



Grundlagen der Web-Entwicklung

INF3172

Das Laminas-Framework

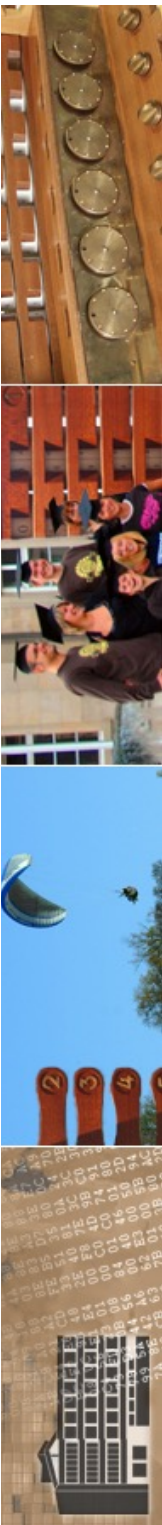
Thomas Walter

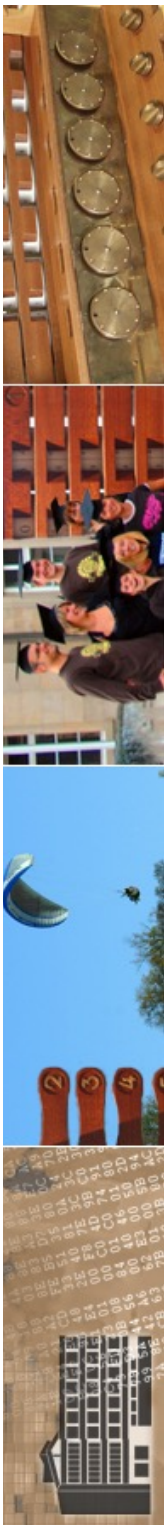
13.01.2022

Version 1.0

La//inas

La//inas



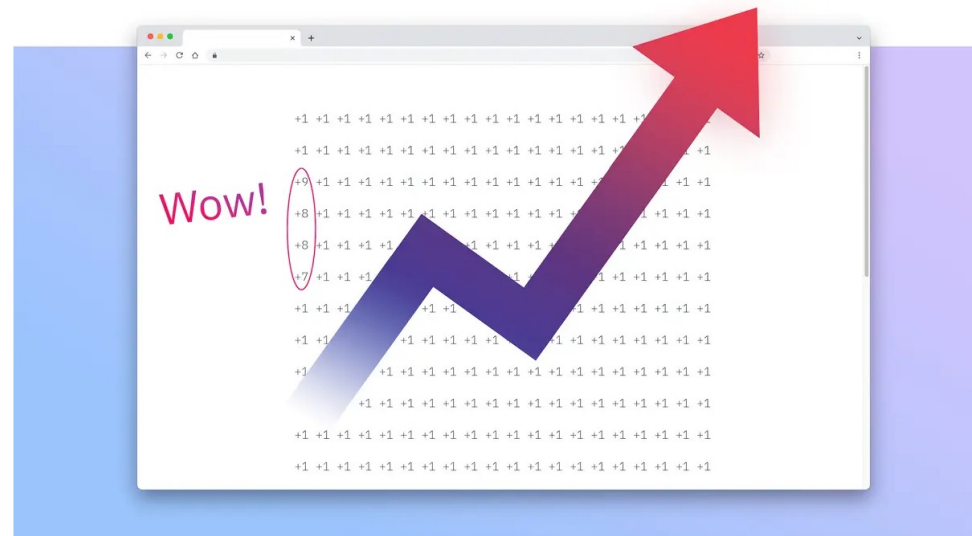


Web-Analyse mit Matomo: Seitenbesuche DSGVO-konform auswerten

Matomo fordert Google Analytics heraus: Das Web-Tool analysiert Seitenaufrufe ähnlich umfangreich wie Google, schützt aber die Privatsphäre der Besucher besser.

Lesezeit: 21 Min. In Pocket speichern

4



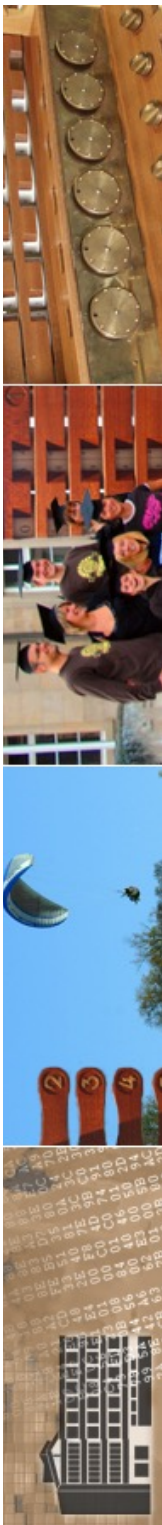
06.01.2022 11:33 Uhr

Von Daniel Berger

Wer eine Website betreibt, will auch wissen, ob sie besucht wird – und welche Seiten die Besucher am meisten aufrufen. Wer sich intensiv mit der Suchmaschinenoptimierung (SEO) befasst und gern neue Strategien ausprobiert, ist brennend an einer detaillierten Erfolgsmessung interessiert. Diese liefert Matomo, das bis 2018 noch Piwik hieß und sich seit Jahren erfolgreich als Alternative zu Google Analytics positioniert. Die Web-Anwendung ist Open-Source, funktionsreich und obendrein gratis.



La//inas





Frameworks

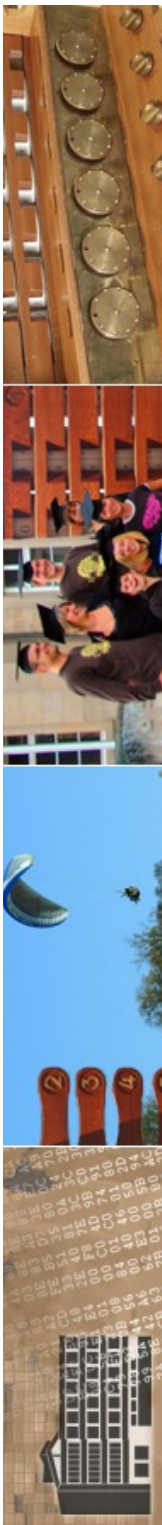
- Framework: wiederverwertbares Softwaresystem mit bereits implementierter, genereller (generischer) Funktionalität
 - Spezialisierung führt zu konkreter Anwendung
 - Framework setzt Architektur um
 - Framework folgt Design Pattern
 - Beispiele
 - Ruby: Ruby on Rails
 - PHP: ZEND Framework, Cake-PHP, Symfony, FLOW3
 - Python: Django





der Begriff "ZEND"

- Buchstaben einiger PHP-Entwickler:
 - www.zend.com
 - Andi Gutmans und Zeev Suraski
- ZEND-Engine
 - nicht-freie PHP-Engine mit besserer Performance etc.
- ZEND Studio
 - professionelle, kommerzielle IDE für PHP

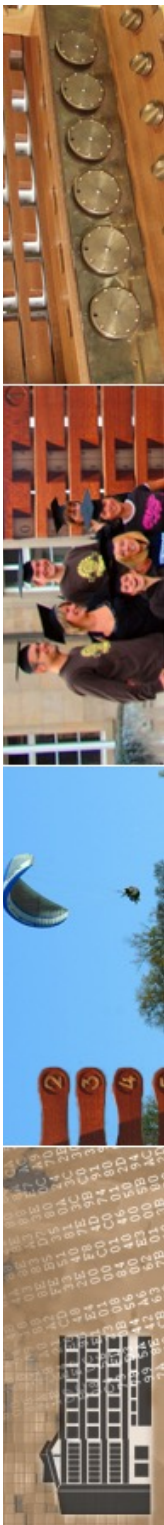


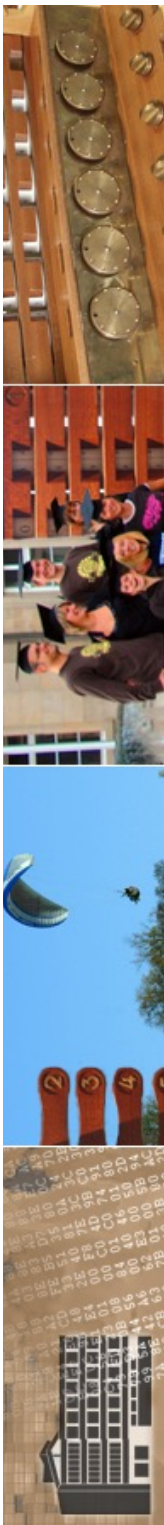


von ZF zu Lamias

- Zend Framework wechselte 2019 zur Linux Foundation

La//inas





INSTALL DOCUMENTATION TRAINING

We've moved!

