

Symfony Doctrine Commands

ensure-production-settings	Verify Doctrine for production
data:load	Load data fixtures to your database
database:create	Create the configured databases
database:drop	Drop the configured databases
generate:entities	Generate entity classes and method stubs
generate:entity	Generate a new entity in a bundle
generate:proxies	Generates proxy classes
generate:repositories	Generate repository classes
mapping:import	Import mapping info from database
mapping:info	Show basic info about mapped entities
schema:create	Processes the schema and create it or generate the SQL
schema:drop	Drop the database schema or generate the SQL
schema:update	Update the database schema or generate the SQL

Symfony Doctrine DBAL Configuration

```
doctrine:
  dbal:
    default_connection: default
    connections:
      default:
        dbname: database
        host: localhost
        port: 1234
        user: user
        password: secret
        driver: pdo_mysql
        driver_class: MyNamespace\MyDriverImpl
        options:
          foo: bar
        path: %kernel.data_dir%/data.sqlite
        memory: true
        unix_socket: /tmp/mysql.sock
        wrapper_class: MyDoctrineDbalConnectionWrapper
        charset: UTF-8
        logging: %kernel.debug%
        platform_service: MyOwnDatabasePlatformService
    conn1: # ...
  types:
```

Symfony Doctrine ORM Configuration

```
doctrine:
  orm:
    auto_generate_proxy_classes: true
    proxy_namespace: Proxies
    proxy_dir: %kernel.cache_dir%/doctrine/orm/Proxies
    default_entity_manager: default # Required
    entity_managers:
      default: # At least one has to be defined
        # The name of a DBAL connection (default used if not set)
        connection: conn1
        mappings: # Required
          HelloBundle: ~
        class_metadata_factory_name: Doctrine\ORM\Mapping\ClassMetadataFactory
        # Cache drivers have to be array, apc, xcache or memcache
        metadata_cache_driver: array
        query_cache_driver: array
        result_cache_driver:
          type: memcache
          host: localhost
          port: 11211
          instance_class: Memcache
        class: Doctrine\Common\Cache\MemcacheCache
    em2: # ...
```

Entity Repository (Method Doctrine\ORM\EntityRepository)

```
find($id) @return Entity
$em->getRepository('MyProject\Domain\User')->find($id);

findAll() @return Array
$em->getRepository('MyProject\Domain\User')->findAll();

findOneBy(array $criteria) @return Entity
$em->getRepository('MyProject\Domain\User')->findOneBy(array('nickname' => 'dude'));

findBy(array $criteria) @return Array
$em->getRepository('MyProject\Domain\User')->findBy(array('age' => 20));
```

Doctrine Query Langage

DQL FUNCTIONS

- ABS(arithmetic_expression)
- CONCAT(str1, str2)
- CURRENT_DATE()
- CURRENT_TIME()
- CURRENT_TIMESTAMP()
- LENGTH(str)
- LOCATE(needle, haystack [, offset])
- LOWER(str)
- MOD(a, b)
- SIZE(collection)
- SQRT(q)
- SUBSTRING(str, start [, length])
- TRIM(str)

AGGREGATE FUNCTIONS

- AVG
- COUNT
- MIN
- MAX
- SUM

HYDRATION MODES

- Query::HYDRATE_OBJECT or `getSingleResult()` / `getResult()`
- Query::HYDRATE_ARRAY or `getArrayResult()`
- Query::HYDRATE_SCALAR or `getScalarResult()`
- Query::HYDRATE_SINGLE_SCALAR or `getSingleScalarResult()`

OTHERS EXPRESSION

- ALL / ANY / SOME
- BETWEEN a AND b
- NOT BETWEEN a AND b
- IN(x1, x2, ...) / NOT IN (x1, x2, ...)
- LIKE a / NOT LIKE a
- IS NULL / IS NOT NULL
- EXISTS / NOT EXISTS
- INSTANCE OF

Examples

```
$em ->createQuery($query)
->setParameter(':p1', 'xx')
->setParameters(array(':p1' => 'xx', ':p2' => 'xy'))
->getResult(HYDRATE_MODE);
```

```
SELECT u FROM MyProject\Model\User u
SELECT DISTINCT u.id FROM Article a JOIN a.user u
SELECT a FROM Article a JOIN a.user u ORDER BY u.name ASC
SELECT u.username, u.name FROM User u
SELECT u, a FROM ForumUser u JOIN u.avatar a
SELECT u, p FROM User u JOIN u.phonenumbers p
SELECT u FROM ForumUser u ORDER BY u.id ASC
SELECT COUNT(u.id) FROM Entities\User u
SELECT u FROM ForumUser u WHERE u.id = ?1
SELECT u FROM ForumUser u WHERE (u.name = :name OR u.name = :name2) AND u.id = :id
SELECT u FROM User u WHERE ((u.id + 5000) * u.id + 3) < 10000000
SELECT u.id, a.id as article_id FROM User u LEFT JOIN u.articles a
SELECT u, a, p, c FROM User u JOIN u.articles a JOIN u.phonenumbers p JOIN a.comments c
SELECT u.name FROM User u WHERE u.id BETWEEN ?1 AND ?2
SELECT u FROM User u WHERE u.id IN (1, 2)
SELECT CONCAT(u.id, u.name) FROM User u WHERE u.id = ?1
SELECT u FROM User u WHERE EXISTS (SELECT p.phone FROM Phone p WHERE p.user = u.id)
SELECT u.id FROM User u WHERE :groupId MEMBER OF u.groups
SELECT u FROM User u WHERE SIZE(u.phonenumbers) > 1
SELECT u FROM User u WHERE u.phonenumbers IS EMPTY
SELECT u FROM Model\CompanyPerson u WHERE u INSTANCE OF Model\CompanyEmployee
```