élao

Agence Web - iPhone & iPad Apps - Experts Symfony

Doctrine Sheets

Symfony Doctrine Commands

cache:clear-metadata	Clears all metadata cache for an entity manager.
cache:clear-query	Clears all query cache for an entity manager.
cache:clear-result	Clears result cache for an entity manager.
database:create	Creates the configured databases.
database:drop	Drops the configured databases.
ensure-production-settings	Verify that Doctrine is properly configured for a production environment.
fixtures:load	Load data fixtures to your database.
generate:crud	Generates a CRUD based on a Doctrine entity.
generate:entities	Generates entity classes and method stubs from your mapping information
generate:entity	Generates a new Doctrine entity inside a bundle
generate:form	Generates a form type class based on a Doctrine entity
mapping:convert	Convert mapping information between supported formats.
mapping:import	Imports mapping information from an existing database
mapping:info	Shows basic information about all mapped entities
query:dql	Executes arbitrary DQL directly from the command line.
query:sql	Executes arbitrary SQL directly from the command line.
schema:create	Executes (or dumps) the SQL needed to generate the database schema
schema:drop	Executes (or dumps) the SQL needed to drop the current database schema
schema:update	Executes (or dumps) the SQL needed to update the database schema to match the current mapping metadata
schema:validate	Validates the doctrine mapping files

DQL (Doctrine Query Language)

DQL Functions

IDENTITY(single_association_path_expression)	Retrieve the foreign key column of association of the owning side.
ABS(arithmetic_expression)	Returns the absolute value of the arithmetic_expression parameter.
CONCAT(str1, str2)	Concats the given strings.
CURRENT_DATE()	Returns the current date.
CURRENT_TIME()	Returns the current time.
CURRENT_TIMESTAMP()	Returns a timestamp of the current date and time.
LENGTH(str)	Returns the length of the given string.
LOCATE(needle, haystack [, offset])	Locate the first occurrence of the substring in the string.
LOWER(str)	Returns the string lowercased.
MOD(a, b)	Returns the remainder of division of the a number by the b number.
SIZE(collection)	Return the number of elements in the specified collection.
SQRT(q)	Return the square-root of q.
SUBSTRING(str, start [, length])	Return substring of given string.
TRIM([LEADING | TRAILING | BOTH] [â€˜trcharâ€™ FROM] str)	Trim the string by the given trim char, defaults to whitespaces.
UPPER(str)	Return the upper-case of the given string.
DATE_ADD(date, days, unit)	Add the number of days to a given date. (Supported units are DAY, MONTH).
DATE_SUB(date, days, unit)	Substract the number of days from a given date. (Supported units are DAY, MONTH).
DATE_DIFF(date1, date2)	Calculate the difference in days between date1 and date2.

AGGREGATE FUNCTIONS

AVG, COUNT, MIN, MAX, SUM

OTHERS EXPRESSION

ALL/ANY/SOME
BETWEEN a AND b / NOT BETWEEN a AND b
IN / NOT IN
LIKE / NOT LIKE
IS NULL / IS NOT NULL
EXISTS / NOT EXISTS

DQL / Examples

```php
<?php
$query = $em->createQuery('SELECT a
                           FROM CmsArticle a
                           JOIN a.user u
                           ORDER BY u.name ASC');

$articles = $query->getResult(); // array of CmsArticle objects
```

Hydration Mode

Query::HYDRATE_OBJECT
Hydrates the result set into the object graph.
Query::HYDRATE_ARRAY
Hydrates the result set into an array that represents the object graph.
Query::HYDRATE_SCALAR
If you want to return a flat rectangular result set instead of an object graph you can use scalar hydration.
Query::HYDRATE_SINGLE_SCALAR
If you have a query which returns just a single scalar value you can use single scalar hydration

QueryBuilder / Low Level

add
setParameter, setParameters
getQuery
setFirstResult, setMaxResults

QueryBuilder / Helpers

select, delete, update
set
from
innerJoin, leftJoin
where, andWhere, orWhere
groupBy, addGroupBy
having, andHaving, orHaving
orderBy, addOrderBy

QueryBuilder / Expr / Conditional objects

andX orX

QueryBuilder / Expr / Comparaison objects

eq, neq
lt, lte
gt, gte
isNull, isNotNull

QueryBuilder / Expr / Arithmetic objects

prod diff sum quot

QueryBuilder / Expr / Pseudo-function objects

exists all some any not in notIn like between

QueryBuilder / Expr / Function objects

trim concat substr lower upper length avg max min abs sqrt count countDistinct

Examples / Low Level

```php
<?php
$qb->add('select', 'u')
   ->add('from', 'User u')
   ->add('where', 'u.id = ?1')
   ->add('orderBy', 'u.name ASC')
   ->setParameter(1, 100);
```

Examples / Helpers

```php
<?php
$qb->select('u')
   ->from('User u')
   ->where('u.id = ?1')
   ->orderBy('u.name ASC');
   ->setParameter(1, 100);
```

Examples / Expr

```php
<?php
$qb->add('select', new Expr\Select(array('u')))
    ->add('from', new Expr\From('User', 'u'))
    ->add('where', $qb->expr()->orX(
        $qb->expr()->eq('u.id', '?1'),
        $qb->expr()->like('u.nickname', '?2')
    ))
    ->add('orderBy', new Expr\OrderBy('u.name', 'ASC'));
```

