TYPO3.Flow - Feature # 33838

Status:	Resolved	**Priority:**	Should have
Author:	Dominique Feyer	**Category:**	Persistence
Created:	2012-02-10	**Assigned To:**	Karsten Dambekalns
Updated:	2015-03-06	**Due date:**	
PHP Version:			
Has patch:	No		
Complexity:			
Subject:	Add a way to configure Doctrine Mapping Type		

Description

As I need some JSON object storage in FLOW3, i need to use a custome Doctrine Mapping Type to handle to conversion between object and database.

As the documentation of Doctrine2 say here:
http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/reference/basic-mapping.html#doctrine-mapping-types

Doctrine looks really flexible on this point, and if you found a solution to enable this in FLOW3 this could amazing.

The main problem is that $bootstrap->getEarlyInstance('Doctrine\ORM\EntityManager') or getObjectManager()->get('Doctrine\ORM\EntityManager') always return NULL in the boot method of the file package.php

Associated revisions

Revision e1fc9669 - 2015-03-05 15:01 - Karsten Dambekalns

[FEATURE] Make custom Doctrine mapping types configurable

This removes the hardcoded registration of the ObjectArray mapping type
and instead introduces mapping type configuration from settings::

```
TYPO3:
  Flow:
    persistence:
      doctrine:
      # DBAL custom mapping types can be registered here
        dbal:
          mappingTypes:
            'mytype':
              dbType: 'db_mytype'
              className: 'Acme\Demo\Doctrine\DataTypes\MyType'
```

See the Doctrine documentation for more details:
http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/cookbook/custom-mapping-types.html

Change-Id: Iab0f14117b30a4924a7947af44b9516c241912da
Resolves: #33838
Releases: master, 3.0

Revision a72251ed - 2015-03-06 10:16 - Karsten Dambekalns

[FEATURE] Make custom Doctrine mapping types configurable

This removes the hardcoded registration of the ObjectArray mapping type
and instead introduces mapping type configuration from settings::

```
TYPO3:
  Flow:
    persistence:
      doctrine:
      # DBAL custom mapping types can be registered here
        dbal:
          mappingTypes:
            'mytype':
              dbType: 'db_mytype'
              className: 'Acme\Demo\Doctrine\DataTypes\MyType'
```

See the Doctrine documentation for more details:
http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/cookbook/custom-mapping-types.html

Change-Id: Iab0f14117b30a4924a7947af44b9516c241912da
Resolves: #33838
Releases: master, 3.0

Revision e9c30e84 - 2015-06-03 18:59 - Bastian Waidelich

[BUGFIX] Adjust YAML schema to recent changes

This is a follow-up to the "Make custom Doctrine mapping types
configurable" feature (Iab0f14117b30a4924a7947af44b9516c241912da)
adjusting the YAML schema.

This also tweaks the existing schema rules according to our CGL.

Change-Id: Ie6a040fa897a422d79769810dfd3a36ddc029bba
Related: #33838
Releases: master, 3.0

Revision 1cf6b11d - 2015-06-04 10:25 - Bastian Waidelich

[BUGFIX] Adjust YAML schema to recent changes

This is a follow-up to the "Make custom Doctrine mapping types
configurable" feature (Iab0f14117b30a4924a7947af44b9516c241912da)
adjusting the YAML schema.

This also tweaks the existing schema rules according to our CGL.

Change-Id: Ie6a040fa897a422d79769810dfd3a36ddc029bba
Related: #33838
Releases: master, 3.0

History

#1 - 2012-02-10 14:48 - Karsten Dambekalns

- Category set to Persistence

- Status changed from New to Accepted

#2 - 2012-02-10 14:52 - Dominique Feyer

Same problem in 1.0.2

#3 - 2012-05-25 12:57 - Karsten Dambekalns

- Tracker changed from Bug to Feature

#4 - 2013-10-21 17:16 - Adrian Föder

it looks like you can do the following, at least, Package.php:

```
1public function boot(\TYPO3\Flow\Core\Bootstrap $bootstrap) {
2    $dispatcher = $bootstrap->getSignalSlotDispatcher();
3    $dispatcher->connect('TYPO3\Flow\Core\Booting\Sequence', 'afterInvokeStep', function(\TYPO3\Flow\Core\Booting\Step $step) {
4        if ($step->getIdentifier() === 'typo3.flow:persistence') {
5            \Doctrine\DBAL\Types\Type::addType('yourtype', 'Your\Whatever\Type');
6        }
7    });
8}
```

The original "idea" of this approach can be seen here,
https://github.com/Flowpack/Flowpack.ElasticSearch/blob/master/Classes/Flowpack/ElasticSearch/Package.php

// edit/update: I doubt whether that signal/slot-delay is necessary at all? Since Type::addType is statically accessed, no need to "wait" for the EntityManger or so to be available and factory'd...?

#5 - 2013-10-21 19:16 - Alexander Schnitzler

As Adrian said it's possible to use \Doctrine\DBAL\Types\Type::addType('yourtype', 'Your\Whatever\Type'); and so on to add a new mapping type but unfortunately flow does not have a look at the mapping at all.

So imagine you want to use a geographical point object with latitude and longitude as a model property but save it as string (0.3434224,12.433445) the annotation reflection expects you to add a OneToOne-Statement though you do not want to save a related object. So the annotation parser somewhen and somehow has to look for custom mapping types.

#6 - 2013-11-04 09:21 - Karsten Dambekalns

Alexander Schnitzler wrote:

> *As Adrian said it's possible to use \Doctrine\DBAL\Types\Type::addType('yourtype', 'Your\Whatever\Type'); and so on to add a new mapping type but unfortunately flow does not have a look at the mapping at all.*

It does take it into account, but you need register in two places to make it work. Check https://review.typo3.org/22825 for how it works. To be fair, this isn't accessible to the "end user" so far.

#7 - 2015-03-05 15:01 - Gerrit Code Review

- Status changed from Accepted to Under Review

Patch set 2 for branch **master** of project **Packages/TYPO3.Flow** has been pushed to the review server.
It is available at http://review.typo3.org/37559

#8 - 2015-03-06 10:16 - Gerrit Code Review

Patch set 1 for branch **3.0** of project **Packages/TYPO3.Flow** has been pushed to the review server.
It is available at http://review.typo3.org/37585

#9 - 2015-03-06 10:30 - Karsten Dambekalns

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:e1fc9669266532c2a8604f624cef38809f8548d5.