Zend Framework is now the [Laminas Project](#). Please update your bookmarks.

(Unless you accept cookies, this notice will appear on every page.)

Zend Framework Has Launched

Zend Framework is now the Laminas Project!

Learn More





The Linux Foundation forms new Laminas project to support continued growth of Zend Framework and PHP tooling

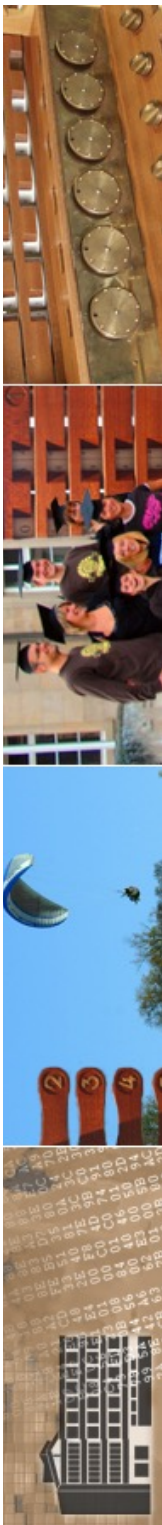
In conjunction with [Zend Technologies](#) and [Rogue Wave Software](#), we are excited to announce that the [Zend Framework](#) is transitioning to the Linux Foundation and will launch later this year as a new project called [Laminas](#).

The Zend Framework is a collection of professional PHP packages that can be used to develop web applications and services using PHP 5.6+, and it provides 100% object-oriented code using a broad spectrum of language features.

Over the years, the Zend Framework has seen wide adoption across industries and application types with [more than 400 million lifetime installs](#). It is used by companies including the BBC, BNP Paribas, and Offers.com. It has formed the basis of numerous business applications and services including eCommerce platforms, content management, healthcare systems, entertainment platforms and portals, messaging services, APIs, and many others.

The Linux Foundation will provide a vendor-neutral home for the Zend Framework community to continue to advance PHP tooling for the next generation of web services and APIs, while maintaining existing Zend Framework components, including the Apigility and Expressive subprojects.

To learn more about the Laminas project and how to get involved, please visit GetLaminas.org.





TRY FREE

Enterprise PHP Solutions, Groundbreaking PHP Support.

Support Your Mission-Critical Applications With Secure PHP Runtimes From Zend

TRY ZENDPHP

PHP Expertise & Support on Your Schedule

Working with PHP? Zend can help support, migrate, or modernize your PHP application.



Upgrade Flexibility

Zend supports older versions of PHP with security patches and updates, so you can upgrade on your own schedule.



Long-Term Support

Zend offers 3+ years of security and defect fixes after community support ends, meaning peace of mind for non-LTS PHP apps.



Security & Compliance

ZendPHP runtimes are certified by PHP experts and include the latest security patches and updates for enhanced PHP security and compliance.



Expert Guidance

From migrations and performance audits to consultative support, Zend provides expert guidance for your PHP project.



Framework Support

With enterprise support for Zend Framework and Laminas, you get ongoing security patches, expert consultative support, and more.



Performance Monitoring

Monitor performance of your PHP application and get custom alerts when your application needs attention with Zend Server and Z-Ray.



Laminas Project, the enterprise-ready PHP Framework and components

A community-supported, open source continuation of Zend Framework.

Components and MVC

Enterprise-ready PHP components and MVC framework

Components for any PHP application, including dependency injection, event dispatchers, input validation and filtering, pagination and navigation, feed generation and parsing, and much, much more.

A standards-based, event-driven MVC framework with a flexible workflow.

[Laminas documentation](#)

Mezzio

PHP Middleware in Minutes

Build PHP middleware applications, using the PSR-7 (HTTP Messages) and PSR-15 (HTTP Request Handlers) specifications, using a variety of routing and templating options, and the dependency injection container you prefer.

[Mezzio documentation](#)

API Tools

Build Beautiful RESTful APIs

Build RESTful APIs using standards such as Hypertext Application Language and Problem Details for HTTP APIs, and provide validation, authentication, authorization, and versioning right out of the box!

[API Tools documentation](#)






Performance

Engineered with performance tuning in mind. Zend Framework 3 runs up to 4x faster than version 2 releases!



Extensible

Open architecture based on Middleware or MVC, following [PHP-FIG](#) standards.



Secure

Built with security best practices. Store passwords using [bcrypt](#), encrypt with [AES-256](#), and more.



Enterprise Ready

A proven history of success running business critical and high-usage applications. Ready for [PHP 7](#).

Latest blog posts [\(full list \)](#)

- [From Zend to Laminas](#)
- [Zend Framework/Homestead Integration](#)
- [PHP 7.2 Support!](#)
- [Async Expressive? Try Swoole!](#)
- [Expressive 3!](#)

Latest security advisories [\(full list \)](#)

- [ZF2019-01: Information disclosure in zend-developer-tools](#)
- [ZF2018-01: URL rewrite vulnerability](#)
- [ZF2016-04: Potential remote code execution in zend-mail via Sendmail adapter](#)
- [ZF2016-03: Potential SQL injection in ORDER and GROUP functions of ZF1](#)

Issues and Security

Have you found a bug in Zend Framework?

[Report here!](#)

Have you identified a security vulnerability in ZF?

Please report it to us at zf-security@zend.com





About

Overview

Zend Framework is a collection of professional PHP packages with **more than 480 million** installations. It can be used to develop web applications and services using PHP 5.6+, and provides 100% object-oriented code using a broad spectrum of language features.

Zend Framework uses [Composer](#) as a package dependency manager; [PHPUnit](#) to test all packages; and [Travis CI](#) as a Continuous Integration service. Zend Framework also follows [PHP-FIG](#) standards, and includes an implementation of [PSR-7](#) for HTTP message interfaces (as shepherded by [Matthew Weier O'Phinney](#), Zend Framework project lead).

Zend Framework 3 evolved from both Zend Framework 2 and 1; cumulatively, the previous versions were downloaded more than 15 million times.

The principal sponsor of Zend Framework is [Zend](#), a [Rogue Wave Company](#), but many others have contributed components or significant features to the framework. Companies such as Google, Microsoft, and Strikelron have partnered with Zend to provide interfaces to web services and other technologies they wish to make available to ZF developers.

Zend Framework 3 could not deliver and support all of these features without the help of the vibrant community. Community members, including contributors, make themselves available on mailing lists, IRC channels and other forums. Whatever question you have about Zend Framework, the community is always available to address it.



Laminas und ZEND-Framework

- ZF: erste Version 30.6.2007
 - Januar 2020: 3 und 2.5
 - Januar 2021: Lamias
- siehe <http://www.zend.com/de/community/framework>
 - wichtig: <http://framework.zend.com/manual>
- sehr flexible Lizenz, auch kommerzieller Einsatz
- **MVC direkt abgebildet**
- Internationalisierung und Lokalisierung





Zend Framework

http://framework.zend.com/

Repositories 168

People 20

Projects 1

Grow your team on GitHub

GitHub is home to over 28 million developers working together. Join them to grow your own development teams, manage permissions, and collaborate on projects.

Sign up

Find a repository...

Type: All

Language: All

zend-test

Test component from Zend Framework

PHP 12 35 Updated 2 hours ago



Top languages

- PHP
- HTML
- CoffeeScript
- CSS

zend-http

Http component from Zend Framework

PHP 90 76 Updated 2 hours ago



Most used topics

- php
- psr-7
- psr-11
- zend-framework
- zendframework

zend-problem-details

Provides Problem Details for HTTP APIs (RFC 7807) support for PSR-7 applications.

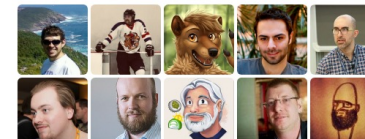
api php rest zend-expressive psr-7 zend-framework

PHP 41 15 Updated 5 hours ago



People

20 >





Documentation

Zend Framework 3

Zend Framework is a collection of **60+ packages** for professional PHP development. Each package is available on [GitHub](#) and can be installed via [Composer](#).

Tutorials

We provide tutorials for getting started with zend-mvc, understanding advanced topics of many components, and migration from version 2 to version 3.

