

jmAMF 1.0 Documentation

2009-2010 Jérôme Vieilledent

Introduction

This extension allows you to develop and use services with [AMF protocol](#) to communicate with Flash/Flex applications with eZ Publish.

AMF protocol supports primitive types such as **boolean**, **string**, **integer**, **array**, **object**, **typed objects**... It is a binary protocol directly implemented in the core of Flash Player, which means that there is no parsing (contrary to REST XML or SOAP services). Transfers are much faster (no unneeded information is transmitted) and you can even gzip it (with Apache mod_deflate for instance) ! Besides, it is Open Source ;-).

It uses [Zend_AMF library](#) , which is officially sponsored by Adobe. It is compatible with **eZ Publish 4.x**.

Please note that Zend_AMF requires PHP 5.2.4 or later version

Installation

1. Download the compressed file under /extension directory and uncompress it.
2. Activate the extension.
3. Run the bin/php/ezpgenerateautoloads.php script for the extensions :

```
php bin/php/ezpgenerateautoloads.php -e
```

4. Clear the caches :

```
php bin/php/ezcache.php --clear-all --purge
```

Configuration and usage

Flex configuration

Your *RemoteObject* endpoint must be *http://your-ezp-host-domaine/jvamf/endpoint* You can edit your endpoint either in your Flex application's **services-config.xml** file or directly in your `<mx:RemoteObject/>` tag. If you use the services-config.xml file (recommended), your endpoint should be like this : `http://{server.name}:{server.port}/jvamf/endpoint`.

Here's an example of a channel definition in a services-config.xml file :

```
<channel-definition id="espacepresse-amf" class="mx.messaging.channels.AMFChannel">
  <endpoint url="http://{server.name}:{server.port}/jvamf/endpoint" class="flex.messaging.endpoints.AMFEndpoint"/>
</channel-definition>
```

For more information to configure your Flex application to use your AMF services, you should read this : <http://framework.zend.com/manual/en/zend.amf.server.html#zend.amf.server.flex>

eZ Publish configuration

Take a look at the INI file *extension/jvamf/settings/jvamf.ini* (DO NOT edit the file, make an override of it !).

- You can specify directories in which *Zend_AMF* will look for service classes, depending on your *RemoteObject* call in your Flex application. Note that the PHP class that will be used for the service

correspond to the **source** attribute of your `<mx:RemoteObject/>` tag.

For example, with this Flex RemoteObject :

```
<mx:RemoteObject id="myRemoteObject" showBusyCursor="true" destination="mydestination" source="MyAMFService" />
```

When contacting the server, `javaMF` will look for a **service class** named **MyAMFService**. This must be accessible to `Zend_AMF`. You can specify one or several directories to scan in `javmf.ini`. So if your class is in `extension/myextension/classes/services/MyAMFService.php` (the name of the file is case sensitive), you should add in an override of `javmf.ini` :

```
[AMF]
ServicesDir[]=extension/myextension/classes/services
```

- You can explicitly declare Class Mapping from PHP => ActionScript. Note that you can also declare this mapping in your Value Objects (VO) classes. For more info, read the comments in `javmf.ini` and <http://framework.zend.com/manual/en/zend.amf.server.html#zend.amf.server.typedobjects>
- You can switch to `Zend_AMF Production Mode`, which means that no details will be sent to the client when throwing exceptions. <http://framework.zend.com/manual/en/zend.amf.server.html#zend.amf.server.errors>

Objects and Exceptions

You can use several native types and objects as services arguments. For example, you can transfer ActionScript **Date objects**; they will be converted to `Zend_Date` objects in PHP. Also, each time you throw an exception, the Flex RemoteObject will raise an **onError** event with the details of your exception (except if the *Production mode* has been enabled, in this case no detail will be provided).

Below ActionScript to PHP mapping :

ActionScript	PHP
undefined	null
null	null
int	integer
Number	float
Boolean	boolean
String	string
Array	array
XML	DomDocument
flash.utils.ByteArray	string
uint	float
Object	object
RemoteClass Object	class mapped object
Date	Zend_Date
ArrayCollection	object

Debugging your services

To debug your services, you can use [Charles Debugging Proxy](#) - Please note that this software IS NOT Open Source. Charles fully supports AMF protocol and allows you to visualize the backend results of your service call.