Zend_Validate_Phone - Thomas Weidner

<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_Validate_Phone Component Proposal

Proposed Component Name	Zend_Validate_Phone
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Validate_Phone
Proposers	Thomas Weidner
Zend Liaison	TBD
Revision	1.0 - 11 June 2009: Initial Draft. (wiki revision: 5)

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_Validate_Phone is a validator which checks if a given input conforms a expected format for a phone number.

2. References

- Wikipedia - Telephone numbers
- Wikipedia - E.123
- ITU-T

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** check if a string conforms a expected phone number format
- This component **will not** change the format, only check
- This component **will** accept national and international formats
- This component **will** also allow to check if a given number conforms a given country or multiple countries
- This component **will not** check if a given number does exist

4. Dependencies on Other Framework Components

- Zend_Validate_Exception
- Zend_Validate_Abstract

5. Theory of Operation

The component validates if a given string conforms a expected input. The expected format can be a national number, a international number, or a self defined format. It can also check if number parts are given. (country code, extension).

6. Milestones / Tasks

- Milestone 1: [DONE] Proposal finished
- Milestone 2: Proposal accepted
- Milestone 3: Working implementation
- Milestone 4: Unit tests
- Milestone 5: Documentation
- Milestone 6: Moved to core

7. Class Index

- Zend_Validate_Phone

8. Use Cases

UC-01

Simply validation... accepts all sorts of phone numbers

```
$validate = new Zend_Validate_Phone();
$validate->isValid('(123) 456789');
$validate->isValid('+49 (123) 456789');
$validate->isValid('0049 123 456789-123');
$validate->isValid('456789 - 123');
$validate->isValid('Phone: 456 - 132'); // FALSE
```

UC-02

Validation for one country

```
$validate = new Zend_Validate_Phone('de_AT');
$validate->isValid('(1234) 56789');
$validate->isValid('+49 (123) 456789'); // FALSE
$validate->isValid('0049 123 456789-123'); // FALSE
$validate->isValid('0043 123 456789-123');
$validate->isValid('456789 - 123');
$validate->isValid('Phone: 456 - 132'); // FALSE
```

UC-02

Validation with set options (no extension)

```
$validate = new Zend_Validate_Phone(array('locale' => 'de_AT', 'format' =>
Zend_Validate_Phone::NO_EXTENSION));
$validate->isValid('(1234) 56789');
$validate->isValid('+49 (123) 456789'); // FALSE
$validate->isValid('0049 123 456789-123'); // FALSE
$validate->isValid('0043 123 456789-123'); // FALSE because of extension
$validate->isValid('456789 - 123'); // FALSE
$validate->isValid('456789'); // FALSE
$validate->isValid('Phone: 456 - 132'); // FALSE
```

UC-03

Validation with set options (national format)

```
$validate = new Zend_Validate_Phone(array('locale' => 'de_AT', 'format' =>
Zend_Validate_Phone::NATIONAL));
$validate->isValid('(1234) 56789');
$validate->isValid('+49 (123) 456789'); // FALSE
$validate->isValid('0049 123 456789-123'); // FALSE
$validate->isValid('0043 123 456789-123'); // FALSE because of international format
$validate->isValid('456789 - 123');
$validate->isValid('Phone: 456 - 132'); // FALSE
```

UC-04

Validation with self defined format

```
$validate = new Zend_Validate_Phone(array('locale' => 'de_AT', 'format' => array('TCN' => 4, 'CCN' =>
3, 'LCN' => '5-7', 'EXT' => 0)));
```

The resulting number would be like
Territory code: 4 numbers
City code: 3 numbers
Local code: 5 to 7 numbers
Extension: 0 numbers

+49 1234 456789 -> false -> territory code 2 and not 4
0049 (1234) 456789 -> true
0049 (1234) 456789-123 -> false, extension given

9. Class Skeletons

```
class Zend_Validate_Phone extends Zend_Validate_Abstract {}

    const NATIONAL;
    const INTERNATIONAL;
    const NO_EXTENSION;

    public function __construct($parameters);
    public function getTerritory();
    public function setTerritory($territory);
    public function getFormat();
    public function setFormat($format);
    public function isValid($input);
}
```

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