[Read tutorials](#)

zend-authentication

Authenticate users via a variety of adapters, and provide the authenticated identity to your application.

[Github](#)

[Documentation](#)

zend-barcode

Programmatically create and render barcodes as images or in PDFs.

[Github](#)

[Documentation](#)

zend-cache

Caching implementation with a variety of storage options, as well as codified caching strategies for callbacks, classes, and output.

[Github](#)

[Documentation](#)

zend-captcha

Generate and validate CAPTCHAs using Figlets, images, ReCaptcha, and more.

[Github](#)

[Documentation](#)

Zend Framework 3

[Overview](#)

[Getting started](#)

[Tutorials](#)

Zend Framework 2

[Getting started](#)

[Reference guide](#)

[API](#)

Zend Framework 1

[Getting started](#)

[Reference guide](#)

[API](#)

Services

[Training & Certification](#)

[Support & Consulting](#)

[Webinars](#)



A Dependency Manager for PHP

Latest: **2.0.8** ([changelog](#))

Composer 2.0 is now available! [Read our announcement!](#)

[Getting Started](#)

[Download](#)

[Documentation](#)

[Browse Packages](#)

[Issues](#)

[GitHub](#)

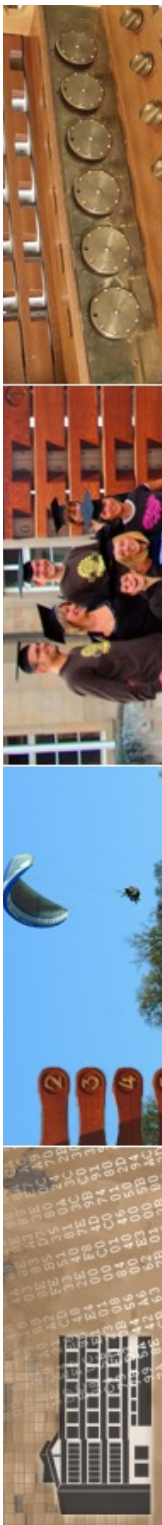
Authors: [Nils Adermann](#), [Jordi Boggiano](#) and many [community contributions](#)

Sponsored by:



Logo by: [WizardCat](#)

Composer and all content on this site are released under the [MIT license](#).





```
1 | $ composer create-project mezzio/mezzio-skeleton mezzio
```





LTE für ZEND und Lamias

Long Term Support

Periodically, we will announce a Long Term Support (LTS) version. LTS versions are supported for a duration of 3 years from the time of release with security and critical bug fixes.

Adopting an LTS version

Opting-in to an LTS version can be done by modifying the [Composer](#) requirement for Zend Framework:

```
$ composer require "zendframework/zendframework:~2.4.0"
```

The above will modify your `composer.json` to pin to the semantic version `2.4.0`, ensuring you only get updates in the 2.4 series. If you want to pin to a different LTS version, specify the `X.Y.0` version for it instead.

Versions

Version	Released	LTS Ends
Zend Framework 1.12	2015-04-13	2016-09-28
Zend Framework 2.4	2015-03-31	2018-03-31





Support and Long Term Support Policies

We provide two support policies, one governing general support for components, and another governing long term support for skeleton applications.

Terminology

- **Active support:** During this period, a package may receive new features, bugfixes, and/or security patches.
- **Security support:** During this period, a package may receive security patches and/or critical bugfixes only.
- **Security patches (or fixes):** Patches created due to a report of a potential or known security vulnerability, as verified by the [security team](#).
- **Critical bugfixes:** Fixes for bugs introduced within a new minor release cycle.
- **Skeleton package:** Any package defined as a [Composer "project" type](#), for the purpose of initiating a new project/application.
- **LTS:** Long-Term Support.

Component support policy

This policy governs individual component packages.

- When a new major release of a component is made, the previous minor release enters its *security support* phase for a period of 12 months, receiving only critical bugfixes and security fixes.
- Otherwise, a component remains in *active support* unless we provide notice of deprecation/abandonment. Once such a notice is made, the component will enter a *security support* phase for a period of 12 months.
- When a new minor release of a component is made, the previous minor release is no longer supported and will not receive security fixes.

Users are encouraged to update to the latest supported version. As we follow [semantic versioning](#), your code will continue to work without changes within the same major version.

Long Term Support Policy

This policy governs *skeleton packages* and their direct dependencies. Direct dependencies are any Zend Framework dependencies listed in the skeleton `composer.json`. At the time of publication, skeleton applications we provide include:

- [zf-apigility-skeleton](#) (Apigility applications)
- [ZendSkeletonApplication](#) (zend-mvc applications)
- [zend-expressive-skeleton](#) (Expressive applications)

Entwicklungsumgebung für das ZEND Framework

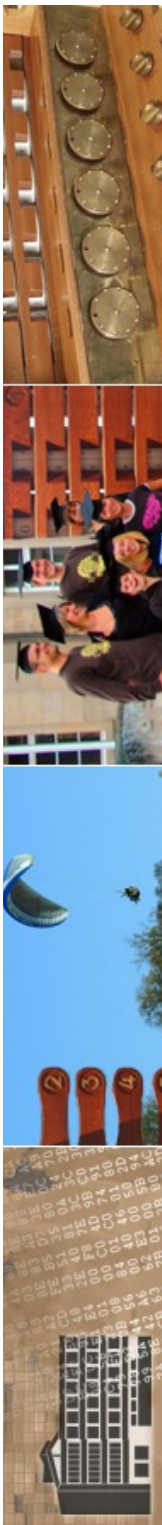
- IDE: Integrierte Entwicklungsumgebung
- ZEND-Studio for Eclipse
 - <http://www.zend.com/de/products/studio/>
 - nicht frei





ZF Version 1 und Version 2

- Unterschiede in der Version 2
 - neues Lizenzmodell (BSD)
 - noch stärker modularisiert
 - neuer ModulManager
 - neue Sicherheitsfunktionen (Kryptographie)
 - ZF1 wird noch weiter gepflegt





ZF Version 3

- Release Juni 2016
- Performance!
- PHP7
- entkoppeln von Paketen für bessere Wiederverwertbarkeit
- verbesserte Dokumentation





Laminas

Original Project	Original GitHub Organization	Original PHP Namespace	New Project Name	New GitHub Organization	New PHP Namespace
Zend Framework (Components)	zendframework	Zend	Laminas (Components)	laminas	Laminas
Zend Framework MVC	zendframework	Zend	Laminas (MVC)	laminas	Laminas
Apigility	zfcampus	ZF/ZF\Apigility	Laminas API Tools	laminas-api-tools	Laminas\ApiTools
Expressive	zendframework	Zend\Expressive	Mezzio	mezzio	Mezzio





Migration

La//inas | Documentation

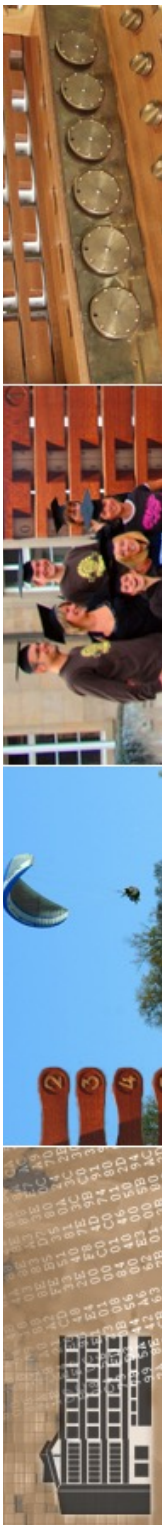
Migration to Laminas

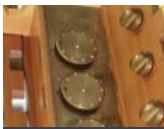
[Overview](#) » Migration to Laminas

Migration to Laminas

On this page

- Preparation
- Install laminas-migration
- Run the migration command
- Verify changes (optional)
- Install dependencies
- Test
- Post migration (optional)
- Summary



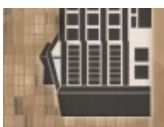


[Home](#) » [Documentation](#) » [tutorials](#) » [Migration](#) » [To Version 3](#) » [Overview](#)

Migration from Zend Framework v2 to v3

Zend Framework v2 to v3 has been intended as an incremental upgrade. We have even made efforts in the past year to provide forwards compatibility features in v2 versions of components, to allow users to prepare their code for upgrade.

