
Phorms Documentation

Release 0.1.0

Josh Yudaken

August 17, 2016

1	Element interface	3
2	Indices and tables	5
3	Show me the code	7

Contents:

class Base

Base

Base::setDate (*\$year*, *\$month*, *\$day*)

Set the date.

Parameters

- **\$year** (*int*) – The year.
- **\$month** (*int*) – The month.
- **\$day** (*int*) – The day.

Returns Either false on failure, or the datetime object for method chaining.

Base::setTime (*\$hour*, *\$minute* [, *\$second*])

Set the time.

Parameters

- **\$hour** (*int*) – The hour
- **\$minute** (*int*) – The minute
- **\$second** (*int*) – The second

Returns Either false on failure, or the datetime object for method chaining.

constant Base::ATOM

Y-m-dTH:i:sP

Element interface

interface Element

The root class for any form element. This includes controls (input boxes, select, textarea), stack items (fieldset, actions) as well as containers (forms),

Element::setDate (*\$year*, *\$month*, *\$day*)
Set the date.

Parameters

- **\$year** (*int*) – The year.
- **\$month** (*int*) – The month.
- **\$day** (*int*) – The day.

Returns Either false on failure, or the datetime object for method chaining.

Element::setTime (*\$hour*, *\$minute* [, *\$second*])
Set the time.

Parameters

- **\$hour** (*int*) – The hour
- **\$minute** (*int*) – The minute
- **\$second** (*int*) – The second

Returns Either false on failure, or the datetime object for method chaining.

constant Element::ATOM
Y-m-dTH:i:sP

By convention CakePHP renders a view with an inflected version of the action name. Returning to our online bakery example, our RecipesController might contain the `view()`, `share()`, and `search()` actions. The controller would be found in `/app/Controller/RecipesController.php` and contain:

```
# /app/Controller/RecipesController.php

class RecipesController extends AppController {
    public function view($id) {
        //action logic goes here..
    }

    public function share($customerId, $recipeId) {
        //action logic goes here..
    }
}
```

```
public function search($query) {  
    //action logic goes here..  
}  
}
```

Indices and tables

- `genindex`
- `modindex`
- `search`

Show me the code

```
<?php

session_start();
Phorms\Csrf::setSecret('mysecret');

$form = new Phorms\Form([
    // Set up a fieldset for the fields
    '@fieldset:Your request',

    // Input boxes are the default, and are defined simply:
    'firstname' => 'Firstname',

    // Passing an array as the second option allows extra properties:
    'email' => ['Email', 'type'=>'email', 'required'=>True],

    // Some types are automatically detected, such as a select box:
    'topic' => ['Topic', array(
        'topics/barley.txt' => 'Barley',
        'topics/rice.txt' => 'Rice',
        'topics/wheat.txt' => 'Wheat',
    )],

    // Controls can also be provided as objects (simple challenge response)
    new Phorms\Element_Checkbox(array(
        'name' => 'human',
        'caption' => 'Are you human?',
        'required' => True,
    )),

    // Opening another fieldset will automatically close the previous one
    '@fieldset:Action(s)',
    '@submit:Notify me'
]);

if ($data = $form->data()) {
    // Form was submitted, but check if there were errors
    if ($errors = $form->check($data)) {
        // Inform the user of the errors (or just dump them for now)
        var_dump($errors);
    }else{
        // There were no errors, generate and send the email

        // This following line would usually be dangerous because you allow the
```

```
// user to specify the file path (he could choose any file in the system!)
// Luckily Phorms will validate that the submitted value was actually one
// of the options in the dropdown box.
$body = file_get_contents($data['topic']);

// Send out an email, the $data['email'] field was required and validated already
mail($data['email'], 'Hi ' . ($data['firstname'] ?: 'there'), $body);

// Choose to exit here, although a 303 redirect is recommended (Post-Redirect-Get pattern)
print 'We have sent you an email.';
exit(0);
}
}

// Render the form, if there were any errors $data includes the entries
$form->render($data);
```