

Rails + Flash alapú technológiák

Varga Péter
(pepusz.hu)

Miről lesz szó?

- Egyszerű URL paraméter átadás
- XML alapú kommunikáció
- RubyAmf

Egyszerű URL paraméter átadás

Egyszerű URL paraméter átadás

- Miért jó ez?

Egyszerű URL paraméter átadás

- Miért jó ez?
 - egyszerű megérteni

Egyszerű URL paraméter átadás

- Miért jó ez?
 - egyszerű megérteni
- Miért nem jó?

Egyszerű URL paraméter átadás

- Miért jó ez?
 - egyszerű megérteni
- Miért nem jó?
 - nem éppen nevezhető rugalmasnak

Paraméter átadás a gyakorlatban

Paraméter átadás a gyakorlatban

Rails

Paraméter átadás a gyakorlatban

Rails

```
1 | class UrlParamsController < ApplicationController
2 |   def helloworld
3 |     render :text=>"Hello world budapest.rb #{params[:name]}"
4 |   end
5 |
6 | end
7 |
```

Paraméter átadás a gyakorlatban

Rails

```
1 | class UrlParamsController < ApplicationController
2 |   def helloworld
3 |     render :text=>"Hello world budapest.rb #{params[:name]}"
4 |   end
5 |
6 | end
7 |
```

Flex

Paraméter átadás a gyakorlatban

Rails

```
1 class UrlParamsController < ApplicationController
2   def helloworld
3     render :text=>"Hello world budapest.rb #{params[:name]}"
4   end
5
6 end
7
```

Flex

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
3   <mx:Script>
4     <![CDATA[
5       import mx.rpc.http.HTTPService;
6       private var hwS:HTTPService = new HTTPService;
7       private function sendBtClicked(e:MouseEvent):void{
8         hwS.url = "http://localhost:3000/url_params/helloworld?name="+ nameField.text;
9         hwS.send();
10      }
11    ]]>
12 </mx:Script>
13 <mx:VBox x="10" y="10" width="583" height="589">
14   <mx:Button id="sendBt" label="Küldés" click="{sendBtClicked(event)}"/>
15   <mx:Label text="Név:" />
16   <mx:TextInput id="nameField"/>
17   <mx:TextArea id="textBoxTa" width="263" height="247" text="{hwS.lastResult}"/>
18 </mx:VBox>
19 </mx:Application>
20
```

Paraméter átadás a gyakorlatban

Demo

Rails

```
1 class UrlParamsController < ApplicationController
2   def helloworld
3     render :text=>"Hello world budapest.rb #{params[:name]}"
4   end
5
6 end
7
```

Flex

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
3   <mx:Script>
4     <![CDATA[
5       import mx.rpc.http.HTTPService;
6       private var hwS:HTTPService = new HTTPService;
7       private function sendBtClicked(e:MouseEvent):void{
8         hwS.url = "http://localhost:3000/url_params/helloworld?name="+ nameField.text;
9         hwS.send();
10      }
11    ]]>
12 </mx:Script>
13 <mx:VBox x="10" y="10" width="583" height="589">
14   <mx:Button id="sendBt" label="Küldés" click="{sendBtClicked(event)}"/>
15   <mx:Label text="Név:" />
16   <mx:TextInput id="nameField"/>
17   <mx:TextArea id="textBoxTa" width="263" height="247" text="{hwS.lastResult}"/>
18 </mx:VBox>
19 </mx:Application>
20
```

XML alapú kommunikáció

XML alapú kommunikáció

- Miért jó?

XML alapú kommunikáció

- Miért jó?
 - Minden ami miatt szeretjük az xmlt

XML alapú kommunikáció

- Miért jó?
 - Minden ami miatt szeretjük az xmlt
 - szabványos

XML alapú kommunikáció

- Miért jó?
 - Minden ami miatt szeretjük az xmlt
 - szabványos
 - strukturált

XML alapú kommunikáció

- Miért jó?
 - Minden ami miatt szeretjük az xmlt
 - szabványos
 - strukturált
 - stb

- Miért lehet “rossz”?
 - Sok “felesleges” tartalom miatt nagy fájl méret
 - Hálózati overhead megnőhet
 - Parsolni kell

XML...

példa

XML...

példa

- Lépések:

XML...

példa

- Lépések:
 - Készítünk egy VO-t ActionScriptben (Value Object vagy más néven DTO Data Transfer Object)