This is not a comprehensive migration guide, however. While we know the majority of the areas where breakage can and will occur, we also know that only when developers are actually updating will we see the full situation. As such, treat this as a work in progress, and please feel free to propose updates or changes via [issues](#) or [pull requests](#) so we can improve!





Installation V 1/2

- das eigentliche Framework besteht aus PHP-Sourcen
 - herunterladen, entpacken
 - Apache mit PHP entsprechend konfigurieren
 - in `php.ini` ist die Direktive `include_path` anzupassen





Installation V3 (und V2 ab 2.5)

Get started

Installation of Zend Framework is only a [Composer](#) command away:

```
$ composer require zendframework/zendframework
```

This will install all **61** packages required in [composer.json](#). Alternately, all Zend Framework packages can be installed individually; for instance, if you need only the MVC package, you can install with the following command:

```
$ composer require zendframework/zend-mvc
```

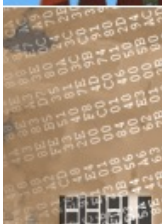
A list of all the packages can be found in the [documentation page](#).



```

D:\www\ZendFramework\ZendFramework-1.11.0\ZendFramework-1.11.0\INSTALL.txt - Notepad++
Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
anmerkungen_uh.bt httpd.conf jdbc-test-x.pl adventskalender.bt altesdatum.bt ohnedatum.bt webs4_20102_06.tex webs4_20102_07.tex INSTALL.bt README.bt
1  INSTALLATION
2  -----
3
4  Zend Framework requires no special installation steps. Simply download the framework,
5  extract it to the folder you would like to keep it in, and add the library directory
6  to your PHP include_path. To use components in the extras library, add the extras/library
7  directory to your PHP include_path, as well.
8  If you would like to use Zend_Tool, simply add bin/zf.bat (for Windows) or
9  bin/zf.sh (for anything else) to your system executable path.
10
Normal text file 3355 chars 3438 bytes 84 lines Ln: 1 Col: 1 Sel: 0 (0 bytes) in 0 ranges UNIX ANSI INS

```





Composer? Composer!

- Composer:
 - Paket-Manager für PHP
 - erkennt Abhängigkeiten
 - <https://getcomposer.org/>



A Dependency Manager for PHP

Latest: **2.0.8** ([changelog](#))

Composer 2.0 is now available! [Read our announcement!](#)

[Getting Started](#)

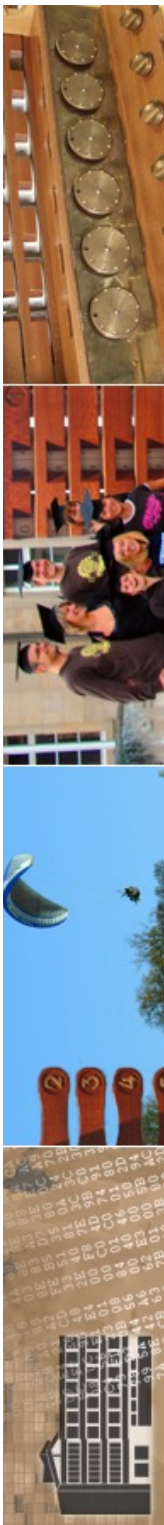
[Download](#)

[Documentation](#)

[Browse Packages](#)

[Issues](#)

[GitHub](#) 29





Installation

```
1 | $ composer require --dev laminas/laminas-component-installer
```

Global Installation

You can also install the plugin globally, in which case it will be active for every project you manage on your machine.

```
1 | $ composer global require laminas/laminas-component-installer
```





implicit_flush	Off
include_path	.;C:\Program Files (x86)\php\php-5.3.2-Win32-VC6-x86;C:\Program Files (x86)\php\php-5.3.2-Win32-VC6-x86\ext;C:\Program Files (x86)\PostgreSQL\8.4\lib;C:\Program Files (x86)\php\ZendFramework-1.11.0\library
log_errors	On
log_errors_max_len	1024





Namenskonvention und Struktur

- viele moderne Frameworks verwenden eine **Namenskonvention**
 - angelehnt an Java-Packages
 - Beispiel: PHP-Datei (Klasse)
`Zend_Currency_Validate`
ist Datei `Validate.php` und liegt im Ordner
`Zend/Currency`





Laden von Klassen-Dateien

- klassisch-PHP: `require_once`
- Alternative: Klasse `Zend_Loader` mit statischer Methode `loadClass()`
 - lädt Datei/Klasse und prüft, ob in Datei korrekte Klasse vorhanden ist
 - kein doppeltes Einbinden, wenn Klasse schon vorhanden
 - ```
require_once('Zend/Loader.php');
Zend_Loader::loadClass('Zend_Beispielklasse');
```





# Performance

- allgemein:  
*Frameworks mindern die Performance, da deutlicher Overhead*
- Optimierung sinnvoll, etwa durch Caching des PHP-Codes
  - Zwischenzeitliches Speichern des kompilierten PHP-Codes
    - eAccelerator
    - APC
    - Zend Platform





# MVC

- ZEND-Framework setzt konsequent auf MVC:  
**Model-View-Controller** (→ Design Patterns)

– empfohlene Verzeichnisstruktur  
`application`

`/controllers`

`/models`

`/views`

`/scripts`

`/helpers`

`/filters`

`html`

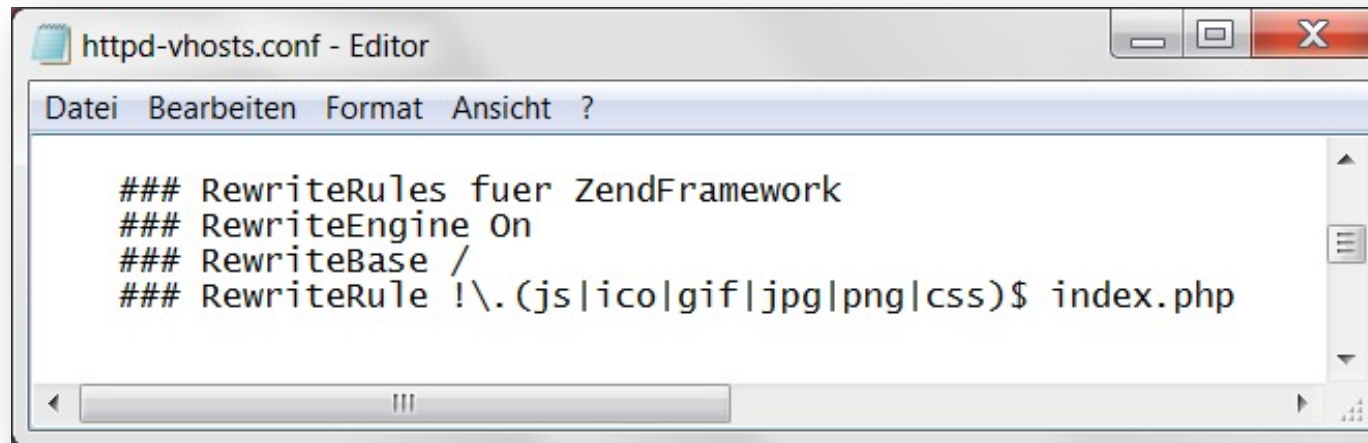
(→ `htdocs`)



# Rewrite Rule

- mittels einer Rewrite-Rule auf dem Apache werden alle Requests außer Endungen js/ico/gif/jpg/png/css an die Datei index.php weitergeleitet:

```
RewriteRule !\.(js|ico|gif|jpg|png|css)$
index.php
```



```
RewriteRules fuer ZendFramework
RewriteEngine On
RewriteBase /
RewriteRule !\.(js|ico|gif|jpg|png|css)$ index.php
```



# Anlegen eines Projektes

- das Dienstprogramm zf erzeugt ein neues "leeres" Projekt

```

DOS Shell
d:\www\zend111>
d:\www\zend111>
d:\www\zend111>zf create project zfstart
Creating project at D:/www/zend111/zfstart
Note: This command created a web project, for more information setting up your V
HOST, please see docs/README

d:\www\zend111>

