EXTRAS package for the PHPXMLRPC library

Gaetano Giunta

1.0.0

Table of Contents

Introduction

This collection of utilities provides many additional features to the PHP classes that are part of the PHPXMLRPC library.

Although very useful for specific purposes, these files have been omitted from the base distribution in order to keep it small and focused.

PHPXMLRPC is a php library which implements the XML-RPC protocol.

XML-RPC is a format devised by Userland Software for achieving remote procedure call via XML using HTTP as the transport. XML-RPC has its own web site, xmlrpc.com

System requirements

The PHPXMLRPC library package, version 4.10.1 or later, is a prerequisite for the extras package.

The PHPJSONRPC library package, version 1.0.0 or later, is a prerequisite for the extras package to support json-rpc.

A working php setup is of course needed.

The library is tested to work with php versions ranging from 5.4 to 8.5. Please note that not all library features are available for every PHP version. The "curl" php extension is needed to support https calls. The "json" and "xmlrpc" php extensions do not interfere with the operations of either the base library or extras.

A complete testsuite has not yet been implemented for every component that is part of the extras. This prevents complete regression tests to be carried out against all php versions. If you encounter any quirk, especially with old php installs, please report it on the project development pages.

List of modules in the distribution

The 'extras' library is formed by separate components, grouped together for ease of download. The terms 'component' and 'module' are used interchangeably within this manual to indicate groups of files that provide a particular capability.

Included modules as of the current release are:

- AJAX: demo of an ajaxified version of the phpxmlrpc lib: supports executing xml-rpc/json-rpc calls directly from the client browser. Needs the jsxmlrpc lib.
- DOCXMLRPC: subclasses of xml-rpc and json-rpc servers that auto-generate HTML documentation of exposed services. Easy as a breeze to use, and extremely user-friendly (it is used on the main php-xmlrpc website, too).
- PROXY: subclass of xml-rpc server that can act as remote (transparent) xmlrpc proxy to forward calls to a remote server. Can either forward any received call or probe the remote server first for existing methods.

Ajax module

files

- axjaxdemo.php: demo file showcasing construction of a complete ajax client/server solution in a single php file, based on phpxmlrpc and phpjsonrpc
- JRpcServer.php, JSJsonRpcServer.php, JSWrapper.php: the classes which implement the required logic

usage

To be documented...

DocXmlRpc module

files

- docServer.php, docServerjsonRpc.php: demo files showcasing construction of a self-documenting server solution in a single php file, based on phpxmlrpc and phpjsonrpc
- SelfDocumentingServer.php, SelfDocumentingjsonrpcServer.php, ServerDocumentor.php, XmlrpcSmartyTemplate.php: contain the classes extending the servers with the capability to generate html documentation of exposed methods
- docxmlrpcs.css: the stylesheet used by default by the self-generated html documentation

usage

The purpose of this module is to let users have human-readable documentation automatically generated for all xml-rpc methods that are implemented by a given xml-rpc server.

The documentation produced will be in HTML format, and it will exactly match the information that the xml-rpc server class makes available to clients via usage of the system.methodHelp, system.methodSignature and system.listMethods xml-rpc calls. As an extra feature, documentation for single parameters of xml-rpc methods can be added. Html forms are optionally included with every method synopsis description page, to help the developer do quick'n'dirty debugging.

The simplest way to make usage of the extra capabilities of this module is to take an existing phpxmlrpc Server and swap the php class used with SelfDocumentingServer:

```
// define the dispatch map describing all the xml-rpc methods exposed by this server
and the php functions that implement them
$dmap = array(
  '' => array(),
  ...
);
```

```
// include the php code implementing the xml-rpc methods
...

// build the server and let it do its job: that's it!
$server = new SelfDocumentingServer($dmap);
```

Since version 0.4, the html forms that are generated by the server class can take advantage of the javascript-based visual xml-rpc value editor that is part of the jsxmlrpc library (downloadable as a separate package from its github project pages, or linkable directly from a js cdn), making it even easier to invoke the implemented webservices via a browser interface. This optional feature can be enabled by setting the `editorpath` member of the server:

```
$s = new SelfDocumentingServer($dmap, false);
$s->editorpath = '../javascript/'; // enable link to js visual editor of content: set
this to the directory where it is located
$s->setdebug(3); // enable maximum debugging level, just in case
$s->service();
```

Proxy module

files

- `proxyServer.php`: demo of a simple xml-rpc service acting as proxy to the webservices of tanoconsulting.com
- `ReverseProxy.php`: xml-rpc proxy server class definition

usage

To be documented...