





(/github/soliantconsulting)

soliantconsulting/zf-apigility-doctrine 

(<https://github.com/soliantconsulting/zf-apigility-doctrine>) &  (/github/zfcampus/zf-apigility-doctrine)

Forked from zfcampus/zf-apigility-doctrine (/github/zfcampus/zf-apigility-doctrine)

A Doctrine Apigility Admin and Server

Language: PHP (/languages/PHP)

Apigility for Doctrine

build **unknown** (<http://travis-ci.org/soliantconsulting/zf-apigility-doctrine>)

This module provides the classes for integrating Doctrine with Apigility.

Installation

Installation of this module uses composer. For composer documentation, please refer to getcomposer.org (<http://getcomposer.org/>).

```
$ php composer.phar require zfcampus/zf-apigility-doctrine:dev-master
```

This library provides two modules. The first, ZF\Apigility\Doctrine\Server provides the classes to serve data created by the second, ZF\Apigility\Doctrine\Admin. The Admin module is used to create apigility resources and the Server serves those created resources. Generally you would include Admin in your `development.config.php` and Server in your `application.config.php`

API Resources

/apigility/api/doctrine[/:object_manager_alias]/metadata[/:name]

This will return metadata for the named entity which is a member of the named object manager. Querying without a name will return all metadata for the object manager.

/apigility/api/module[/:name]/doctrine[/:controller_service_name]

This is a Doctrine resource creation route like Apigility Rest

/apigility/api/module[:name]/rest[:controller_service_name] POST Parameters:

```
{
  "objectManager": "doctrine.entitymanager.orm_default",
  "resourceName": "Artist",
  "entityClass": "Db\\Entity\\Artist",
  "routeIdentifierName": "artist_id",
  "entityIdentifierName": "id",
  "routeMatch": "/api/artist",
  "pageSizeParam": "limit", // optional default null
  "hydratorName": "DbApi\\V1\\Rest\\Artist\\ArtistHydrator", // Optional default generator
  "hydrateByValue": true // Optional default true
}
```

Hydrating Entities by Value or Reference

By default the admin tool hydrates entities by reference by setting `$config['zf-rest-doctrine-hydrator']['hydrator_class']['by_value']` to false.

Collections

The API created with this library implements full featured and paginated collection resources. A collection is returned from a GET call to an entity endpoint without specifying the id. e.g. `GET /api/resource`

Reserved GET variables

```
orderBy
query
```

Sorting Collections

Sort by columnOne ascending

```
/api/user_data?orderBy%5BcolumnOne%5D=ASC
```

Sort by columnOne ascending then columnTwo decending

```
/api/user_data?orderBy%5BcolumnOne%5D=ASC&orderBy%5BcolumnTwo%5D=DESC
```

Querying Collections

Queries are not simple key=value pairs. The query parameter is a key-less array of query definitions. Each query definition is an array and the array values vary for each query type.

Each query type requires at a minimum a 'type' and a 'field'. Each query may also specify a 'where' which can be either 'and' or 'or'. Embedded logic such as and(x or y) is not supported.

Building HTTP GET query with PHP. Use this to help build your queries.

PHP Example

```
echo http_build_query(
    array(
        'query' => array(
            array('field' => 'cycle', 'where' => 'and', 'type' => 'between', 'from' => '1', 'to' => '100'),
            array('field' => 'cycle', 'where' => 'and', 'type' => 'decimation', 'value' => '1000'),
        ),
        'orderBy' => array('columnOne' => 'ASC', 'columnTwo' => 'DESC')
    )
);
```

Javascript Example

```
$(function() {
    $.ajax({
        url: "http://localhost:8081/api/db/entity/user_data",
        type: "GET",
        data: {
            'query': [
                {
                    'field': 'cycle',
                    'where': 'or',
                    'type': 'between',
                    'from': '1',
                    'to': '100'
                },
                {
                    'field': 'cycle',
                    'where': 'or',
                    'type': 'gte',
                    'value': '1000'
                }
            ]
        },
        dataType: "json"
    });
});
```

Querying Relations

It is possible to query collections by relations - just supply the relation name as `fieldName` and identifier as `value`.

1. Using an RPC created by this module for each collection on each resource:

/resource/id/childresource/child_id

2. Assuming we have defined 2 entities, `User` and `UserGroup` ...

```
/**
 * @Entity
 */
class User {
    /**
     * @ManyToOne(targetEntity="UserGroup")
     * @var UserGroup
     */
    protected $group;
}
```

```
/**
 * @Entity
 */
class UserGroup {}
```

... we can find all users that belong to UserGroup id #1 with the following query:

```
$url = 'http://localhost:8081/api/user';
$query = http_build_query(array(
    array('type' => 'eq', 'field' => 'group', 'value' => '1')
));
```

Format of Date Fields

When a date field is involved in a query you may specify the format of the date using PHP date formatting options. The default date format is `Y-m-d H:i:s`. If you have a date field which is just `Y-m-d` then add the format to the query.

```
'format' => 'Y-m-d',
'value' => '2014-02-04',
```

Available Query Types

ORM and ODM

Equals

```
array('type' => 'eq', 'field' => 'fieldName', 'value' => 'matchValue')
```

Not Equals

```
array('type' => 'neq', 'field' => 'fieldName', 'value' => 'matchValue')
```

Less Than

```
array('type' => 'lt', 'field' => 'fieldName', 'value' => 'matchValue')
```

Less Than or Equals

```
array('type' => 'lte', 'field' => 'fieldName', 'value' => 'matchValue')
```

Greater Than

```
array('type' => 'gt', 'field' => 'fieldName', 'value' => 'matchValue')
```

Greater Than or Equals

```
array('type' => 'gte', 'field' => 'fieldName', 'value' => 'matchValue')
```

Is Null

```
array('type' => 'isnull', 'field' => 'fieldName')
```

Is Not Null

```
array('type' => 'isnotnull', 'field' => 'fieldName')
```

Dates in the In and NotIn filters are not handled as dates. It is recommended you use multiple Equals statements instead of these filters.

