**Cake Document**

**What is Cake php ...?**

CakePHP is an open-source web, rapid development framework that makes building web applications simpler, faster and require less code.

**Understanding Model-View-Controller**

**1. Model layer**

The Model layer represents the part of your application that implements the business logic

**2. View layer**

The View renders a presentation of modeled data.

**3. Controller layer**

The Controller layer handles requests from users. It is responsible for rendering a response with the aid of both the Model and the View layer.

Link => https://book.cakephp.org/2.0/en/cakephp-overview/understanding-model-view-controller.html

Link => https://book.cakephp.org/3.0/en/installation.html

**Before starting Check PHP version:**

1. Command :- php -v

2. Required Version :- PHP 5.6.0 (CLI) or higher

**Installation**

1. Using Composer.

2. Simpley dowenload from site

After dowenload Make sure Permission folder (**temp/logs**).

**Database Configuration**

1. config/app.php

**Generate a basic application**

**Before u start First u need config database name in config file.**

First yo need to go through your project path.

1. bin/cake bake all users

This will generate the controllers, models, views, their corresponding test cases.

**Note :-** Database tables must required before Cakeback.

**OR**

Create all files by launching bake commands:

1. bin/cake bake model Categories

2. bin/cake bake controller Categories

3. bin/cake bake template Categories

bin/cake bake all companies

**Baisc Notes**

1. Naming For Database Like => **usres,roles** etc.....

Foreign keys in **hasMany**, **belongsTo** or **hasOne** relationships

Link => https://book.cakephp.org/2.0/en/getting-started/cakephp-conventions.html#

2. Default fields that we need to add in table

1. id

2. forienkey if existes

3. status

4. is\_deleted

5. create

6. modified

3. Users tabel you need to mention two fields must required for authentication is (**api\_key , api\_key\_plain**).

**Authentication For Api side.**

First u need to defined type of your api extension in **config/routes.php**

Like => $routes->extensions(['**json**','**xml**']);

**Types of Authentication.**

**1. Form Authentication.**

=>Allows you to authenticate users based on form POST data. Usually, this is a login form that users enter information into.

**2. Basic Authentication.**

=> Allows you to authenticate users using Basic HTTP authentication. ( authenticate itself with a **api\_key** and **api\_key\_plain**).

=> You don’t require a login action in your controller.

Link => <https://book.cakephp.org/3.0/en/controllers/components/authentication.html>

You can configure authentication handlers in your controller’s **beforeFilter()** or **initialize()** methods.

**Initialize() =>** You can load your components.

**BeforeFilter() =>** You can identify the function which will get without authenticate.

Like => login , add , uploadimage.

**Example => $this->Auth->allow(['login','add','upload']);**

**CAKE QUERIES**

Making condition in json

// $passData = array("conditions" => array("status" => 1,"OR" => array(array("inviter" => 1,"invitee" => 5),array("invitee" => 1,"inviter" => 5))));

// echo json\_encode($passData);exit;

1) http://192.168.0.128/i\_gotcha/users/index.json

**Request:**

{ "get":"all"}

“**get parameter is MUST**”

**Response:**

{

"code": 200,

"url": "users/index.json",

"message": "Record(s) found",

"users": [

{

"id": 10,

"fname": "111",

"lname": "444",

"created": "2017-03-15T12:29:33",

...

},

{

"id": 11,

"fname": "333",

"lname": "444",

"created": "2017-03-15T12:29:33",

...

},

...

}

2) AND

http://192.168.0.128/i\_gotcha/users.json

**Request:**

{"conditions":{"fname":"test", "lname":"query"}, "get":"all"}

**sql** => where fname = 'test' AND lname = 'query'

**Response:**

{

"code": 200,

"url": "users/index.json",

"message": "Record(s) found",

"users": [

{

"id": 10,

"fname": "test",

"lname": "query",

"created": "2017-03-15T12:29:33",

...

},

{

"id": 11,

"fname": "test",

"lname": "query",

"created": "2017-03-15T12:29:33",

...

},

...

}

3) OR

http://192.168.0.128/i\_gotcha/users.json

**Request:**

{"conditions":{"OR":{"fname":"Mike", "lname":"West"}}, "get":"all"}

**sql** => where fname = 'Mike' OR lname = 'West'

**Response:**

{

"code": 200,

"url": "users/index.json",

"message": "Record(s) found",

"users": [

{

"id": 10,

"fname": "Mike",

"lname": "Allen",

"created": "2017-03-15T12:29:33",

...

},

{

"id": 11,

"fname": "John",

"lname": "West",

"created": "2017-03-15T12:29:33",

...

},

...

