Zend_Epp - Thomas Gelf

<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_Epp Component Proposal

Proposed Component Name	Zend_Epp
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Epp
Proposers	Thomas Gelf
Zend Liaison	TBD
Revision	1.0 - 5 July 2010: Initial Draft. (wiki revision: 12)

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

The Extensible Provisioning Protocol (EPP) is a flexible XML-based protocol allowing service providers to perform object-provisioning operations using a shared central object repository. As of today most top-level domain registries and most of the (sparse) ENUM-registries provide an EPP interface.

This library provides a registrar with everything it needs for it's communication with an EPP service (as an EPP client). It should be extensible in a way allowing it to be useful also for registries (EPP server).

2. References

- Wikipedia: Extensible Provisioning Protocol

Origin, sponsorship
Research and development for this component is being sponsored by my current employer, Raiffeisen OnLine. Being a well-known ISP in South Tyrol (Northern Italy), we believe in the power of Open Source Software. We are intensively using the Zend Framework in various project and therefore thought that this library could be a nice addition to this powerful community project. The EPP project has been started as part of our internal PHP library and is right now being rearchitectured to fit ZF requirements.

3. Component Requirements, Constraints, and Acceptance Criteria

- This library **will** try to strictly implement most (all?) of the following standards:
    - RFC 5730 (4930, 3730): Extensible Provisioning Protocol (EPP)
    - RFC 5731 (4931, 3731): EPP Domain Name Mapping
    - RFC 5732 (4932, 3732): EPP Host Mapping
    - RFC 5733 (4933, 3733): EPP Contact Mapping
    - RFC 3915: Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)
    - RFC 3735: Guidelines for Extending the Extensible Provisioning Protocol (EPP)
- This library **will** also support registry-specific extensions:
    - Some registries have proprietary EPP extensions
    - Once the base library is complete it should be attractive enough to get contributions for most/all known extensions please see report-rfc4930-4934 for an older overview)
- To make sure that extensibility as of RFC 3735 is given, a few extensions **will** be implemented together with the first version of Zend_Epp:
    - NIC.it is extending contact and domain objects
    - EURid did a lot of tricky extensions and also added new commands (sunrise period) - an attempt to implement them would make sure that the library is really as extensible as required by RFC 3735
- DNSSEC as of RFC 5910 (4310) **would** also be an interesting extension
- If there is any interest in ENUM-related extensions: I'd love to get some input, especially from people having test accounts (e.g. enum.at):
    - RFC 4414: E.164 Number Mapping for the Extensible Provisioning Protocol (EPP)
    - RFC 5076 (3761): ENUM Validation Information Mapping for the Extensible Provisioning Protocol
- This library will be able to support different transports and provide the most used ones (TCP, HTTP) with it's first version
    - RFC 5734 (4934, 3734): Extensible Provisioning Protocol (EPP) Transport over TCP
- The first version **will** allow to run an EPP client (registrar)
- A following version **will** also allow to run an EPP server (registry)
- Therefore class structure **will** already be done in a way to allow this library to be used in a server implementation
- This library **will not** provide any backend functions for persistent data storage
- It **will not** automagically handle all EPP-related jobs as it is nothing but a plain stupid EPP library
- This library **will** however provide some "shortcuts" for routine jobs such as message queue handling
- This library **will** take care of completely transparent EPP session handling
- This library **will** be able to handle multiple parallel EPP sessions, also when connecting to different registries

4. Dependencies on Other Framework Components

- Zend_Date
- Zend_Exception

5. Theory of Operation

TBD

6. Milestones / Tasks

Describe some intermediate state of this component in terms of design notes, additional material added to this page, and / code. Note any significant dependencies here, such as, "Milestone #3 can not be completed until feature Foo has been added to ZF component XYZ." Milestones will be required for acceptance of future proposals. They are not hard, and many times you will only need to think of the first three below.

- Milestone 1: Finish proposal and architectural description
- 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_Epp_Registrar
- Zend_Epp_Registry (abstract)
    - Zend_Epp_Registry_Eurid

        - Zend_Epp_Registry_NicIt
    - Zend_Epp_Object (abstract)
        - Zend_Epp_Object_Contact
        - Zend_Epp_Object_Domain
        - Zend_Epp_Object_Extension (abstract)
            - Zend_Epp_Object_Extension_ItNic_Contact
            - Zend_Epp_Object_Extension_ItNic_Domain
            - Zend_Epp_Object_Extension_Eurid_Contact
            - ...
    - Zend_Epp_Request
        - Zend_Epp_Hello
        - Zend_Epp_Command
            - Zend_Epp_Command_Check
            - Zend_Epp_Command_Create
            - Zend_Epp_Command_Delete
            - Zend_Epp_Command_Info
            - Zend_Epp_Command_Login
            - Zend_Epp_Command_Logout
            - Zend_Epp_Command_Poll
            - Zend_Epp_Command_Renew
            - Zend_Epp_Command_Transfer
            - Zend_Epp_Command_Update
            - Zend_Epp_Command_Extension (abstract)
        - Zend_Epp_Request_Extension (abstract)
    - Zend_Epp_Response
        - Zend_Epp_Greeting
        - Zend_Epp_Response_Extension (abstract)
    - Zend_Epp_Extension (abstract)
    - Zend_Epp_Transport (abstract)
        - Zend_Epp_Transport_Http
        - Zend_Epp_Transport_Tcp
    - Zend_Epp_Exception (there will possibly be different Exception types)

8. Use Cases

UC-01

... (see good use cases book)

9. Class Skeletons

```
class Zend_Epp_Exception extends Zend_Exception {}

class Zend_Epp {
...
```

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