



# CakePHP 4 Insert, Save, Update (hasOne) Associative Data



Asyraf Wahi Anuar - January 27, 2021

<https://codethepixel.com/CakePHP-4-Insert-Save-Update-hasOne-Associative-Data>

---

Insert, save and edit the associative data is a common process in web applications. Most of the related associative data is based on the 'hasMany' relationship. However, in this tutorial, the focus is on the 'hasOne' relationship e.g user has one profile. Technically, some of the developers combined the user profile information into the same user table. However, if the user profile comprises lots of data, it is recommended to separate the storage to make it more systematic. The following tutorial will guide you on how to insert, save and edit the associated data.

There are 2 tables used as a sample for this tutorial as shown below:

**Table: users**

**Table: profiles**

Relationship: User has one profile

## Define Relationship

First, define the relationship and ensure that it has a correct relationship.

File: UsersTable.php:

Location: .../src/Model/Table

```
public function initialize(array $config): void
{
    parent::initialize($config);

    //...others relationship
    $this->hasOne('Profiles'); //User has one profile. If table name: user_pro
}
```

File: ProfilesTable.php:

Location: .../src/Model/Table

```
public function initialize(array $config): void
```

```

{
    parent::initialize($config);

    //...others relationship
    $this->belongsTo('Users', [
        'foreignKey' => 'user_id',
        'joinType' => 'INNER',
    ]);
}

```

## Associated Input Field

The visible input in the add function is username, password, email, status. To include the profile input (fullname, dob and gender) into user add page, simply add the input field as follows:

File: add.php

Location:../templates/Users

```

<?php
    echo $this->Form->control('username');
    echo $this->Form->control('password');
    echo $this->Form->control('email');
    echo $this->Form->control('status');
    echo $this->Form->control('profile.fullname'); //If table name: user_profile
    echo $this->Form->control('profile.dob');
    echo $this->Form->control('profile.gender');
?>

```

## Saving Associated Data

Then to save the associated data to the Profiles table, add the following codes into UsersController.

File: UsersController.php

Location:../src/Controller

public function add()

```

{
    $user = $this->Users->newEmptyEntity();
    if ($this->request->is('post')) {
        $user = $this->Users->patchEntity($user, $this->request->getData());
        if ($this->Users->save($user, ['associated' => ['Profiles']])) { //Save
            $this->Flash->success(__('The user has been saved.'));

            return $this->redirect(['action' => 'index']);
        }
        $this->Flash->error(__('The user could not be saved. Please, try again.
    });
    $this->set(compact('user'));
}

```

## Set Accessible: Profile Table

File: User.php

Location:../src/Model/Entity

```
protected $_accessible = [  
    'username' => true,  
    'password' => true,  
    //others fields  
    'profile' => true, //profile fields. If table name: user_profiles used use  
];
```

## View Profile Information

To view the profile information, the user's view controller needs to get the user profile based on user\_id as shown below:

File: UsersController.php

Location:../src/Controller

```
public function view($id = null)  
{  
    $user = $this->Users->get($id, [  
        'contain' => ['Profiles'],  
    ]);  
  
    $this->set(compact('user'));  
}
```

In your user view page, call the related profile information that you required using the following codes:

File: view.php

Location:../templates/Users

```
<?= h($user->profile->fullname) ?> <!--if table user_profiles use user_profile
```

## Edit Associated Data

To edit the associative data from the user edit view, the public function edit in the users' controller needs to get the profiles information based on the user\_id, then -save the associated data as shown below:

File: UsersController.php

Location:../src/Controller

```
public function edit($id = null)  
{  
    $user = $this->Users->get($id, [  
        'contain' => ['Profiles'],  
    ]);  
  
    if ($this->request->is(['patch', 'post', 'put'])) {  
        $formdata = $this->getRequest()->getData();
```

```

        $user = $this->Users->patchEntity($user, $formdata,['associated'=>['Pr
        if ($this->Users->save($user)) {
            $this->Flash->success(__('The user has been saved.'));

            return $this->redirect(['action' => 'index']);
        }
        $this->Flash->error(__('The user could not be saved. Please, try again
    }

    $this->set(compact('user'));
}

```

The following codes are used to recall the profile value in edit input (same as add.php):

File: edit.php

Location:../templates/Users

```

echo $this->Form->control('profile.fullname'); //If table name: user_profiles
echo $this->Form->control('profile.dob');
echo $this->Form->control('profile.gender');

```

That all. Happy coding :)