XML...

példa

- Lépések:
 - Készítünk egy VO-t ActionScriptben (Value Object vagy más néven DTO Data Transfer Object)
 - Ez az XML alapú kommunikációnál nem szükséges de ajánlott

XML...

példa

- Lépések:
 - Készítünk egy VO-t ActionScriptben (Value Object vagy más néven DTO Data Transfer Object)
 - Ez az XML alapú kommunikációnál nem szükséges de ajánlott
 - VO-ból XML

XML...

példa

- Lépések:
 - Készítünk egy VO-t ActionScriptben (Value Object vagy más néven DTO Data Transfer Object)
 - Ez az XML alapú kommunikációnál nem szükséges de ajánlott
 - VO-ból XML
 - XML elküld, feldolgoz

XML...

példa

- Lépések:
 - Készítünk egy VO-t ActionScriptben (Value Object vagy más néven DTO Data Transfer Object)
 - Ez az XML alapú kommunikációnál nem szükséges de ajánlott
 - VO-ból XML
 - XML elküld, feldolgoz
 - Válasz vissza

Flex oldala

- VO

```
1 package V0
2 {
3     [Bindable]
4     public class UserVO
5     {
6         public var username:String;
7         public var password:String;
8         public var email:String;
9     }
10 }
11 }
```

- SignUp UI

```
32 <mx:HTTPService id="signUp" url="http://localhost:3000/xml_alapu/create_user"
33                 contentType="application/xml" resultFormat="e4x" method="POST"
34                 result="handleSignUpResult(event)" />
35
36 <mx:Form x="10" y="10" width="100%" height="100%">
37     <mx:FormItem label="Felhasználó név:">
38         <mx:TextInput id="userNameTi"/>
39     </mx:FormItem>
40     <mx:FormItem label="Jelszó:">
41         <mx:TextInput displayAsPassword="true" id="passwordTi"/>
42     </mx:FormItem>
43     <mx:FormItem label="Email:">
44         <mx:TextInput id="emailTi"/>
45     </mx:FormItem>
46     <mx:FormItem>
47         <mx:Button id="saveBt" label="Mentés" click="sendSignUpRequest()" />
48     </mx:FormItem>
49 </mx:Form>
50 </mx:Canvas>
51
```

Flex oldala

- Scripts

```
3 <mx:Script>
4   <![CDATA[
5     import mx.controls.Alert;
6     import VO.UserVO;
7     import mx.rpc.events.ResultEvent;
8
9     private function sendSignUpRequest():void{
10      var user:UserVO = new UserVO;
11      user.username = userNameTi.text;
12      user.password = passwordTi.text;
13      user.email = emailTi.text;
14      var sendXml:XML = <user>
15                                     <username>{user.username}</username>
16                                     <password>{user.password}</password>
17                                     <email>{user.email}</email>
18      </user>;
19      signUp.request = sendXml;
20      signUp.send();
21    }
22    private function handleSignUpResult(e:ResultEvent):void{
23      var result:Object = e.result;
24      if(result=="error"){
25        Alert.show("Nem sikerült menteni", "Error");
26      }else{
27        Alert.show("jeppe");
28      }
29    }
30  ]]>
31 </mx:Script>
```

Rails oldala

```
1 class XmlAlapuController < ApplicationController
2   def create_user
3     @user = Users.new(params[:user])
4     @user.save
5     render :text=>"saved"
6   rescue ActiveRecord::RecordInvalid
7     render :text=>"error"
8   end
9
10 end
11
```

Demo

RubyAmf

RubyAmf

- Mi is az a AMF?

RubyAmf

- Mi is az a AMF?
 - Action Message Format

RubyAmf

- Mi is az a AMF?
 - Action Message Format
 - HTTP réteg fölött bináris típus

RubyAmf

- Mi is az a AMF?
 - Action Message Format
 - HTTP réteg fölött bináris típus
 - Flash oldalon natív támogatás

- Miért jó?

