->where(['\_id' => $id]) // implicit typecast to \MongoId ->one();

To get the actual Mongo ID string, you should typecast the \Mongoid instance to a string:

$query = new Query;

$row = $query->from('customer')->one();

var\_dump($row['\_id']); // outputs: "object(MongoId)"var\_dump((string)$row['\_id']);

How it works...

The Query, ActiveQuery, and ActiveRecord classes of this extension extends yii\db\QueryInterface and yii\db\BaseActiveRecord, and therefore they are compatible with the built-in framework Query, ActiveQuery, and ActiveRecord classes.

You can use the yii\mongodb\ActiveRecord class for your models and the yii\mongodb\ActiveQuery builder to retrieve your models and use them in your data provider:

use yii\data\ActiveDataProvider; use app\models\Customer;

$provider = new ActiveDataProvider([

'query' => Customer::find(),

'pagination' => [

'pageSize' => 10,

]

]);

For general information on how to use Yii’s ActiveRecord, please refer to the Chapter 3, ActiveRecord, Model, and Database.

See also

* For more information about the extension, refer to the following URLs:

° [https://github.com/yiisoft/yri2-mongodb/blob/master/docs/gnide/RFADME.md](https://github.com/yiisoft/yii2-mongodb/blob/master/docs/guide/README.md) ° <http://www.yiiframework.com/doc-2.0/ext-mongodb-index.html>

* And for information about the original library, refer to:

o [httpsy/docs.mongodb.org/mannal/](https://docs.mongodb.org/manual/)

* For ActiveRecord usage refer to the Chapter 3, ActiveRecord, Model, and Database