Zend_Protobuff

<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\Protobuff Component Proposal

Proposed Component Name	Zend\Protobuff
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend\Protobuff
Proposers	Rostyslav Mykhajliw
Zend Liaison	https://github.com/necromant2005/zeus-proto
Revision	1.0 - 31 December 2010: Beta. (wiki revision: 7)

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\Protobuff is a simple component that provide encoding and decoding Google Protocol Buffers

2. References

- Protocol Buffers Overview
- pb4php

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** correctly reads .proto files
- The generated skeleton class **will** support Google Class API.
- This component **will** support encode data.
- This component **will** support decode data.
- This component **will** provide client through Google Protocol Buffers
- This component **will** provide server based on Google Protocol Buffers

4. Dependencies on Other Framework Components

- Zend\Exception

5. Theory of Operation

The component is instantiated with a mind-link that ...

6. Milestones / Tasks

- Milestone 1: [DONE] Create base abstract class
- Milestone 2: [DONE STAGE-1] Realize encoder Protocol buffers
- Milestone 3: [DONE STAGE-1]Realize decoder Protocol buffers
- Milestone 4: Add support .proto files specification
- Milestone 5: Add generation classes based on .proto file

7. Class Index

- Zend\Protobuff\AbstractProtobuff
- Zend\Protobuff\Encoder
- Zend\Protobuff\Decoder
- Zend\Protobuff\Code\Generator

8. Use Cases

UC-01

```
$encoder = new Zend\Protobuff\Encoder;
$encoder->encode(array(
   'name'=> array(
      'required'=>true,
      'type' => Zend\Protobuff\AbstractProtobuff::TYPE_STRING,
   ),
   'age'=> array(
      'required'=>false,
      'type' => Zend\Protobuff\AbstractProtobuff::TYPE_INT32,
      'default'=>0,
   ),
), array(
   'name'=>'peter',
   'age'=>25,
));
```

UC-02

```
$decoder = new Zend\Protobuff\Decoder;
$decoder->decode(array(
   'name'=> array(
      'required'=>true,
      'type' => Zend\Protobuff\AbstractProtobuff::TYPE_STRING,
   ),
   'age'=> array(
      'required'=>false,
      'type' => Zend\Protobuff\AbstractProtobuff::TYPE_INT32,
      'default'=>0,
   ),
), $buffer);
```

UC-01

```
$encoder = new Zend\Protobuff\Encoder;
$encoder->encode('path/to/user.proto', array(
   'name'=>'peter',
   'age'=>25,
));
```

UC-02

```
$decoder = new Zend\Protobuff\Decoder;
$decoder->decode('path/to/user.proto', $buffer);
```

9. Class Skeletons

```
abstract class Zend\Protobuff\AbstractProtobuff
{
   public function getWireTypeClass(array $options)
}
```

```
class Zend\Protobuff\Encoder extends \Zend\Protobuff\AbstractProtobuff
{
    public function encode($map, $values)
}
```

```
class Zend\Protobuff\Decoder extends \Zend\Protobuff\AbstractProtobuff
{
    public function decode($map, $buffer)
}
```

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