



Flash és PHP kommunikáció

Web Konferencia 2007

Ferencz Tamás

Jasmin Media Group Kft

A lehetőségek

- FlashVars
- External Interface
- Loadvars
- XML
- SOAP
- Socket
- AMF
- AMFphp
- PHPObject

Flash Vars

- Flash verziótól függetlenül működik
- A flash induláskor megkapja ezeket az adatokat
- Egyoldalú kommunikáció
- Kétszer kell megadni (embed / object)
- A paraméterek száma a string hosszúság miatt véges.

External Interface

- Flash és Javascript kommunikáció
- Flashból Javascript hívás
- Flash függvények kiajánlása Javascript számára.
- Offline alkalmazási területen a flash és a projector közötti kommunikáció. Az FSCommand kiváltása.
- Flash 8 – as verziótól elérhető

[Adobe Flash - External Interface](#) / [Javascript Ellenőrzés Flashből](#)

Loadvars

- Két irányú kommunikáció
- Szabad adatszerkezet
- Szabványos HTTP kapcsolat (Post / Get)
- Saját magunknak kell feldolgozót írni az adatszerkezetünkhöz
- Sok adat átvitelekor a feldolgozás jelentős időt vesz igénybe
- A böngészőn megy át az adatforgalom a cache problémákat nekünk kell kezelni

[Adobe Loadvars Tutorial](#)

XML Load

- Az érkezett adat már xml-ként feldolgozható
- Get paramétert lehet küldeni a szervernek (file névhez hozzáfűzve adhatod meg az urlencodolt változókat)
- A böngészőn megy át az adatforgalom a cache problémákat nekünk kell kezelni.
- Az átvitt adat mennyisége az XML szerkezete miatt hajlamos nagyon megnőni

SOAP

- Standard kommunikációs protokoll
- Rengetek külső függvény könyvtár áll rendelkezésre
- Az eredmények gyors feldolgozása
- XML + SOAP overhead => Nagy adatforgalom

[Mi a SOAP ?](#) [Soap Alapok](#)

Socket

- Kliens-Szerver kapcsolat
- Eredetileg XML-hez írták, de bármilyen formátumot képes kezelni.
- „Real Time” kommunikáció
- Flash9 óta bináris kommunikációs lehetőség is megjelent.
Felhasználható pl: Ftp, irc, imap protokollok közvetlen kezelésére.
- A nem szabványos portok sok helyen tiltva vannak.
- Egyedi szerver alkalmazást igényel

AMF

- HTTP alapú bináris formátum.
- Kis méret
- Flash oldalról natív támogatás.
- PHP modul a nagyobb sebességért
- HTTP_POST_DATA elérhetőség kell

[Open Flash - AMF documentation](#) [AMF - Adat formátumok Flash oldal](#)

AMFPHP

- Teljesen Objektum Orientált
- Teljes AMF2 –es protokoll támogatás
- Kis overhead a serializációs folyamatokon
- AMF php egy RPC réteg a php -hoz mely képes kommunikálni:
 - Flash és Flex alkalmazásokkal Remoting
 - JavaScript és Ajax alkalmazásokkal JSON
 - XML kliensekkel XML-RPC

Belső folyamatok

- Unserializálja a bejövő kérést
- Megkeresi a válaszáért felelős osztályt
- Initializálja az osztályt
- Biztonsági ellenőrzést végez
- Az osztályban meghívja a megfelelő függvényt az adott paraméterekkel
- Serializálja a függvénytől vissza térő adatokat

Belső felépítés

- Amf-core : Az amf file-jai
- Browser: egy webes tesztfelület
- Services: ide rakjuk a saját file-jainkat.
- A services könyvtárban user.php
- User.php –ban user class annak az „constructor” függvénye a function user()
- A user function tartalmazza a „method table” leíró részt.