- Miért jó?
 - Mivel bináris ezért szerializált

- Miért jó?
 - Mivel bináris ezért szerializált
 - A leggyorsabb adatküldést teszi lehetővé a szerver és a kliens között

- Miért jó?
 - Mivel bináris ezért szerializált
 - A leggyorsabb adatküldést teszi lehetővé a szerver és a kliens között
 - AMF objektumok tömörített formában kerülnek küldésre (zlib)

- Miért jó?
 - Mivel bináris ezért szerializált
 - A leggyorsabb adatküldést teszi lehetővé a szerver és a kliens között
 - AMF objektumok tömörített formában kerülnek küldésre (zlib)
 - nem kell parsolnunk, mivel ActionScript objektumokat küldünk bytecodeként

- Miért jó?
 - Mivel bináris ezért szerializált
 - A leggyorsabb adatküldést teszi lehetővé a szerver és a kliens között
 - AMF objektumok tömörített formában kerülnek küldésre (zlib)
 - nem kell parsolnunk, mivel ActionScript objektumokat küldünk bytecodeként
 - nincs külön overhead a felesleges header és egyéb címkék miatt, mint a SOAP vagy egyéb XML alapú technológiáknál

- Miért jó?
 - Mivel bináris ezért szerializált
 - A leggyorsabb adatküldést teszi lehetővé a szerver és a kliens között
 - AMF objektumok tömörített formában kerülnek küldésre (zlib)
 - nem kell parsolnunk, mivel ActionScript objektumokat küldünk bytecodeként
 - nincs külön overhead a felesleges header és egyéb címkék miatt, mint a SOAP vagy egyéb XML alapú technológiáknál
 - aszinkronként is tudjuk kezelni

RubyAmf install

script/plugin install <http://rubyamf.googlecode.com/svn/tags/current/rubyamf/>

RubyAmf install

script/plugin install <http://rubyamf.googlecode.com/svn/tags/current/rubyamf/>

- Új fájl: app/controllers/rubyamf_controller.rb

RubyAmf install

script/plugin install <http://rubyamf.googlecode.com/svn/tags/current/rubyamf/>

- Új fájl: app/controllers/rubyamf_controller.rb
- Új fájl: config/rubyamf_config.rb

RubyAmf install

script/plugin install <http://rubyamf.googlecode.com/svn/tags/current/rubyamf/>

- Új fájl: app/controllers/rubyamf_controller.rb
- Új fájl: config/rubyamf_config.rb
- config/initializers/mime_types.rb <<

RubyAmf install

script/plugin install <http://rubyamf.googlecode.com/svn/tags/current/rubyamf/>

- Új fájl: app/controllers/rubyamf_controller.rb
- Új fájl: config/rubyamf_config.rb
- config/initializers/mime_types.rb <<
 Mime::Type.register “application/x-amf”, :amf

RubyAmf install

script/plugin install <http://rubyamf.googlecode.com/svn/tags/current/rubyamf/>

- Új fájl: app/controllers/rubyamf_controller.rb
- Új fájl: config/rubyamf_config.rb
- config/initializers/mime_types.rb <<
 Mime::Type.register “application/x-amf”, :amf
- config/routes.rb <<
 ActionController::Routing::Routes.draw do |map|
 map.rubyamf_gateway ‘rubyamf_gateway’,
 :controller=>‘rubyamf’, :action=>‘gateway’
 end

RubyAmf config

```
ClassMappings.translate_case = false
```

```
public var created_at:Date;
```

```
ClassMappings.translate_case = true
```

```
public var createdAt:Date; //created_at in rails
```

RubyAmf config

```
ClassMappings.assume_types = true
```

rails:

```
class Users < ActiveRecord::Base
```

flex:

```
[RemoteClass(alias="Users")]
```

```
public class Users{
```

RubyAmf config

```
ClassMappings.assume_types = false
```

rails:

```
class Users < ActiveRecord::Base
```

flex:

```
[RemoteClass(alias="Users")]
```

```
public class Users{
```

```
ClassMappings.register(  
:actionscrip => 'Users',
```

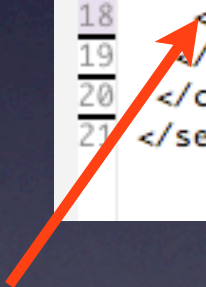
```
:ruby => 'Users',
```

```
:type => 'active_record',
```

```
:attributes => ["id", "username", "password", "email"])
```