```





# Verzeichnisbaum eines Projektes

```

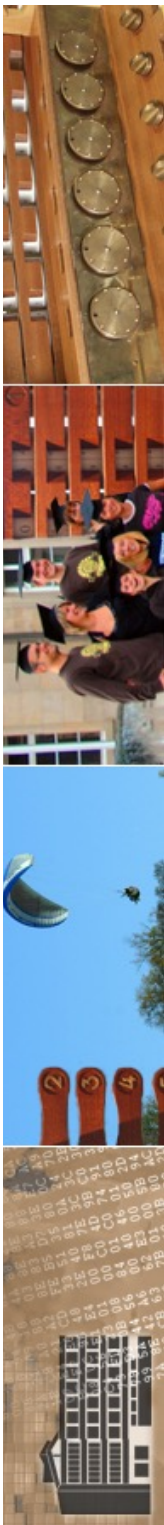
zfstart
|-- application
| |-- Bootstrap.php
| |-- configs
| | `-- application.ini
| |-- controllers
| | |-- ErrorController.php
| | `-- IndexController.php
| |-- models
| `-- views
| |-- helpers
| `-- scripts
| |-- error
| | `-- error.phtml
| `-- index
| `-- index.phtml
|-- library
|-- public
| |-- .htaccess
| `-- index.php
`-- tests
 |-- application
 | `-- bootstrap.php
 |-- library
 | `-- bootstrap.php
 `-- phpunit.xml

```

© 2022 Universität Tübingen

La//inas

UNIT 3/12 - Grundlagen der Web-Entwicklung, Wintersemester 2021/22





# Konfiguration eines Projektes

- im Unterordner  
`application/configs`  
die Datei  
`application.ini`

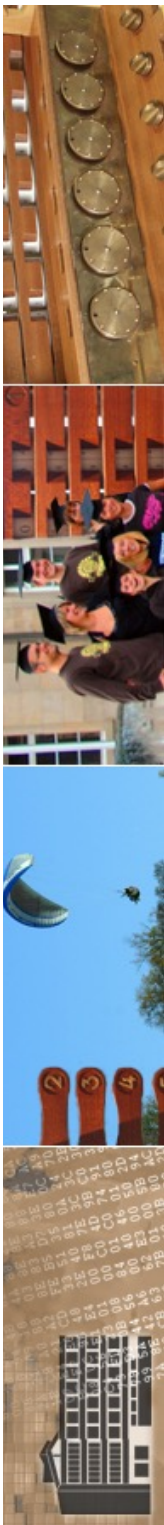




```

D:\www\zend111\zfstart\application\configs\application.ini - Notepad++
Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ? X
ToDos_DFG.txt Programm_Tuebingen.txt zki_tagung_infos.txt index.php BeispielController.php index.phtml DbBeispielController.p
1 [production]
2 phpSettings.display_startup_errors = 0
3 phpSettings.display_errors = 0
4 includePaths.library = APPLICATION_PATH "../library"
5 bootstrap.path = APPLICATION_PATH "/Bootstrap.php"
6 bootstrap.class = "Bootstrap"
7 appnamespace = "Application"
8 resources.frontController.controllerDirectory = APPLICATION
9 resources.frontController.params.displayExceptions = 0
10
11 [staging : production]
12
13 [testing : production]
14 phpSettings.display_startup_errors = 1
15 phpSettings.display_errors = 1
16
17 [development : production]
18 phpSettings.display_startup_errors = 1
19 phpSettings.display_errors = 1
20 resources.frontController.params.displayExceptions = 1
21
MS ini file 633 chars 653 bytes 21 lines Ln: 1 Col: 1 Sel: 0 (0 bytes) in 0 ranges UNIX ANSI INS

```

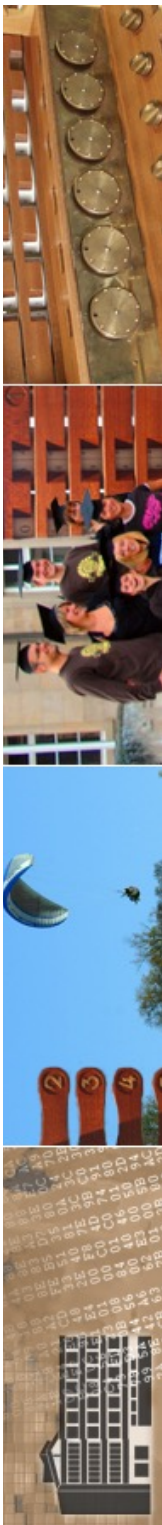






# der FrontController

- Aufgabe des FrontControllers
  - steuert letztlich das Erscheinungsbild der Anwendung
  - setzt wichtige Konfigurationen
  - nimmt alle Anfragen entgegen





# der FrontController

- ein einfacher FrontController (index.php):

```
require_once 'Zend/Controller/Front.php';
// Instanz auslesen
$fc = Zend_Controller_Front::getInstance();
// Verzeichnis setzen fuer ActionController
$fc->setControllerDirectory('...wo auch immer...');
// Views unterdruecken
$fc->setParam('noViewRenderer', true);
//Ausnahmenbehandlung
$fc->throwExceptions(true);
// Error Handler ausschalten
$fc->setParam('noErrorHandler', true);
$fc->dispatch();
```





```

D:\www\ZendFramework\index.php - Notepad++
Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
unibrief_cls unibrief_cls id_kopf.tex Tagebuch.bt medizin-profs.bt ksk.bt httpd.conf localconf.php webst4_20092_05.tex
1 <?php
2 /*
3 * Grundlagen der Web-Entwicklung
4 * das ZEND-Framework
5 * einfacher FrontController
6 */
7
8 require_once 'Zend/Controller/Front.php';
9 // Instanz auslesen
10 $fc = Zend_Controller_Front::getInstance();
11 // Verzeichnis setzen fuer ActionController
12 $fc->setControllerDirectory('D:\www\ZendFramework\Controllers');
13 // Views unterdruecken
14 $fc->setParam('noViewRenderer', true);
15 //Ausnahmenbehandlung
16 $fc->throwExceptions(true);
17 // Error Handler ausschalten
18 $fc->setParam('noErrorHandler', true);
19 $fc->dispatch();
20
21 ?>
PHP Hypertext Preproc nb char : 547 nb line : 21 Ln:1 Col:1 Sel:0 Dos\Windows ANSI INS

```





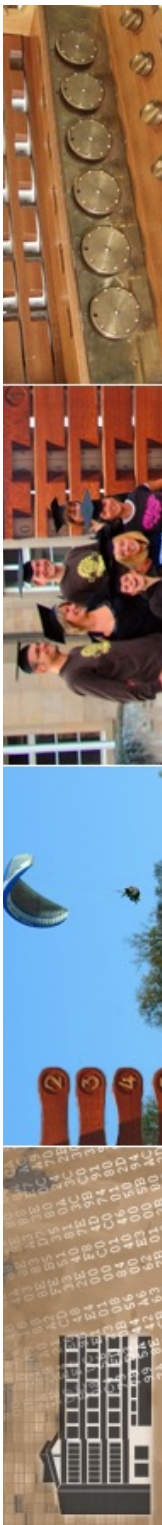
# der ActionController

- Aufgabe des ActionController: Übernimmt requests vom FrontController und erzeugt Antwort
- Minimalvariante:

```
require_once('Zend/Controller/Action.php');

class IndexController extends Zend_Controller_Action
{
 public function indexAction() {
 echo('Grundlagen der Web-Entwicklung:

 ActionController');
 } // indexAction
} // class IndexController
```



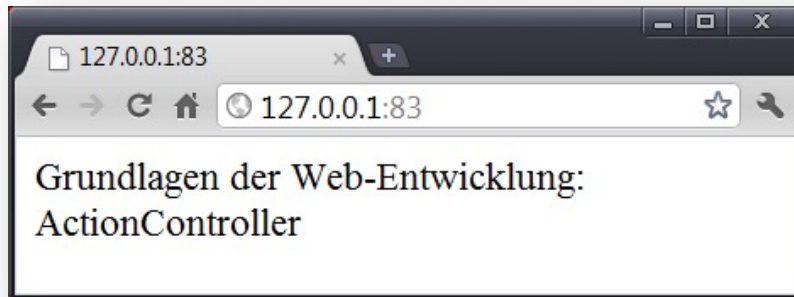
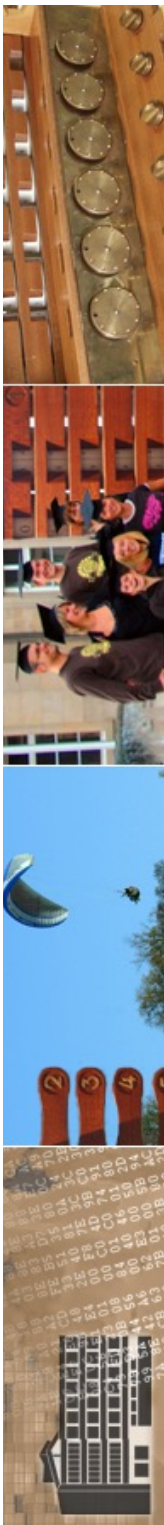


```

D:\www\ZendFramework\Controllers\IndexController.php - Notepad++
Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
unibrief.cls id_kopf.tex Tagebuch.txt medizin-profs.txt ksk.txt httpd.conf localconf.php webst4_20092_05.tex INSTALL.txt index
1 <?php
2 /*
3 * Grundlagen der Web-Entwicklung
4 * das ZEND-Framework
5 * einfacher FrontController
6 */
7
8 require_once('Zend/Controller/Action.php');
9
10 class IndexController extends Zend_Controller_Action {
11
12 public function indexAction() {
13 echo('Grundlagen der Web-Entwicklung:
ActionController');
14 } // indexAction
15 } // class IndexController
16
17 ?>
PHP Hypertext Preprocessor file nb char : 365 nb line : 17 Ln:1 Col:1 Sel:0 Dos\Windows ANSI INS

