



# Unittesting mit PHP

# Inhalt

- > Ziele
- > Einstieg in PHP
- > Unittesting mit Beispiel
- > Test-Auswertung

# Ziel

- > Dieses PHP ist wirklich so doof wie ich dachte
- > Wow, das geht mit PHP. Hätte ich nicht gedacht!

## > Variabeln

# PHP

```
1 <?php
2
3 $var = "";
4 $var = 8;
5 $var = new MyObject();
```

## > Vergleichen

# PHP

```
1 <?php
2
3 var_dump("1" == 1); //true
4 var_dump("1" == true); //true
5 var_dump(0 == false); //true
6 var_dump("" == false); //true
7 var_dump("0" == false); //true
8
9 var_dump("0" === false); //false
0 var_dump("0" !== false); //true
```

## > Typsicherheit

# PHP

```
1 <?php
2
3 function myFunc(MyObject $obj) {
4     //tu was
5 }
6
7 $var = new MyObject();
8 myFunc($var);
9
0 $var2 = new MyOtherObject();
1 //fehler
2 myFunc($var2);
```

## &gt; Klassen

## PHP

```
1 <?php
2 namespace fadal\web\mvc\actionresult {
3     use fadal\data\objectmodel;
4     use fadal\web\mvc;
5
6     /**
7      * Viewresultat, welches den Inhalt einer View evaluiert.
8      *
9      * @copyright 2010 nambu GmbH
10     * @package mvc
11     * @version 0.5
12     * @author dominik
13     */
14     class ViewActionResult implements ITemplateActionResult {
15
16         /**
17          * @var ClassBase
18          */
19         protected $model;
20
21         private $viewName;
22
23         /**
24          * @param mixed $model = null
25          * @param string $viewName = null
26          * @param string $controllerName = null
27          * @param array $assign = null
28          */
29         public function __construct($model = null, $viewName = null,
30             $this->assign = $assign;
31             $this->model = $model;
32             $this->viewName = $viewName;
33             $this->controllerName = $controllerName;
34     }
35 }
```



# Test

```

6 class AddressTest extends PHPUnit_Framework_TestCase {
7
8     public function testListeAction() {
9         $addresses = new Addresses();
10        $addresses->add(new Address(1, 1, "Musterstrasse 12", 6000, "Luzern"));
11        $addresses->add(new Address(2, 1, "Musterweg 21", 6000, "Luzern"));
12        $addresses->add(new Address(3, 1, "Muster 5", 3000, "Bern"));
13        $addrRepo = $this->getMock('IAddressRepository');
14        $addrRepo->expects($this->once())->method('getAddresses')->will($this->returnValue($addresses));
15        //
16        $controller = new AddressController($addrRepo);
17        $result = $controller->Liste();
18        //
19        $this->assertTrue($result instanceof ViewActionResult);
20        $this->assertTrue($controller->isValid());
21        $this->assertTrue($result->getModel() instanceof Addresses);
22        $this->assertEquals(3, $result->getModel()->getSize());
23        $this->assertEquals(2, $result->getModel()->getItemByIndex(1)->getId());
24    }

```

# Controller

```

1 <?php
2 require_once DOCUMENTROOT."/class/class.CPController.php";
3 require_once FWPATH."/objectmodel/base/class.PagingInfo.php";
4
5 /**
6  * @Authenticate
7  */
8 class AddressController extends CPController {
9
10     /**
11      * @var IAddressRepository
12      */
13     private $repository;
14     /**
15      * @var IEmailChimpService
16      */
17     private $mailchimpService;
18
19     public function __construct(IAddressRepository $repository, IEmailChimpService $mailchimpService, ICPCContext $context) {
20         $this->repository = $repository;
21         $this->mailchimpService = $mailchimpService;
22         $this->context = $context;
23     }
24
25     public function Liste($addressFilter = null) {
26         $paging = new PagingInfo(10);
27         $addresses = $this->repository->getAddresses(AddressFinder::select()
28             ->where(array(Address::prenameDataItem(), Address::surnameDataItem(), Address::companyDataItem(), Address::cityDataItem(), Address::
29                 $addressFilter, CompareFilter::LIKE_COMPARER, !isEmpty($addressFilter))
30             ->paging($paging)
31             ->order(Address::surnameDataItem(), Order::ORDER_ASC));
32         $this->assignPagingInfo("addressesPaging", $paging);
33
34         return $this->getView($addresses);

```

```

1  {* @Layout: _Layout.tpl *}
2  {boxtitle}
3  Adressen
4  ----
5  <a href="/Address/Create" class="plainlink">neue Adresse</a>
6  {/boxtitle}
7
8  {box}
9  <div class="buttonbar boxpadding">
10 <form name="filter" action="/Address/Liste" method="get">
11     Suche: {input name=addressFilter}
12     <input type="submit" value="suchen" />
13 </form>
14 </div>
15
16 <table class="full lined">
17     <tr>
18         <th>Adresse</th>
19         <th>Erstellt</th>
20         <th style="width:40px;"></th>
21     </tr>
22     {foreach item=address from=$model name=loop}
23     <tr>
24         <td>{include file="Shared/address.tpl" address=$address}</td>
25         <td>{$address.createdTS|date_format:"%d.%m.%Y"}</td>
26         <td>
27             <a href="/Address/Edit/{$address.id}">{icon name=EDIT}</a>
28         </td>
29     </tr>

```



> Livedemo

# Livedemo

# Fragen

