restricted pages
additionalParams	Additional parameters	noCache	Deactivate cache for target page
addQueryString	Query params kept in the URI	noCacheHash	No cHash parameter
addQueryStringMethod	GET | POST | GET,POST | POST,GET		
arguments	Arguments	pageType	Page type (default 0)
argumentsToBeExcludedFromQueryString	Arguments to be removed from the action URI	pageUid	Target page UID (default current)
controller	Target Controller	pluginName	Target Plugin
extensionName	Target Extension	section	Anchor to be added to the URI

link.email / uri.email - generates an email link

```
<f:link.email email="foo@bar.tld">some custom content</f:link.email>
```

email	Email address		

link.external / uri.external - generates links to external targets

```
<f:link.external uri="http://www.typo3.org" target="_blank">external link</f:link.external>
```

defaultScheme	Scheme - 'http' is default	uri	Target URL

link.page / uri.page - generates links to TYPO3 pages

```
<f:link.page pageUid="1" additionalParams="{extension_key: {foo: 'bar'}}">page link</f:link.page>
```

absolute	Render absolute URI	noCache	Deactivate cache for target page
additionalParams	Additional parameters	noCacheHash	No cHash parameter
addQueryString[Method]	Query params kept in the URI	pageType	Page type (default 0)
argumentsToBeExcludedFromQueryString	Arguments to be removed from the action URI	linkAccessRestrictedPages	Show even access restricted pages
section	Anchor	pageUid	Target page UID (default current)

uri.resource - creating URIs to resources

```
<link href="{f:uri.resource(path:'css/stylesheet.css')}" rel="stylesheet" />
```

absolute	Render absolute URI	path	The path & filename of the resource
extensionName	Target extension name		

security ViewHelper

security.ifAuthenticated - implements an ifAuthenticated/else condition for FE users/groups

```
<f:security.ifAuthenticated><f:then>Access.</f:then><f:else>No access.</f:else></f:security.ifAuthenticated>
```

security.ifHasRole - implements an ifHasRole/else condition for FE users/groups

```
<f:security.ifHasRole role="Admin"><f:then>You are Admin.</f:then><f:else>You aren't.</f:else></f:security.ifHasRole>
```

role	Group role (either UID or title)

switch / case ViewHelper

```
<f:switch expression="{person.gender}"><f:case value="male">Mr.</f:case><f:case default="TRUE">Mrs./Ms.</f:case></f:switch>
```

be ViewHelper - backend module viewhelper

be.container - View helper which allows you to create extbase based modules in the style of TYPO3 default modules.

```
<f:be.container pageTitle="foo" enableJumpToUrl="false" enableClickMenu="false" loadPrototype="false" loadScriptaculous="false" scriptaculousModule="someModule,someOtherModule" loadExtJs="true" loadExtJsTheme="false" extJsAdapter="jQuery" enableExtJsDebug="true" loadJQuery="true" includeCssFiles="0: ,{f:uri.resource(path:\'Styles/Styles.css\')}" includeJsFiles="0: ,{f:uri.resource(path:\'JavaScript/Library1.js\')}', 1: ,{f:uri.resource(path:\'JavaScript/Library2.js\')}'" addJsInlineLabels="{0: ,label1', 1: ,label2'}">your module content</f:be.container>
```

addJsInlineLabels	Custom labels to add JS inline labels	loadExtJs	Specifies whether to load ExtJS
enableClickMenu	If TRUE (default), loads clickmenu.js	loadExtJsTheme	Whether to load ExtJS „grey" theme
enableExtJsDebug	If TRUE, debug version of ExtJS is loaded.	loadJQuery	Whether to loadjQuery library
enableJumpToUrl	If TRUE (default), includes "jumpToUrl"	loadPrototype	Specifies whether to load prototype lib
extJsAdapter	Load alternative adapter (ext-base is def.)	loadScriptaculous	Specifies whether to load scriptaculous
includeCssFiles	List of custom CSS files to be loaded	pageTitle	Title tag of the module
includeJsFiles	List of custom JS files to be loaded	scriptaculousModule	Add. modules for scriptaculous

be.pageInfo - return page info icon

be.pagePath - current page path, prefixed with „Path:" and wrapped in a span with the class "typo3-docheader-pagePath"

be.tableList - renders a record list as known from the TYPO3 list module

```
<f:be.tableList tableName="fe_users" fieldList="{0: 'name', 1: ,email'}" storagePid="1" levels="2" filter='foo' recordsPerPage="10" sortField="name" enableClickMenu="false" clickTitleMode="info" alternateBackgroundColors="true" />
```