```







# ein zweiter ActionController

- nun ergänzen wir im gleichen Verzeichnis einen zweiten ActionController mit dem Namen BeispielController
- dieser soll zwei Methoden implementieren:
  - `indexAction`
  - `beispielAction`





```

1 <?php
2 /*
3 * Grundlagen der Web-Entwicklung
4 * das ZEND-Framework
5 * BeispielController mit zwei Methoden
6 */
7
8 require_once('Zend/Controller/Action.php');
9
10 class BeispielController extends Zend_Controller_Action {
11
12 public function indexAction() {
13 echo('Grundlagen der Web-Entwicklung:
Methode indexAction aus BeispielController');
14 } // indexAction
15
16 public function beispielAction() {
17 echo('Grundlagen der Web-Entwicklung:
Methode beispielAction aus BeispielController');
18 } // indexAction
19
20 } // class IndexController
21
22 ?>

```

PHP Hypertext Preprocessor file      nb char : 562   nb line : 22      Ln : 1   Col : 1   Sel : 0      Dos\Windows ANSI      INS



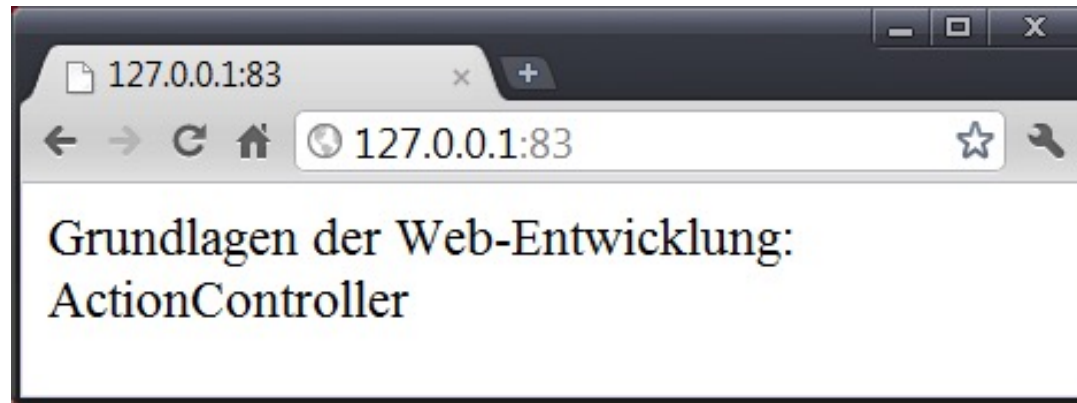
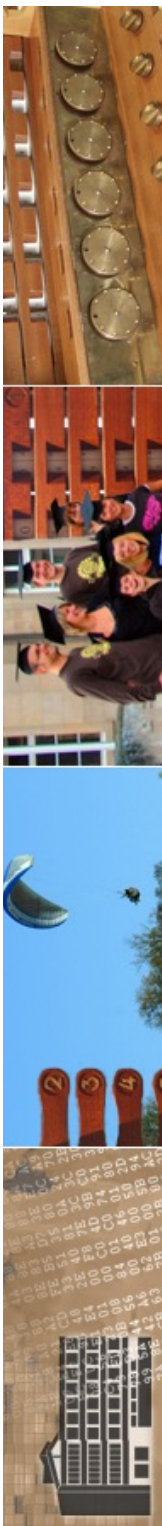




# was passiert?

- nun sind schon viele Varianten möglich:
  - / → IndexController/indexAction
  - /Beispiel → BeispielController/indexAction
  - /Beispiel/beispiel →  
BeispielController/beispielAction







# Struktur der ActionController

- die Controller können zur Übersichtlichkeit in verschiedene Verzeichnisse (samt Unterstruktur) abgelegt werden (Unterordner controllers)
  - Realisierung über assoziatives PHP-Array
  - zentrales Key-Value-Paar default

```

11
12 // Verzeichnisstruktur setzen fuer ActionController
13 $fc->setControllerDirectory(
14 array('default' => 'D:\www\ZendFramework\controllers',
15 'cms' => 'D:\www\ZendFramework\controllers\cms',
16 'admin' => 'D:\www\ZendFramework\controllers\admin'
17);
18

```

© 2  
Tü

PHP Hypertext Preprocessor | nb char : 706 | nb line : 27 | Ln : 11 | Col : 1 | Sel : 0 | Dos\Windows ANSI | INS



# die View-Komponente

- Nutzung der View steuert FrontController:
  - `$fc->setParam('noViewRenderer', true);`
    - (Ersetzung true → false)
- erzeugt zunächst nur Fehlermeldung, da kein View-Template vorhanden ist





# View-Template

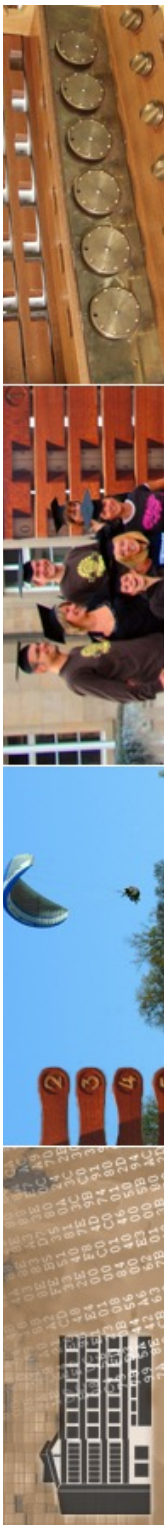
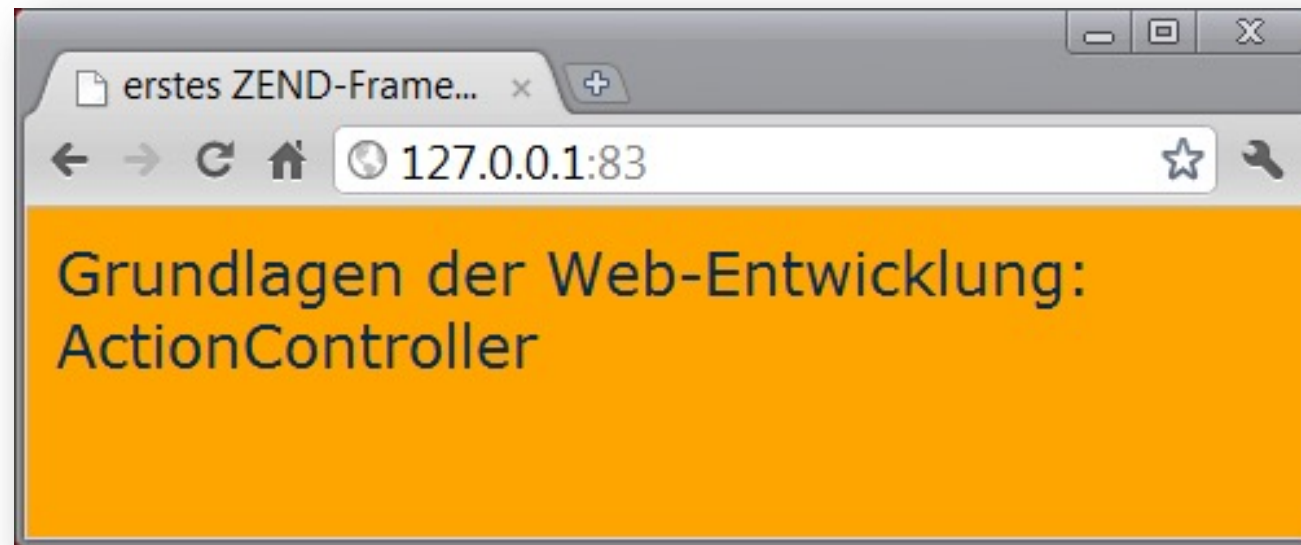
- View-Template: HTML-Code mit PHP (Endung .phtml)
- Namenskonvention: Im Unterordner `views/scripts` Unterordner mit Namen des Controllers, dort Datei mit Namen der Methode des Controller
- Beispiel:
  - `.../views/scripts/index/index.phtml` ist Template für den Controller `indexController`





```
D:\www\ZendFramework\views\scripts\index\index.phtml - Notepad++
Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
unibrief_cls unibrief.cls id_kopf.tex Tagebuch.bt medizin-profs.bt ksk.bt httpd.conf localconf.php index.php IndexController.php BeispielController.php
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-5">
6 <title>erstes ZEND-Framework-Template</title>
7 </head>
8 <body>
9 <?php
10
11 ?>
12 </body>
13 </html>
```

```
http://127.0.0.1:82/ view-source:http://127.0.0.1:82/
view-source:http://127.0.0.1:82/
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-5">
6 <title>erstes ZEND-Framework-Template</title>
7 </head>
8 <body>
9 </body>
10 </html>
11 Grundlagen der Web-Entwicklung:
ActionController
```





```

D:\www\ZendFramework\Controllers\IndexController.php - Notepad++
Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
Tagebuch.txt medizin-profs.bt ksk.bt httpd.conf localconf.php index.php IndexController.php BeispielController.php index - Kopie.php
1 <?php
2 /*
3 * Grundlagen der Web-Entwicklung
4 * das ZEND-Framework
5 * einfacher ActionController
6 */
7
8 require_once('Zend/Controller/Action.php');
9
10 class IndexController extends Zend_Controller_Action {
11
12 public function indexAction()
13 {
14 // View ansprechen
15 $view = $this->getView();
16 $view->ausgabe = 'index';
17 } // indexAction
18 } // class IndexController
19 ?>
PHP Hypertext Preprocessor file nb char : 445 nb line : 19