}

4) AND & OR

http://192.168.0.128/i\_gotcha/users.json

**Request:**

{"conditions":{"country":”India”, "OR":{"fname":"Mike", "lname":"West"}}, "get":"all"}

**sql** => where country ='India' AND (fname = 'Mike' OR lname = 'West')

**Response:**

{

"code": 200,

"url": "users/index.json",

"message": "Record(s) found",

"users": [

{

"id": 10,

"fname": "Mike",

"lname": "Allen",

"country": "India",

"created": "2017-03-15T12:29:33",

...

},

{

"id": 11,

"fname": "John",

"lname": "West",

"country": "India",

"created": "2017-03-15T12:29:33",

...

},

...

}

{"conditions":{"country":”India”, "OR":{"fname":"Mike"}}, "get":"all"}

**sql** => where country ='India' OR fname = 'Mike'

5) IN Clause

http://192.168.0.128/i\_gotcha/users.json

**Request:**

{"conditions":{"id IN":[10, 11]}, "get":"all"}

**sql** => where id IN (10, 11)

**Response:**

{

"code": 200,

"url": "users/index.json",

"message": "Record(s) found",

"users": [

{

"id": 10,

"fname": "Mike",

"lname": "Allen",

"country": "India",

"created": "2017-03-15T12:29:33",

...

},

{

"id": 11,

"fname": "John",

"lname": "West",

"country": "India",

"created": "2017-03-15T12:29:33",

...

}

}

{"conditions":{"country":"India", "id IN":[10, 11]}, "get":"all"}

**sql** => where country = 'India' AND id IN (10, 11)

{"conditions":{"OR":{"country":"India", "id IN":[10, 11]}}, "get":"all"}

**sql** => where country = 'India' OR id IN (10, 11)

6) LIKE Clause

http://192.168.0.128/i\_gotcha/users.json

**Request:**

{"conditions":{"fname LIKE" : "%mike%"}, "get":"all"}

**sql** => where fname LIKE '%mike%'

**Response:**

{

"code": 200,

"url": "users/index.json",

"message": "Record(s) found",

"users": [

{

"id": 10,

"fname": "Mike",

"lname": "Allen",

"country": "India",

"created": "2017-03-15T12:29:33",

...

},

{

"id": 11,

"fname": "Mikey",

"lname": "o'conner",

"country": "India",

"created": "2017-03-15T12:29:33",

...

}

}

{"conditions":{"fname NOT LIKE" : "%mike%"}, "get":"all"}

**sql** => where fname NOT LIKE '%mike%'

7) OPERATOR

http://192.168.0.128/i\_gotcha/users.json

**Operators:**  < , >, <=, >=, <>, !=

**Request:**

{"conditions":{"id >":10, "id <":15}, "get":"all"}

**sql** => where id > 10 AND id < 15

**Request:**

{"conditions":{"id >=":10, "id <=":15}, "get":"all"}

**sql** => where id >= 10 AND id <= 15

**\* Request:**

{"conditions":{"id <>" : 10}, "get":"all"}

**sql** => where id <> 10

**\* Request:**

{"conditions":{"id != " : 10}, "get":"all"}

**sql** => where id != 10

\* Above both Queries are the same but first one with “<>” is recommended as it's sql standard .

8) FIELD parameters

http://192.168.0.128/i\_gotcha/users.json

**Request:**

{"fields":["Users.id", ".Users.fname", "Users.lname"], "get":"all"}

**sql** => SELECT id, fname, lname FROM users

**Request:**

{"fields":["DISTINCT Users.fname"], "get":"all"}

**sql** => SELECT DISTINCT fname FROM users

9) CONTAIN parameters

http://192.168.0.128/i\_gotcha/users.json

**Request:**

{"contain":["Feeds"], "get":"all"}

**Response:**

{

"code": 200,

"url": "users/index.json",

"message": "Record(s) found",

"users": [

{

"id": 10,

"fname": "Mike",

"lname": "Allen",

"country": "India",

"feeds": [

{

"id": 60,

"user\_id": 10,

"place\_name": "test"

},

{

"id": 61,

"user\_id": 10,

"place\_name": "test",

}

]

},

{

"id": 11,

"fname": "Mikey",

"lname": "o'conner",

"country": "India",

}

}

10) ORDER parameters

http://192.168.0.128/i\_gotcha/users.json

**Request:**

{"order":{"fname":"ASC"}, "get":"all"}

**Request:**

{"conditions":{"status":"1"}, "order":{"fname":"ASC", "lname":"DESC"}, "get":"all"}

