

[READ-ONLY] A flexible, lightweight and powerful Object-Relational Mapper for PHP, implemented using the DataMapper pattern. This repo is a split of the main code that can be found in <https://github.com/cakephp/cakephp>

1,822 commits

7 branches

105 releases

82 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

markstory

Fix invalid SQL generation from leftJoinWith() & auto-fields

Latest commit 76de1e0 16 days ago

Association	Fix few errors reported by phpstan level 3	2 months ago
Behavior	Fix notice errors when scope is unset.	a month ago
Exception	Fixing style errors.	20 days ago
Locator	Merge branch 'master' into 3.next	8 months ago
Rule	Fix few errors reported by phpstan level 3	2 months ago
Association.php	Fix invalid SQL generation from leftJoinWith() & auto-fields	16 days ago
AssociationCollection.php	Update version in deprecated tag.	5 months ago
AssociationsNormalizerTrait.php	Use HTTPS for the opensource.org URL	8 months ago
Behavior.php	Improve doc blocks for IDE understanding.	5 months ago
BehaviorRegistry.php	Make BehaviorRegistry::className() return null on failure.	2 months ago
EagerLoadable.php	Fix errors reported by phpstan.	3 months ago
EagerLoader.php	Use interface const instead of magic strings for LEFT/RIGHT/INNER joi...	5 months ago
Entity.php	Use HTTPS for the opensource.org URL	8 months ago
LICENSE.txt	Use HTTPS for the cakefoundation.org URL	8 months ago
LazyEagerLoader.php	Merge pull request #10815 from albertcansado/deprecated-dirty	8 months ago
Marshaller.php	Fix incorrect validator object checking.	2 months ago
PropertyMarshalInterface.php	Use HTTPS for the opensource.org URL	8 months ago
Query.php	Fix errors reported by phpstan.	3 months ago
README.md	Add meta data cache doc for ORM split.	a month ago
ResultSet.php	Fix more doc blocks.	3 months ago
RulesChecker.php	Use HTTPS for the opensource.org URL	8 months ago
SaveOptionsBuilder.php	Merge branch 'master' into 3.next	8 months ago
Table.php	Fix few errors reported by phpstan level 3	2 months ago
TableRegistry.php	Fix errors reported by phpstan.	3 months ago
composer.json	Use caret operator in composer.json for all sub packages	9 months ago

README.md

downloads 27k license MIT

CakePHP ORM

The CakePHP ORM provides a powerful and flexible way to work with relational databases. Using a datamapper pattern the ORM allows you to manipulate data as entities allowing you to create expressive domain layers in your applications.

Database engines supported

The CakePHP ORM is compatible with:

- MySQL 5.1+
- Postgres 8+
- SQLite3
- SQLServer 2008+
- Oracle (through a [community plugin](#))

Connecting to the Database

The first thing you need to do when using this library is register a connection object. Before performing any operations with the connection, you need to specify a driver to use:

```
use Cake\Datasource\ConnectionManager;

ConnectionManager::setConfig('default', [
    'className' => 'Cake\Database\Connection',
    'driver' => 'Cake\Database\Driver\Mysql',
    'database' => 'test',
    'username' => 'root',
    'password' => 'secret',
    'cacheMetadata' => true,
    'quoteIdentifiers' => false,
]);
```

Once a 'default' connection is registered, it will be used by all the Table mappers if no explicit connection is defined.

Creating Associations

In your table classes you can define the relations between your tables. CakePHP's ORM supports 4 association types out of the box:

- belongsTo - E.g. Many articles belong to a user.
- hasOne - E.g. A user has one profile
- hasMany - E.g. A user has many articles
- belongsToMany - E.g. An article belongsToMany tags.

You define associations in your table's `initialize()` method. See the [documentation](#) for complete examples.

Reading Data

Once you've defined some table classes you can read existing data in your tables:

```
use Cake\ORM\TableRegistry;

$articles = TableRegistry::get('Articles');
foreach ($articles->find() as $article) {
    echo $article->title;
}
```

You can use the [query builder](#) to create complex queries, and a [variety of methods](#) to access your data.

Saving Data

Table objects provide ways to convert request data into entities, and then persist those entities to the database:

```
use Cake\ORM\TableRegistry;

$data = [
    'title' => 'My first article',
    'body' => 'It is a great article',
    'user_id' => 1,
    'tags' => [
        '_ids' => [1, 2, 3]
    ],
    'comments' => [
```

```

        ['comment' => 'Good job'],
        ['comment' => 'Awesome work'],
    ]
];

$articles = TableRegistry::get('Articles');
$article = $articles->newEntity($data, [
    'associated' => ['Tags', 'Comments']
]);
$articles->save($article, [
    'associated' => ['Tags', 'Comments']
])

```

The above shows how you can easily marshal and save an entity and its associations in a simple & powerful way. Consult the [ORM documentation](#) for more in-depth examples.

Deleting Data

Once you have a reference to an entity, you can use it to delete data:

```

$articles = TableRegistry::get('Articles');
$article = $articles->get(2);
$articles->delete($article);

```

Meta Data Cache

It is recommended to enable meta data cache for production systems to avoid performance issues. For e.g. file system strategy your bootstrap file could look like this:

```

use Cake\Cache\Engine\FileEngine;

$cacheConfig = [
    'className' => FileEngine::class,
    'duration' => '+1 year',
    'serialize' => true,
    'prefix' => 'orm_',
],
Cake::setConfig('_cake_model_', $cacheConfig);

```

Additional Documentation

Consult [the CakePHP ORM documentation](#) for more in-depth documentation.