```

```

D:\www\ZendFramework\views\scripts\index\index.phtml - Notepad++
Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
medizin-profs.bt ksk.bt httpd.conf localconf.php index.php IndexController.php BeispielController.php index - Kopie.p
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html;
6 <title>erstes ZEND-Framework-Template</title>
7 </head>
8 <body>
9 <?php
10 echo $this->ausgabe;
11 ?>
12 </body>
13 </html>
PHP Hypertext Preprocessor file nb char : 319 nb line : 13 Ln : 1 Col : 1 Sel : 0 MAC ANSI INS

```







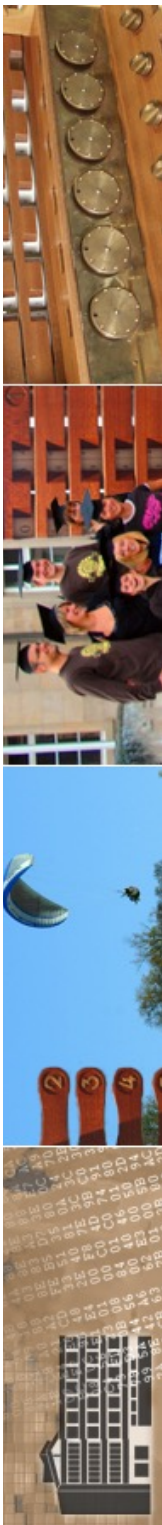
# Vorteil dieses Frameworks

## typische Frameworks

- HTML-artige Templates
- eigene Template-Sprache mit wenigen Direktiven

## ZEND Framework

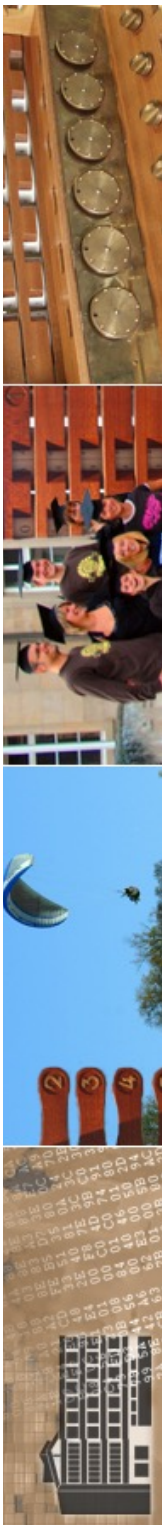
- HTML-artige Templates
- Template-Sprache ist **reines PHP**





# die Verarbeitungsschritte

- FrontController
  - nimmt alle Anfragen entgegen
  - zentraler Dispatcher zu den Action-Controllern
  - Steuerung
- Anfragen und Antworten sind Requests- und Response-Objekte
- Templates steuern View, view-Objekt verwaltet anzuzeigende Inhalte





# Werteübergabe

- Übernahme von Werten aus dem HTTP-Request
  - PHP alleine: Superglobals
    - `$_POST`
    - `$_GET`
    - `$_REQUEST`
  - ZEND Framework: Request-Objekt im ActionController
    - Objekt `Zend_Controller_Request_Http`
    - Methode `getRequest()`
      - `$request = $this->getRequest();`
      - `$wert = $request->getPost('name');`





# die Framework-Methoden

| Framework-Methode         | PHP Superglobal        |
|---------------------------|------------------------|
| <code>getPost ()</code>   | <code>\$_POST</code>   |
| <code>getQuery ()</code>  | <code>\$_GET</code>    |
| <code>getServer ()</code> | <code>\$_SERVER</code> |
| <code>getCookie ()</code> | <code>\$_COOKIE</code> |
| <code>getEnv ()</code>    | <code>\$_ENV ()</code> |





# das Model

- Model: frei definierbar  
(etwa in eigenen PHP-Klassen)
- im ZEND-Framework: Verzeichnis `models`
  - Übersichtlichkeit des Codes





# Error-Handling

- Fehler-Behandlung im Framework
  - im FrontController kann auch der Umgang mit Fehlern grundlegend gesteuert werden
  - Beispiel: HTTP-Fehler 404
    - Zugriff auf nicht existierende Controller-Methode  
`Zend_Controller_Action_Exception`
    - Zugriff auf nicht existierenden Controller  
`Zend_Controller_Dispatcher_Exception`





# Konfiguration im FrontController

- wichtiger Parameter im Front-Controller;  
`$fc->throwExceptions(true);`
  - typisch: true für Testsystem, false für Produktivsystem
- aktivieren des Error-Handlings-Plugins:  
`$fc->setParam('noErrorHandler', true);`
  - false aktiviert Plugin
  - bei Fehler: Methode `errorAction` aus Controller-Klasse `ErrorController`



```
D:\www\ZendFramework\controllers\ErrorController.php - Notepad++
Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Aus
localconf.php index.php IndexController.php BeispielController.php index - K
1 <?php
2 /*
3 * Grundlagen der Web-Entwicklung
4 * das ZEND-Framework
5 * Beispiel fuer ErrorController
6 */
7
8 require_once('Zend/Controller/Action.php');
9
10 class ErrorController extends Zend_Controller_Action {
11
12 public function errorAction() {
13
14 echo('Es ist ein Fehler aufgetreten!

');
15
16 $response = $this->getResponse();
17 $exceptions = $response->getException();
18 foreach ($exceptions as $exception)
19 echo($exception->getMessage(). "
");
20
21 } // errorAction
22
23 } // class ErrorController
24
25 ?>
```

http://127.0.0.1:82/Abc

Es ist ein **Fehler** aufgetreten!