Execute Query / Shortcuts

getResult
Get a result set with HYDRATE_OBJECT
getArrayResult
Get a result set with HYDRATE_ARRAY
getScalarResult
Get a result set with HYDRATE_SCALAR
getOneOrNullResult
Get a single result, return null if no result, throw an exception if more
getSingleResult
Get a single result, throw an exception if no result or more than 1 result
getSingleScalarResult
Get a single result with HYDRATE_SINGLE_SCALAR

Class annotations / Main

@Entity
- repositoryClass
- readOnly

@HasLifecycleCallbacks

@Table
- name
- indexes
- uniqueConstraints

@ChangeTrackingPolicy

@UniqueConstraint
- name
- columns

Properties annotations / Main

@Column		
	type	string, integer, smallint, bigint, boolean, decimal, date, time, datetime, text, object, array, float
	name	Column name (string)
	length	Column length (integer)
	precision	Value precision (decimal number)
	scale	Value scale (decimal number)
	unique	Unique key (true of false)
	nullable	Column can be null (true of false)
	columnDefinition	
@Id		
@Index		
	name	Index name (string)
	columns	Related columns (strings array)
@GeneratedValue		
	strategy	AUTO, SEQUENCE, TABLE, IDENTITY, NONE
@SequenceGenerator		
	sequenceName	
	allocationSize	
	initialValue	
@Version		

Properties annotations / Associations

@OneToOne		
	targetEntity	FQCN of the referenced target entity (string)
	cascade	persist, remove, merge, detach, all
	fetch	Fetch type (LAZY or EAGER)
	orphanRemoval	Remove orphan (true or false)
	inversedBy	Field in the entity that is the inverse side (string)
@OneToMany		
	targetEntity	FQCN of the target entity (string)
	cascade	persist, remove, merge, detach, all
	orphanRemoval	Remove orphan (true or false)
	mappedBy	Property on the targetEntity that is the owning side (string)
	fetch	Fetch type (LAZY or EAGER)
	indexBy	
@ManyToOne		
	targetEntity	FQCN of the target entity (string)
	cascade	persist, remove, merge, detach, all
	fetch	Fetch type (LAZY or EAGER)
	inversedBy	Field in the entity that is the inverse side (string)
@ManyToMany		
	targetEntity	FQCN of the target entity (string)
	mappedBy	Property on the targetEntity that is the owning side (string)
	inversedBy	Field in the entity that is the inverse side (string)
	cascade	persist, remove, merge, detach, all
	fetch	Fetch type (LAZY or EAGER)
	indexBy	
@JoinTable		
	name	Database name of the join table
	joinColumns	An array of @JoinColumn
	inverseJoinColumns	An array of @JoinColumn
@JoinColumn		
	name	Column name that holds the foreign key
	referencedColumnName	Name of the primary key identifier used for join
	unique	Is this relation exclusive between the affected entities
	nullable	Related entity is required
	onDelete	Cascade Action (Database-level)
	columnDefinition	
@OrderBy		

Class annotations / Inheritance

@DiscriminatorColumn @DiscriminatorMap @InheritanceType @MappedSuperClass

Method annotations / Callbacks

@PostLoad @PostPersist @PostRemove @PostUpdate @PrePersist @PreRemove @PreUpdate

Class annotations / Named Query

@ColumnResult @EntityResult @FieldResult @NamedNativeQuery @SqlResultSetMapping

Annotations Exemples

```php
<?php

use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Table(name="contact")
 * @ORM\Entity(repositoryClass="Elao\ErpBundle\Entity\ContractRepository")
 * @ORM\HasLifecycleCallbacks
 */
class Contract
{
    /**
     * @ORM\Column(name="id", type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=32, unique=true, nullable=false)
     */
    private $reference;

    /**
     * @ORM\OneToMany(
     *     targetEntity="Elao\ErpBundle\Entity\Invoice",
     *     mappedBy="contract",
     *     cascade={"persist"},
     *     orphanRemoval=true
     * )
     * @ORM\OrderBy({"createdAt" = "DESC"})
     */
    private $invoices;

    /**
     * @ORM\ManyToOne(
     *     targetEntity="Elao\ErpBundle\Entity\Client",
     *     inversedBy="contracts"
     * )
     * @ORM\JoinColumn(
     *     name="client_id",
     *     referencedColumnName="id",
     *     nullable=false,
     *     onDelete="CASCADE"
     * )
     */
    private $client;

    /**
     * @ORM\ManyToMany(
     *   targetEntity="Elao\UserBundle\Entity\User",
     *   inversedBY="contracts"
     * )
     * @ORM\JoinTable(name="contract_users",
     *      joinColumns={
     *           @ORM\JoinColumn(name="contract_id", referencedColumnName="id")},
     *      inverseJoinColumns={
     *          @ORM\JoinColumn(name="user_id", referencedColumnName="id")}
     * )
     */
    private $users;

    /**
     * @ORM\PrePersist
     * @ORM\PreUpdate
     */
    public function prePersist()
    {
        $this->updatedAt = new \DateTime();
    }
}
```

```php
<?php
/**
 * @ORM\Entity
 * @ORM\InheritanceType("SINGLE_TABLE")
 * @ORM\DiscriminatorColumn(name="discr", type="string")
 * @ORM\DiscriminatorMap({"person" = "Person", "employee" = "Employee"})
 */
class Person
{
    // ...
}

/**
 * @ORM\Entity
 */
class Employee extends Person
{
        // ...
}
```