Felhasználó ellenőrzés – PHP

```
1 <?php
2 class user
3 {
4     function user()
5     {
6         $this->methodTable = array
7         (
8             "check_nick" => array
9             (
10                 "description" => "Nev foglaltsaganak ellenorzese",
11                 "arguments" => array("data - array data {brandid:,sessionid:,nick:}"),
12                 "access" => "remote"
13             )
14         );
15     }
16     /**
17      * Nev foglaltsaganak ellenorzese
18      * @access remote
19      * @param array $data {brandid:,sessionid:,nick:}
20      * @return array {sessionid:,nick:,valid:,err_desc:,err_text:}
21      */
22     function check_nick($data)
23     {
24         ## ellenörzés
25         return true;
26     }
27 }
28 ?>
```

Method Table

- Description: Szabad szöveg, nem kötelező de nagyon ajánlott
- Arguments: Ha a class mapping biztonsági intézkedés be van kapcsolva akkor kötelező, ha nincs akkor csak ajánlott.
- Access: kötelező, ez dönti el, hogy az adott függvényhez hozzá lehet –e férni kívülről. Ha netán mégis elmarad, alap értelmezés szerint minden függvény private besorolású

[AMFphp - Documentáció - Method Table](#)

A flash oldal

```
1 package
2 {
3     import flash.net.Responder;
4     import flash.net.NetConnection;
5     import flash.net.ObjectEncoding;
6     import flash.display.Sprite;
7     public class GWTest extends Sprite
8     {
9         private const amfURL:String = "http://localhost/gateway.php";
10        private var amfConnection:NetConnection;
11        private var responder:Responder;
12        public function GWTest ( )
13        {
14            amfConnection = new NetConnection( );
15            amfConnection.objectEncoding = ObjectEncoding.AMFO; // setting
16            amfConnection.connect( amfURL );
17            responder = new Responder( onResult , onStatus );
18            var message:Object = { };
19            message.brandid = "demo";
20            message.sessionid = "1234567890";
21            message.nick = "membernev";
22            amfConnection.call( "user.check_nick" , responder , message );
23        }
24        public function onResult ( result:Object ):void
25        {
26            for ( var a:* in result ) trace( a + " : " + result[a] );
27        }
28        public function onStatus ( status:Object ):void
29        {
30            for ( var a:* in status ) trace( a + " : " + status[a] );
31        }
32    }
33 }
```

Fejlesztő támogatás

- Service browser egy webes alkalmazás ahol böngészőből lehet tesztelni a frissen megírt kódot. + Segít a flash oldali AS2-es kódok legenerálásában.
- Debug Gateway: ahol minden hiba naplófile-ba kerül. Curl szükséges hozzá.
- Packet logolás, minden bejövő, és kimenő adat file-ba mentése.

Tipikus hibák

- Ha Flash Media Server használja az gateway-t előfordulhat, hogy egy HTTP kérésben több AMF kérés érkezik. A core ezt gond nélkül kezeli, az általunk írt függvényekben, file-okban kell oda figyelni az include-okra, deklarációkra.
- Az AMF bináris formátum, ezért kifejezetten kényes minden hibaüzenetre amit tisztán kiírunk.
- Mindig kell valami választ adni. Egyes 8-as flash verziók kifagynak, ha nem kapnak vissza semmilyen választ.

A jövő

- Jelenleg az 1.2.5 –ös verzió a stabil kiadás
- Developer verzió az 1.9 –es
- Készül a C-ben irt php kiterjesztés a serializáció / unserializáció meggyorsítására. (50-200x sebesség növekedés)
- Jobb Json támogatás
- Jobb autentikációs megoldások
- Methodtable eltűnése
- Teljesen flex-ben irt Service Browser (egységesített browser több flash remoting project között)

Kapcsolódó linkek

- [AMF php website](#)
- [AMF php 1.9 Beta](#)
- [AMF php 1.9 Beta 2](#)

PHPObject

- Saját Kommunikációs osztály
- Kisebb átvitt adat mint az XML loadnál
- Csak PHP és Flash közötti kommunikációra.
- Az átvihető adatok típusa korlátozott

Belső felépítés

- PHP oldalon a definiált services könyvtárban létre kell hozni a class-al megegyező nevű php file-t ami csak az adott class -t tartalmazza. Ezen belül lesz meghívva az adott függvény az átadott paraméterekkel.
- Flash oldalon be kell tölteni a projecthez járó .as file-t.

[PHPObject - Dokumentáció](#)

PHPObject - Kód

- Flash forrás

```
1 myFoo = new PHPObject("Foo");
2 myFoo.getResult_onResult = function(result)
3 {
4     trace(this.getOutput());
5 }
6 myFoo.getResult("Einstein");
```

- PHP Forrás

```
30 <?php
31 class Foo
32 {
33     function getResult($name)
34     {
35         return "$name, you are a genius!";
36     }
37 }
38 ?>
```


