

Extensions

Introduction

- ❖ The PHP execution environment includes the following three modules:

Core PHP

Zend Engine

Extension

- ❖ The Core PHP module handles the very basic operations such as requests, file streams, error handling and similar.

Introduction

- ❖ The Zend Engine module is responsible for converting human readable code into machine understandable op-codes. Once the conversion completes it executes the generated op-code on a virtual machine.
- ❖ The PHP Extensions are bunches of functions, classes and streams that are available to code we write in PHP. The extensions usually target specific tasks, such as the mysql extension that allows us to connect MySQL. and similar.

Introduction

- ❖ When the Zend Engine executes generated code that requires access to any of the extensions it passes the control to the required extension. Once that extension completes the required operation it transfer back the control.

The Available Extensions

- ❖ You can find a complete list of the available extensions at <http://php.net/manual/en/extensions.alphabetical.php>.

Extensions Development

- ❖ It is possible to develop our own new extension. Each extension should include a configuration file. Its name should be config.m4. In most cases the rest of the files will be in C.

Pear

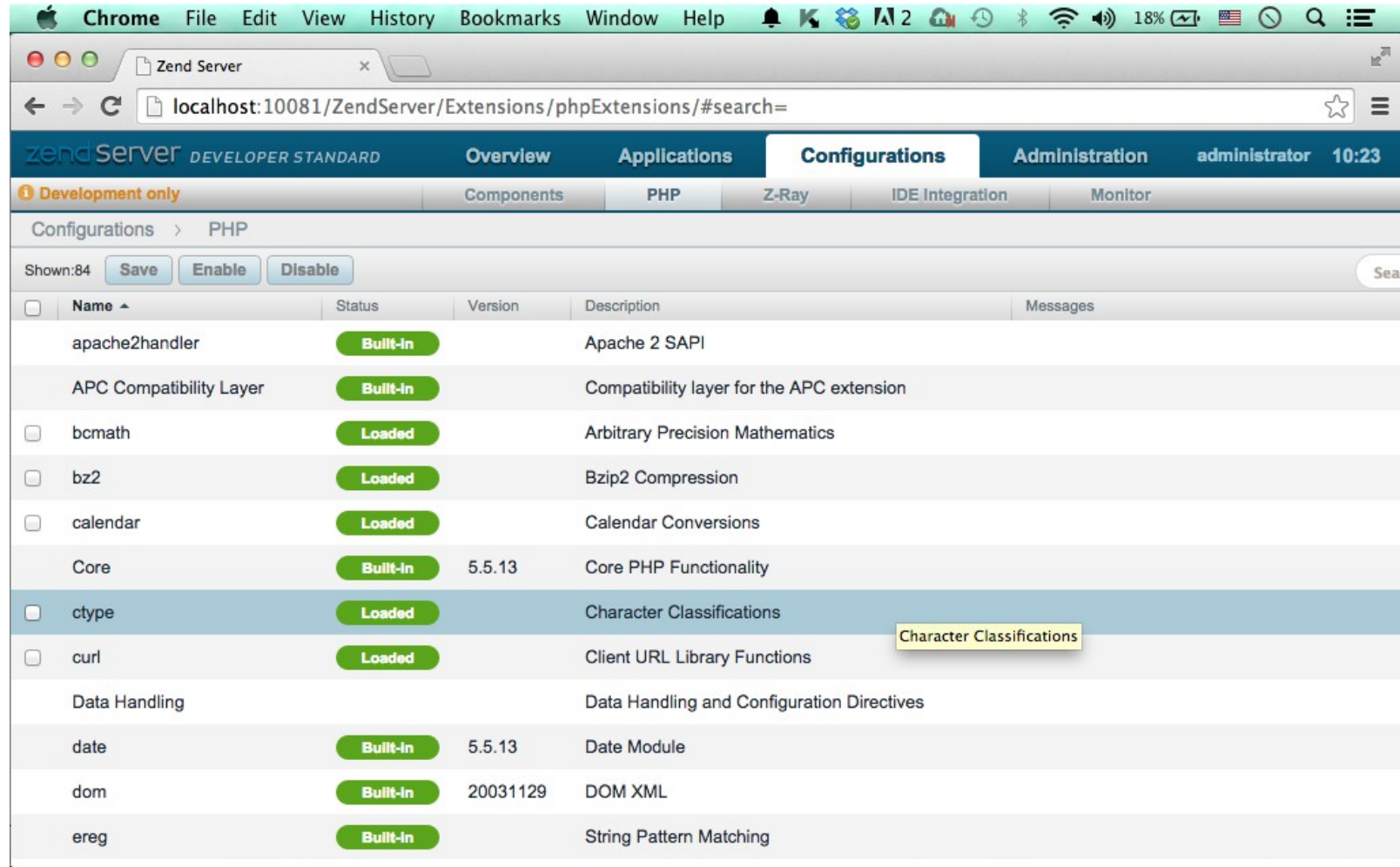
- ❖ Pear is a PHP framework and distribution system for reusable PHP components.
- ❖ Since October 2011, 5 million lines of the PEAR codebase has shifted to github.