Invalid controller specified (Abc)

PHP Hypertext Preprc nb char : 539 nb line : 25 Ln: 1 Col: 1 Sel: 0 Dos\Windows ANSI INS





# Datenbankzugriff

- das ZEND-Framework erlaubt ebenfalls einfachen Datenbankzugriff
- Klasse  
Zend\_Db
  - Datenbanklayer unabhängig vom jeweiligen DBMS
  - mögliche Treiber:
    - MySQLi, db2, Oracle, MS SQL Server, Postgres, SQLite





Home » Documentation » zend-db » zend-db

# zend-db

zend-db

## Installation

### Using Composer

```
1 | $ composer require zendframework/zend-db
```

## Support

- Issues: [github.com/zendframework/zend-db/issues](https://github.com/zendframework/zend-db/issues)
- Source: [github.com/zendframework/zend-db](https://github.com/zendframework/zend-db)
- Chat: [zendframework-slack.herokuapp.com](https://zendframework-slack.herokuapp.com)
- Forum: [discourse.zendframework.com](https://discourse.zendframework.com)



# Connect zum DBMS

- statische Methode `factory()` in Klasse `Zend_Db`
- Parameter: Treiber und PHP-Array mit Optionen:
  - `host`, `user`, `password`, `dbname`, `port` und weitere Optionen





```

D:\www\ZendFramework\controllers\DbBeispielController.php - Notepad++
Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
index.php IndexController.php BeispielController.php index - Kopie.php index.phtml ErrorController.php DbBeispielController.php
7
8 require_once('Zend/Controller/Action.php');
9 require_once('Zend/Db.php');
10
11 class DbBeispielController extends Zend_Controller_Action {
12
13 public function indexAction() {
14
15 // Verbindungsaufbau
16 $options = array(
17 'host' => '127.0.0.1',
18 'username' => 'thomas',
19 'password' => '',
20 'dbname' => 'webkompendium'
21);
22 $dbh = Zend_Db::factory('mysqli', $options);
23 echo(' DB-Verbindung erfolgreich
');

```

PHP Hypertext Preproc nb char : 838 nb line : 36 Ln : 30 Col : 13 Sel : 0 Dos\Windows ANSI INS





# select-Abfrage

- nach DB-Verbindung übliche Select-Abfrage:
  - SQL-Statement definieren
  - Durchführung (query)
  - ResultSet zeilenweise verarbeiten (fetch)



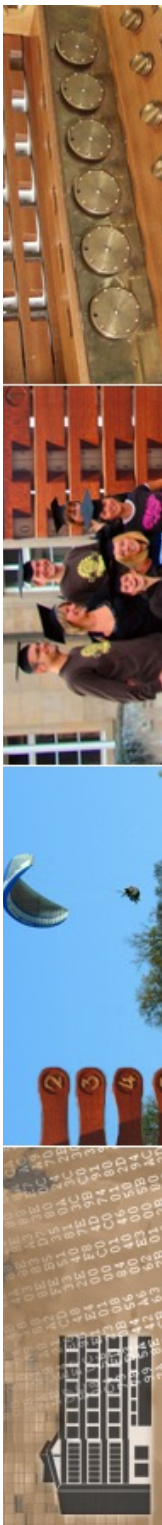
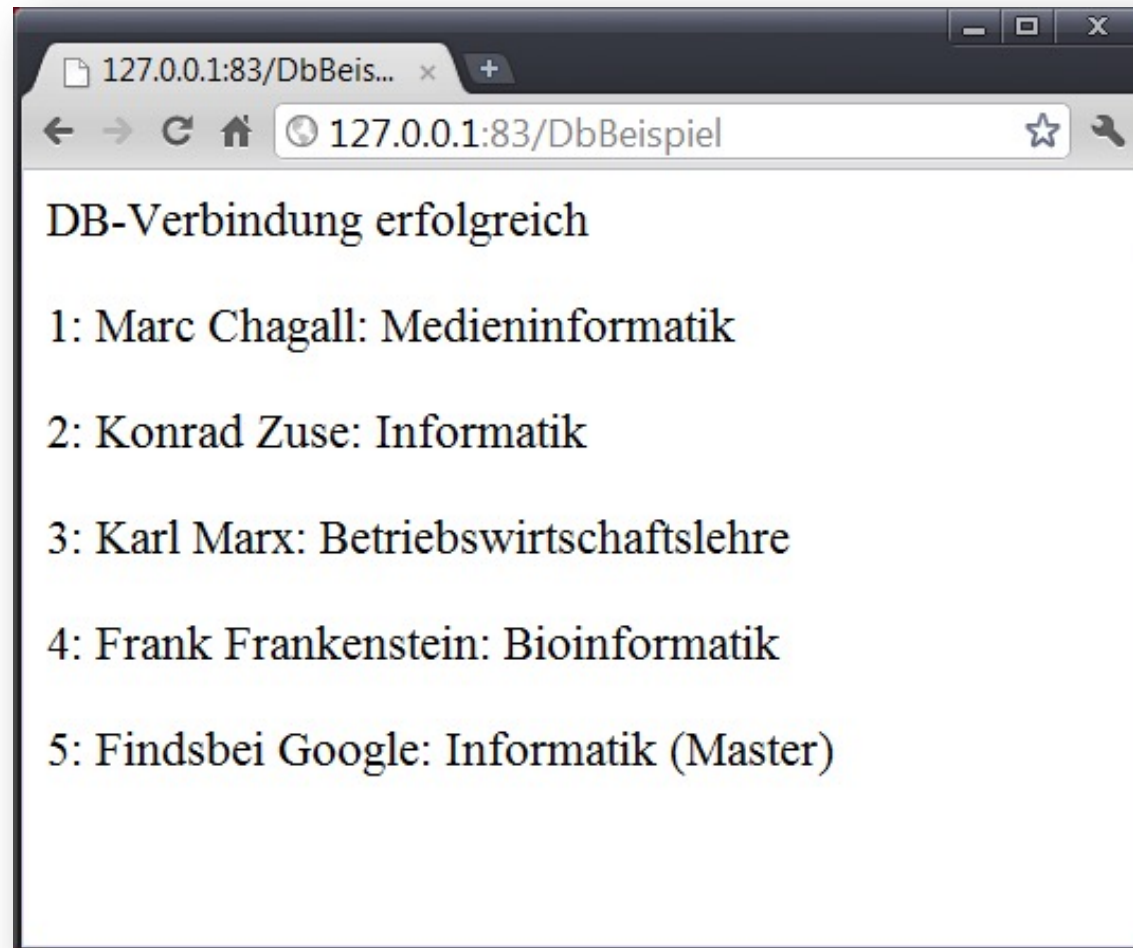


```

D:\www\ZendFramework\controllers\DbBeispielController.php - Notepad++
Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
Readme.txt ToDos_DFG.txt Programm_Tuebingen.txt zki_tagung_infos.txt index.php BeispielController.php index.phtml DbBeispielController.php
20 'dbname' => 'webkompendium'
21);
22 $dbh = Zend_Db::factory('mysqli',$options);
23 echo(' DB-Verbindung erfolgreich
');
24
25 // Select-Abfrage
26 $sql = "SELECT * FROM student";
27 $res = $dbh->query($sql);
28 while($zeile = $res->fetch())
29 echo "<P>".$zeile['mtknr']. ": ".$zeile['vname']. " ".$zeile['nname'].":
30
31 } // indexAction
32
33
34 // class IndexController
35
PHP Hypertext Preprocessor file 776 chars 846 bytes 36 lines Ln : 29 Col : 5 Sel : 0 (0 bytes) in 0 ranges Dos\Windo ANSI INS

```







# Benutzer- und Rechtemanagent

- Das ZEND-Framework bietet Klasse Zend\_Acl (Access Control Lists)
  - Aufbau eines (hierarchischen) Rechtesystems
  - Methoden wie `addRole` und `allow` und `isAllowed`
- Rechte können **vererbt** werden







# eigenes Session-Management

- das ZEND-Framework enthält ein eigenes Session-Management
  - Klasse `Zend_Session`
- stark ähnlich Session-Management von PHP





# Infrastruktur-Klassen

- zum ZEND-Framework gehören wichtige Infrastruktur-Klassen
  - Performance-Optimierung: `Zend_Cache`
  - Plausibilitäts-Test: `Zend_Validate`
  - Filtern von Daten: `Zend_Filter`
  - Formular-Verarbeitung: `Zend_Filter_Input`
  - Logilfes: `Zend_Log`
  - Konfigurationsverwaltung: `Zend_Config`





# Ausblick

- weitere Möglichkeiten des Laminas/ZEND-Frameworks:
  - Web-Services
  - RSS- / Atom-Feeds
  - pdf-Generierung
  - Captcha
  - direkte Schnittstelle zu Google-Diensten
  - Mail
  - und sehr viel mehr.





# ...und nun...

- kennen wir ein modernes, typisches Framework mit seiner Arbeitsweise als konkretes Architekturbeispiel kennen gelernt
- weitere, noch komplexere Beispiele zu Frameworks
- Python-Frameworks