11) GROUP parameters

http://192.168.0.128/i\_gotcha/users.json

**Request:**

{"group":["country"], "get":"all"}

**Request:**

{"conditions":{"status":"1"}, "group":["country", "fname"], "get":"all"}

11) LIMIT , OFFSET parameters

http://192.168.0.128/i\_gotcha/users.json

**Request:**

{ "limit":3, "offset":0, "get":"all"}

**Request:**

{"conditions":{"status":"1"}, "limit":3, "offset":3, "get":"all"}

=> LIMIT , Page parameters

**Request:**

{ "limit":3, "page":1, "get":"all"}

**Request:**

{"conditions":{"status":"1"}, "limit":3, "offset":3, "get":"all"}

12) Aggregate function parameters

$params = array(

'conditions' => ['product\_id'=>$product\_id,'rate != '=>''],

'contain' => [],

'fields' => array('rate'=>'avg(rate)'),

'select' => array()

);

$get\_avg=$this->Addevents->get\_avg($params);

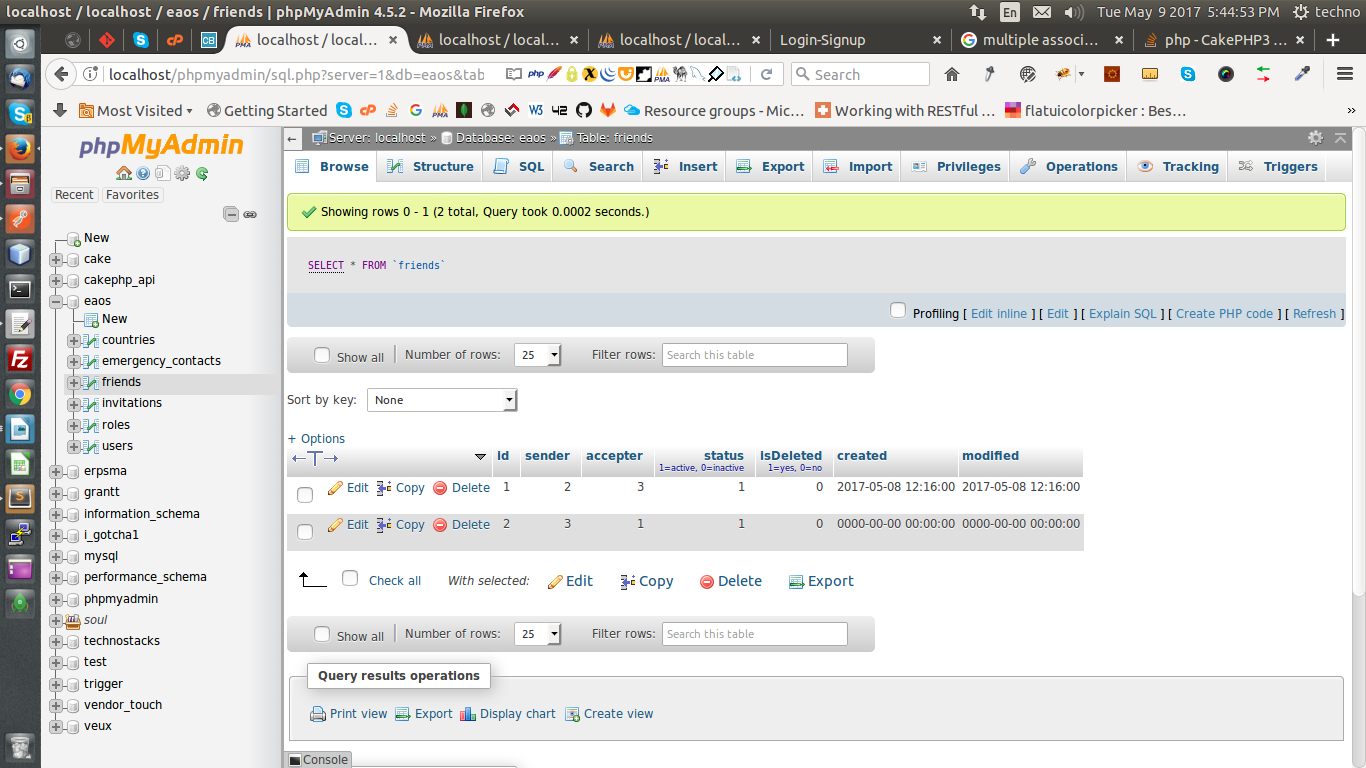
**$avg=AddeventsTable::find('all',$params)->first();**

json query for count with condition & groupby

{"conditions":{"video\_status":"1"}, "group":["user\_id"],"fields":{"video\_status":"count(video\_status)","user\_id":"user\_id"},"get":"all"}

Content fields & limit & order

Request : {"contain":["Feeds"], "fields":[],"order":{"Media.id":"DESC"},"limit":1, "get":"all"}



For making multiple foreign keys in one table which are same pK of one table

write in FriendsTable.php in initialize function

here make aliase

$this->belongsTo('Accepteruser', [

'foreignKey' => 'accepter',

'joinType' => 'INNER',

'className' => 'Users'

]);

$this->belongsTo('Senderuser', [

'foreignKey' => 'sender',

'joinType' => 'INNER',

'className' => 'Users'