Zend Server Control Panel

- ❖ Using Zend Server we can browse its control panel at <http://localhost:10081>.
- ❖ We can find an easy to use interface for managing the extensions Zend server uses.



Zend Server Control Panel



The screenshot shows the Zend Server Control Panel interface in a Chrome browser. The address bar displays `localhost:10081/ZendServer/Extensions/phpExtensions/#search=`. The navigation bar includes tabs for Overview, Applications, Configurations (selected), Administration, and administrator. The Configurations tab is further divided into Components, PHP (selected), Z-Ray, IDE Integration, and Monitor. The main content area shows the PHP configuration page with a table of extensions. The 'ctype' extension is highlighted, and a tooltip displays its description: 'Character Classifications'.

Name	Status	Version	Description	Messages
apache2handler	Built-In		Apache 2 SAPI	
APC Compatibility Layer	Built-In		Compatibility layer for the APC extension	
<input type="checkbox"/> bcmath	Loaded		Arbitrary Precision Mathematics	
<input type="checkbox"/> bz2	Loaded		Bzip2 Compression	
<input type="checkbox"/> calendar	Loaded		Calendar Conversions	
Core	Built-In	5.5.13	Core PHP Functionality	
<input type="checkbox"/> ctype	Loaded		Character Classifications	
<input type="checkbox"/> curl	Loaded		Client URL Library Functions	
Data Handling			Data Handling and Configuration Directives	
date	Built-In	5.5.13	Date Module	
dom	Built-In	20031129	DOM XML	
ereg	Built-In		String Pattern Matching	

Extensions

Introduction

- ❖ The PHP execution environment includes the following three modules:
 - Core PHP
 - Zend Engine
 - Extension
- ❖ The Core PHP module handles the very basic operations such as requests, file streams, error handling and similar.

Introduction

- ❖ The Zend Engine module is responsible for converting human readable code into machine understandable op-codes. Once the conversion completes it executes the generated op-code on a virtual machine.
- ❖ The PHP Extensions are bunches of functions, classes and streams that are available to code we write in PHP. The extensions usually target specific tasks, such as the mysql extension that allows us to connect MySQL. and similar.

Introduction

- ❖ When the Zend Engine executes generated code that requires access to any of the extensions it passes the control to the required extension. Once that extension completes the required operation it transfer back the control.

The Available Extensions

- ❖ You can find a complete list of the available extensions at <http://php.net/manual/en/extensions.alphabetical.php>.

Extensions Development

- ❖ It is possible to develop our own new extension. Each extension should include a configuration file. Its name should be config.m4. In most cases the rest of the files will be in C.

Pear

- ❖ Pear is a PHP framework and distribution system for reusable PHP components.
- ❖ Since October 2011, 5 million lines of the PEAR codebase has shifted to github.

Zend Server Control Panel

- ❖ Using Zend Server we can browse its control panel at <http://localhost:10081>.
- ❖ We can find an easy to use interface for managing the extensions Zend server uses.



Zend Server Control Panel

