

Zend_Service_Payment

<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_Service_Payment Component Proposal

Proposed Component Name	Zend_Service_Payment
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Service_Payment
Proposers	Richard Thomas Darby Felton, Zend liaison
Revision	1.1 - 1 August 2006: Updated from community comments. (wiki revision: 11)

Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

1. Overview

Zend_Service_Payment is a module to provide a standardized api for payment processing. Due to the large range of differences between processors this only suggests a base api, it does not attempt to abstract all of the settings and complicated nature of processors.

2. References

- Basic Zend_Service_Authnet Class that the idea is based off

3. Component Requirements, Constraints, and Acceptance Criteria

Will provide a basic api layout for handling and processing of payments online

- This component **will** provide a basic api
- This component **will** provide a way to load various api's
- This component **will not** attempt to fully abstract all of the settings of the various processors

4. Dependencies on Other Framework Components

- Zend_Exception

5. Theory of Operation

Provide an easy to use basic api for handling payment processing

6. Milestones / Tasks

- Milestone 1: [design notes will be published here](#)
- Milestone 2: Working prototype checked into the incubator supporting use cases #1, #2, ...
- Milestone 3: Working prototype checked into the incubator supporting use cases #3 and #4.
- Milestone 4: Unit tests exist, work, and are checked into SVN.
- Milestone 5: Initial documentation exists.

If a milestone is already done, begin the description with "[DONE]", like this:

- Milestone #: [DONE] Unit tests ...

7. Class Index

- Zend_Service_Exception
- Zend_Service_Payment (factory class)
- Zend_Service_Payment_Authnet

8. Use Cases

```
$authnet = new Zend_Service_Payment('authnet');
$authnet->setParam('x_login','login');
$authnet->setParam('x_tran_key','key');
$authnet->setParam('x_card_num','4111111111111111');
$authnet->setParam('x_amount','1');
$authnet->setParam('x_exp_date','11/11');
$authnet->setParam('x_po_num','test1');
$authnet->setParam('x_ship_to_country','test2');
$authnet->setParam('x_card_code','321');
$authnet->process();
```

9. Class Skeletons

```
class Zend_Magic_Exception extends Zend_Exception {}  
  
class Zend_Magic {  
    ...  
}
```

]]</ac:plain-text-body></ac:macro>
]]</ac:plain-text-body></ac:macro>