]);

--> and in UserTable.php

$this->hasMany('Friends', [

'foreignKey' => 'accepter'

]);

$this->hasMany('Friends', [

'foreignKey' => 'sender'

]);

screen: Chat - Main

name : render list of our frnds

url : http://192.168.0.133/eaos\_api/Friends.json

request:

{

"conditions":{"Friends.status":"1","Friends.isDeleted":"0",

"or": {

"Friends.accepter":"3", // here current use id write

"Friends.sender":"3" // here current use id write

}

},

"contain" : ["Accepteruser","Senderuser"],

"get":"all"

}

response:

{

"code": 200,

"url": "Friends/index.json",

"message": "Record found",

"Friends": [

{

"id": 1,

"sender": 2,

"accepter": 3,

"status": true,

"isDeleted": false,

"created": "2017-05-08T12:16:00",

"modified": "2017-05-08T12:16:00",

"senderuser": {

"id": 2,

"role\_id": 2,

"email": "access@test.com",

"firstname": "mark",

"lastname": "hydenn",

"contactno": "9876543211",

"city": "Ahmedabad",

"country\_id": 2,

"profile\_img": null,

"api\_key": "$2y$10$ZexlJ2f58yOmKSQhueJxc.g/P.5m40vAwSPQTz2VzW6oJC4pnkB0K",

"api\_key\_plain": "24cf57066414a6724df825595bf493c423aa2bc9",

"status": true,

"isDeleted": false,

"created": "2017-05-06T07:19:52",

"modified": "2017-05-06T07:19:52"

},

"accepteruser": {

"id": 3,

"role\_id": 2,

"email": "techno@test.com",

"firstname": "amit",

"lastname": "mehta",

"contactno": "9876543212",

"city": "baroda",

"country\_id": 2,

"profile\_img": null,

"api\_key": "$2y$10$b6xywoRP1.JFfwhfECZBtuYlzdJ5/39gpoytXgS5t7VfQzdlMkdU6",

"api\_key\_plain": "6b9cf23ab534c9da9893965062e8e9566a00824c",

"status": true,

"isDeleted": false,

"created": "2017-05-06T07:21:32",

"modified": "2017-05-06T07:21:32"

}

},

{

"id": 2,

"sender": 3,

"accepter": 1,

"status": true,

"isDeleted": false,

"created": null,

"modified": null,

"senderuser": {

"id": 3,

"role\_id": 2,

"email": "techno@test.com",

"firstname": "amit",

"lastname": "mehta",

"contactno": "9876543212",

"city": "baroda",

"country\_id": 2,

"profile\_img": null,

"api\_key": "$2y$10$b6xywoRP1.JFfwhfECZBtuYlzdJ5/39gpoytXgS5t7VfQzdlMkdU6",

"api\_key\_plain": "6b9cf23ab534c9da9893965062e8e9566a00824c",

"status": true,

"isDeleted": false,

"created": "2017-05-06T07:21:32",

"modified": "2017-05-06T07:21:32"

},

"accepteruser": {

"id": 1,

"role\_id": 2,

"email": "tester@test.com",

"firstname": "Maxwell",

"lastname": "Austin",

"contactno": "9876543210",

"city": "surat",

"country\_id": 2,

"profile\_img": "1494063108\_525157.jpeg",

"api\_key": "$2y$10$qQp0EKo7Ps5d4ZRb8uPKKO.XWF9gSOgwpJdISLvgsXcpbrz0Z7Owa",

"api\_key\_plain": "c6a79acc8671aae02afc788b9d23ed94934b4e9e",

"status": true,

"isDeleted": false,

"created": "2017-05-06T07:16:27",

"modified": "2017-05-06T11:39:16"

}

}

]

}

FK Relation Based Multiple joints

{ "get":"all","contain": ["Cart.Products","Cart.Products.Media","Cart"],"conditions":{"Users.id": 5}}

NOT IN

{  
"conditions": {  
"id NOT IN":[5,6], // here in IN pass current login user id as well as users 's friends 's ids  
"or": {  
"contactno":"9876543210",  
"email":"access@test.com"  
}  
},"get":"all"  
}

(AND) OR (AND)

{"conditions":{

"OR":[{"inviter":1,"invitee":5},{"invitee":1,"inviter":5}]

},

"status":0

}

sql: ("inviter":1 AND "invitee":5) OR (invitee":1,"inviter":5)

update on other fields

$requestData = $this->request->data;

//pr($requestData['status']); exit();

$conditions = $requestData['conditions'];

$fields = array('status'=>$requestData['status'],””=>””);

if ($this->Invitations->updateAll($fields, $conditions)) {

$this->serialize(200, 'Data is Edited');

}else{

$this->serialize(400, 'Data is not Edited');

}