Mimic

User's Guide

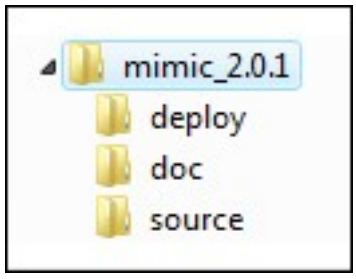
1. Introduction

Mimic is an open source XML-RPC client implemented in JavaScript. It was classified as client, because it is only able to produce requests and parse responses, so you cannot use it to parse requests and produce responses as required for a server implementation.

It is intended to be embedded inside web pages and enable them to access web services based on XML-RPC through the W3C XMLHttpRequest interface.

2. Package Content

When you extract the contents of the package **mimic_2.0.1.zip** you must see a structure like showed bellow:



Content	Description
deploy	Final deploy files.
docs	Manuals and documentation.
source	Source scripts.

3. Install

To install Mimic in your applications you only have to:

- create a folder named **mimic** in your application;
- extract **mimic_2.0.1.zip** contents to a folder outside your application structure;
- open the folder **deploy***;
- copy all contents to the **mimic** folder in your application;
- in any page you want to use mimic, insert in its header the markup:

<script src=”mimic/mimic.js”> </script>

* If you want to debug Mimic under development environment it is recommended to copy the contents inside the **source** folder to your application.

4. How to Use

It is very simple to use Mimic, you only have to produce a request and then process the response. To do it you only have to use two small classes.

The class XmlRpcRequest is used to define the remote method that you want to call, and the parameters you want to pass:

Member	Description
`serviceUrl`	Defines the server URL where the remote method exists.
`methodName`	Define the name of the remote method to call.
`params`	An array that holds all method parameters.
`addParam`	Method used to add a new parameter to the request.
`clearParams`	Method used to clear params.
`send`	Post your request and return a XmlRpcResponse object.
`marshal`	Method used internally to marshal JavaScript data.

It is important to note that you can pass virtually any object do addParam, including instances of the Core JavaScript classes Array, Boolean, Date and String; instances of your custom classes that implicit inherit from the Object class; and instances of Base64 class provided by Mimic.

The second class XmlRpcResponse is used to retrieve the server-side response and transform it back to JavaScript:

Member	Description
`xmlData`	Holds the XML document passed back from server.
`faultValue`	Indicate if the response is a fault.
`currentIsName`	Used internally to check possible property parameter.
`propertyName`	Used internally to hold possible properties' name
`params`	Array that holds JavaScript objects parsed form xmlData
`parseXML`	Parse the xmlData to JavaScript and store it on params
`unmarshal`	Used internally to unmarshal XML-RPC data.