Zend_View_Dtl - Benjamin Eberlei

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

Zend Framework: Zend_View_Dtl Component Proposal

Proposed Component Name	Zend_View_Dtl
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_View_Dtl
Proposers	Benjamin Eberlei
Zend Liaison	TBD
Revision	08/04/2008 - Finished first proposal (wiki revision: 13)

Table of Contents

1. Overview

Zend_View_Dtl is an attempt to completly clone the Django Template Language Library (Django is a Python Web Framework) and integrate it as Zend_View_Interface implementation. It is an object oriented templating language that has some roots in Smarty syntax. It allows to inherit templates and overide specific blocks of the template. All Tags and filters should be ported so that there is full compability between both implementations. Additionally a helper will be suggested to to integrate the View with the DojoX DTL parser in the most simple way, to share templating logic between server and client.

This proposal attempts to bring a DTL Parser into the ZendX extras library as an additional alternative to Zend View, since the main component already has a template engine. Its in no way a proposal to replace Zend View. I have already implemented a first version for my personal Zend Framework extensions library, it can be loaded from: http://www.beberlei.de/calypso/

2. References

- Django Template Language: For template authors
- Django Template Language: For programmers
- DojoX DTL Parser

3. Component Requirements, Constraints, and Acceptance Criteria

- **implements** Zend_View_Interface
- **must** implement all functionality DTL and DojoX DTL offer
- **must** offer functionality to syntesize with DojoX DTL
- **should** integrate with Zend_Cache
- **should** allow easy integration of already exisiting View Helpers

4. Dependencies on Other Framework Components

- Zend_View_Interface
- Zend_View_Exception
- Zend_Json
- Zend_View_Helper_Abstract

5. Theory of Operation

Zend_View_Dtl is a proposed extension to the Zend Framework to allow the integration of a Template Engine into projects that are in need of non-php rendering of views. It implements one of the most state of the art template engines that has proven to be succesful in the Django framework. It uses object oriented patterns like inheritence to generate templates. Using this component would start with overriding your MVC application to use a Zend_View_Dtl object instead of the default Zend_View. View scripts will then be written using the django template language syntax as specified in the Django Template Language documentation.

No high hopes: Using DTL is slower than using the pure Zend_View, but it is convienient for projects that stricly seperate programming from webdesign (In Outsourcing relationsships for example), it offers very nice programming shortcuts that webdevlopers easily understand and its OO background makes using Zend_Layout obsolete. Under the hood it integrates with Zend_Cache to save the compiled Node List structure for faster page generation.

The neat thing to use will be the switches to shift templating from the server to the client using the DojoX DTL Parser, which implements the Django template language via Dojo on the client side. You would be able to specify any template script to render in the client and take data from a JSON string. The template engine would recognize the client side block, generate a javascript snippet to be included on the rendered server site page and on load the client will fetch the missing data from a specified url returning a JSON object. This would allow for a seamless and fast integration of dynamicly ajax generated content and traditional server side templating without any additional costs and complexity.

6. Milestones / Tasks

- Milestone 1: Finish First Running alpha version (clone python to php code) **DONE** (http://www.beberlei.de/calypso/)
- Milestone 2: Finish the proposal (Implement varity of Use Cases)
- Milestone 3: Community Review
- Milestone 4: Implement Helper that allows for DojoX and Zend_View_Dtl integration and build use cases and demo
- Milestone 5: Implement Unit-Tests
- Milestone 6: Ready for Review Phase / Find Bottlenecks in different enviroements and optimize Parsing and Rendering
- Milestone 7: Request for Recommendation
- Milestone 8: Documentation

7. Class Index

- Zend_View_Dtl
- Zend_View_Dtl_Exception
- Zend_View_Dtl_Template
- Zend_View_Dtl_Lexer
- Zend_View_Dtl_Token
- Zend_View_Dtl_Parser
- Zend_View_Dtl_Parser_FilterExpression
- Zend_View_Dtl_Parser_Variable

- Zend_View_Dtl_Parser_Tag_Interface (+ lots of implementations)
- Zend_View_Dtl_Node_Abstract (+ lots of implementations)
- Zend_View_Dtl_Node_Text
- Zend_View_Dtl_Node_Variable
- Zend_View_Dtl_Filter_Abstract (+ lots of implementations)

8. Use Cases