alternateBackgroundColors	/ Rows have alt bgcolors	readOnly	Edit icons won't be shown if TRUE
clickTitleMode	"edit", "show" or "info"	recordsPerPage	Amount of records to be displayed
enableClickMenu	Enables context menu	sortDescending	Records sorted in descending order
fieldList	List of fields to be displayed (arry)	sortField	Table field to sort the results by
filter	Corresponds to „Search String" textbox	storagePid	Records are fetched from this PID
levels	Corresponds to the level selector	tableName	Name of the database table

be.buttons.csh - CSH button as known from the TYPO3 backend

```
<f:be.buttons.csh table="xMOD_csh_corebe" field="someCshKey" iconOnly="1" styleAttributes="border: 1px solid red" />
```

iconOnly	If set, full text will never be shown	styleAttributes	Additional style-attribute
field	Field name (CSH locallang main key)	table	Table name ('_MOD_'+module)

be.buttons.icon - returns button with icon (possible icons in ['coreSpriteImageNames'] typo3/systext/core/ext_tables.php)

```
<f:be.buttons.icon uri="{f:uri.action(action:'new')}" icon="new_el" title="Create new Foo" />
```

icon	One of list above	uri	Target URI
title	Title attribute		

be.buttons.shortcut - returns shortcut button as known from the TYPO3 backend.

```
<f:be.buttons.shortcut getVars="{0: ,M', 1: ,myOwnPrefix'}" setVars="{0: ,function'}" />
```

getVars	List of GET variables to store	setVars	List of SET[] variables to store

be.menus.actionMenu / be.menus.actionMenuItem - select box, used to switch betweenmultiple actions and controllers

```
<f:be.menus.actionMenu><f:be.menus.actionMenuItem label="{f:translate(key='overview')}" controller="Blog" action="index" /></f:be.menus.actionMenu>
```

action	Action to be called	controller	Controller to be called
arguments	Arguments	label	Label of option tag

be.security.ifAuthenticated - implements an ifAuthenticated/else condition for BE users/groups (like security.ifAuthenticated)

be.security.ifHasRole - implements an ifHasRole/else condition for BE users/groups. (like security.ifHasRole)

API: Write own ViewHelper with name [Vhname] -

Put file [Vhname]ViewHelper.php in Classes/ViewHelpers/ with the following class definition:

initializeArguments()	Register arguments within this method with $this->registerArgument($name, $type, $description, $required, $defaultValue)
initialize()	Code before rendering of viewhelper

class [Vhname]ViewHelper extends \TYPO3\CMS\Fluid\Core\ViewHelper\AbstractViewHelper {}

$arguments	Associative array of arguments of viewhelper. Accessible within render()
$templateVariableContainer	Contains all variables which are accessible in the template. Change with add() & remove(), read with get()
$controllerContext	Context of controller - serves as API for the controller. Supports the following properties: request, response, arguments, uriBuilder, flashMessageContainer, ...
$viewHelperVariableContainer	/ Container for exchanging data between viewhelpers. Add with add() & read with get()
render($arg1,$arg2,...)	Render method which is responsible for output of viewhelper. It's required.
renderChildren()	Called within render() will return all rendered content from child elements

class [Vhname]ViewHelper extends \TYPO3\CMS\Fluid\Core\ViewHelper\AbstractTagBasedViewHelper { }

$tagName	Name of Tag which will be gererated by the viewhelper (default is `div`)
$tag	Instance TagBuilder. API: setContent(StagContent), getContent(), forceClosingTag(SforceClosingTag), addAttribute(SattributeName, SattributeValue, SescapeSpecialCharacters = TRUE), addAttributes(array Sattributes, SescapeSpecialCharacters = TRUE), removeAttribute(SattributeName), reset(), render()
registerUniversalTagAttributes()	Registers all universal tag attributes like class, dir, id, lang, style, title, accesskey, ...

widget ViewHelper

widget.autocomplete - autocomplete for fields (needs Static TS 'Fluid: Default AJAX configuration')

```
<f:widget.autocomplete for="name" objects="{posts}" searchProperty="author">
```

for	Name of field for autocomplete	searchProperty	Property of object to search in
objects	Objects to search in		

widget.link / widget.uri - create links to extbase actions within widgets (link with a-tag / uri without)

```
<f:widget.link action="show">link</f:widget.link>
```
(link and paginate have additional option: addQueryStringMethod)

action	Same arguments as action.link or uri	ajax	TRUE for link to ajax widget
arguments	Arguments	section	The anchor to be added to the URI

widget.paginate / be.widget.paginate - renders a pagination of objects (FE and BE)

```
<f:widget.paginate objects="{blogs}" as="paginatedBlogs" configuration="{itemsPerPage: 3, insertAbove: 1, insertBelow: 1}"><ul><li><f:for each="{paginatedBlogs}" as="blog">{blog.title}</li></f:for></ul></f:widget.paginate>
```

as	Identifier of paginated objects	objects	Objects that will be paginated
configuration	itemsPerPage, insertAbove, insertBelow, maximumNumberOfLinks (FE), recordsLabel (BE)		