In

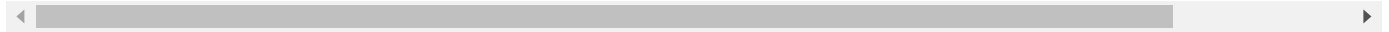
```
array('type' => 'in', 'field' => 'fieldName', 'values' => array(1, 2, 3))
```

NotIn

```
array('type' => 'notin', 'field' => 'fieldName', 'values' => array(1, 2, 3))
```

Between

```
array('type' => 'between', 'field' => 'fieldName', 'from' => 'startValue', 'to' =>
```



Like (% is used as a wildcard)

```
array('type' => 'like', 'field' => 'fieldName', 'value' => 'like%search')
```

ODM Only

Regex

```
array('type' => 'regex', 'field' => 'fieldName', 'value' => '/.*search.*/i')
```

Dependencies

composer.json (https://github.com/soliantconsulting/zf-apigility-doctrine/blob/master/composer.json)					
	Kind	Requirements	Latest Stable	Latest Release	Licenses
doctrine/doctrine-mongo-odm... (/packagist/doctrine%2Fdoctrine-mongo-odm-module)	development	*	1.1.0	1.1.0	MIT (/licenses/MIT)
doctrine/doctrine-orm-module (/packagist/doctrine%2Fdoctrine-orm-module)	development	*	2.1.4	2.1.4	MIT (/licenses/MIT)
doctrine/mongodb-odm (/packagist/doctrine%2Fmongodb-odm)	development	1.0.0-beta9@dev	1.2.7	2.0.0-beta1	MIT (/licenses/MIT)
php (/packagist/php)	runtime	>=5.4			
phpunit/phpunit (/packagist/phpunit%2Fphpunit)	development	3.7.*	8.1.2	8.1.2	BSD-3-Clause (/licenses/BSD-3-Clause)
zfcampus/zf-apigility (/packagist/zfcampus%2Fzf-apigility)	runtime	dev-master	1.4.0	1.4.0	BSD-3-Clause (/licenses/BSD-3-Clause)
zfcampus/zf-apigility-admin (/packagist/zfcampus%2Fzf-apigility-admin)	development	dev-master	1.7.0	1.7.0	BSD-3-Clause (/licenses/BSD-3-Clause)
zfcampus/zf-configuration (/packagist/zfcampus%2Fzf-configuration)	runtime	dev-master	1.3.3	1.3.3	BSD-3-Clause (/licenses/BSD-3-Clause)

Project Statistics

Sourcerank	0 (/github/soliantconsulting/zf-apigility-doctrine/sourcerank)
Repository Size	557 KB
Stars	0 (https://github.com/soliantconsulting/zf-apigility-doctrine/stargazers)
Forks	0 (https://github.com/soliantconsulting/zf-apigility-doctrine/network)
Watchers	3 (https://github.com/soliantconsulting/zf-apigility-doctrine/watchers)
Open issues	disabled
Dependencies	8
Contributors	10 (/github/soliantconsulting/zf-apigility-doctrine/contributors)
Tags	0 (/github/soliantconsulting/zf-apigility-doctrine/tags)
Created	Mar 29, 2014
Last updated	Mar 29, 2014
Last pushed	Mar 28, 2014

Top Contributors [See all \(/github/soliantconsulting/zf-apigility-doctrine/contributors\)](https://github.com/soliantconsulting/zf-apigility-doctrine/contributors)



🔔 Something wrong with this page? Make a suggestion (<https://github.com/librariesio/libraries.io/issues/new>)

Login ([/login](#)) to resync this repository

Libraries.io (/) helps you find new open source packages, modules and frameworks and keep track of ones you depend upon.

🐦 (<https://twitter.com/librariesio>) 🌐 (<https://github.com/librariesio>) 📘 (<https://www.facebook.com/libraries.io>)

✉ (<mailto:support@libraries.io>) 🗯 (<https://slack.libraries.io/>) 🔔

(<https://github.com/librariesio/libraries.io/issues/new>)

(<https://tidelift.com>)

Copyright © 2019 Tidelift, Inc

Code is Open Source under AGPLv3 (<https://github.com/librariesio/libraries.io/blob/master/LICENSE.txt>)
license

Data is available under CC-BY-SA 4.0 (<https://libraries.io/data>) license

Explore (/explore)

- [Platforms \(/platforms\)](#)
- [Languages \(/languages\)](#)
- [Licenses \(/licenses\)](#)
- [Help Wanted \(/explore/help-wanted\)](#)
- [First Pull Request \(/explore/first-pull-request\)](#)
- [Unlicensed Packages \(/explore/unlicensed-libraries\)](#)
- [Trending Repositories \(/github/trending\)](#)

Experiments (/experiments)

- [Bus Factor \(/experiments/bus-factor\)](#)
- [Digital Infrastructure \(/experiments/digital-infrastructure\)](#)
- [Unseen Infrastructure \(/experiments/unseen-infrastructure\)](#)

About (/about)

Team (/team)

Terms and Conditions (/terms)

Privacy Policy (/privacy)

API (/api)