services_config.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <services-config>
3   <services>
4     <service id="rubyamf-flashremoting-service" class="flex.messaging.services.RemotingService"
5     messageTypes="flex.messaging.messages.RemotingMessage">
6       <destination id="rubyamf">
7         <channels>
8           <channel ref="rubyamf"/>
9         </channels>
10        <properties>
11          <source>*</source>
12        </properties>
13      </destination>
14    </service>
15  </services>
16  <channels>
17    <channel-definition id="rubyamf" class="mx.messaging.channels.AMFChannel">
18      <endpoint uri="http://localhost:3000/rubyamf/gateway/" class="flex.messaging.endpoints.AMFEndpoint"/>
19    </channel-definition>
20  </channels>
21 </services-config>
```



services_config.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <services-config>
3   <services>
4     <service id="rubyamf-flashremoting-service" class="flex.messaging.services.RemotingService"
5     messageTypes="flex.messaging.messages.RemotingMessage">
6       <destination id="rubyamf">
7         <channels>
8           <channel ref="rubyamf"/>
9         </channels>
10        <properties>
11          <source>*</source>
12        </properties>
13      </destination>
14    </service>
15  </services>
16  <channels>
17    <channel-definition id="rubyamf" class="mx.messaging.channels.AMFChannel">
18      <endpoint uri="http://localhost:3000/rubyamf/gateway/" class="flex.messaging.endpoints.AMFEndpoint"/>
19    </channel-definition>
20  </channels>
21 </services-config>
```

Buildbe: -services "services-config.xml"

Kommunikáció

```
<mx:RemoteObject id="amfAlapuCom" destination="rubyamf" endpoint="http://localhost:3000/rubyamf/gateway/"  
source="AmfAlapuController" showBusyCursor="true" />
```

Kommunikáció

```
<mx:RemoteObject id="amfAlapuCom" destination="rubyamf" endpoint="http://localhost:3000/rubyamf/gateway/"  
source="AmfAlapuController" showBusyCursor="true" />
```

```
[RemoteClass(alias="V0.User")]
```

Kommunikáció

```
<mx:RemoteObject id="amfAlapuCom" destination="rubyamf" endpoint="http://localhost:3000/rubyamf/gateway/"  
source="AmfAlapuController" showBusyCursor="true" />
```

```
[RemoteClass(alias="V0.User")]
```

```
10     private function sendSignUpRequest():void{  
11         var user:User= new User;  
12         user.username =userNameTi.text;  
13         user.password = passwordTi.text;  
14         user.email = emailTi.text;  
15  
16         amfAlapuCom.create_user({user:user});  
17  
18     }
```

Kommunikáció

```
<mx:RemoteObject id="amfAlapuCom" destination="rubyamf" endpoint="http://localhost:3000/rubyamf/gateway/"  
source="AmfAlapuController" showBusyCursor="true" />
```

```
[RemoteClass(alias="VO.User")]
```

```
10     private function sendSignUpRequest():void{  
11         var user:User= new User;  
12         user.username =userNameTi.text;  
13         user.password = passwordTi.text;  
14         user.email = emailTi.text;  
15  
16         amfAlapuCom.create_user({user:user});  
17  
18     }
```

```
<mx:method name="create_user" result="handleSignUpResult(event)"/>
```

Params[what?]

Params[what?]

- alapértelmezettként, egy tömbben kerülnek átadásra az adatok pl `params[0]`

Params[what?]

- alapértelmezettként, egy tömbben kerülnek átadásra az adatok pl `params[0]`
- ha a `ParameterMappings.scaffolding = true`

Params[what?]

- alapértelmezettként, egy tömbben kerülnek átadásra az adatok pl `params[0]`
- ha a `ParameterMappings.scaffolding = true`
 - akkor ASben egy hasként megadhatjuk a paraméterek nevét, mint a példában is.

Params[what?]

- alapértelmezettként, egy tömbben kerülnek átadásra az adatok pl `params[0]`
- ha a `ParameterMappings.scaffolding = true`
 - akkor ASben egy `has`ként megadhatjuk a paraméterek nevét, mint a példában is.
- regisztrálhatunk konkrét elnevezéseket

Params[what?]

- alapértelmezettként, egy tömbben kerülnek átadásra az adatok pl `params[0]`
- ha a `ParameterMappings.scaffolding = true`
 - akkor ASben egy hasként megadhatjuk a paraméterek nevét, mint a példában is.
- regisztrálhatunk konkrét elnevezéseket
 - `ParameterMappings.register(:controller => :amf_alapu, :action => :create_user, :params => { :user => "[0]['user']" })`

Kiegészítések

Kiegészítések

- `is_amf?`

Kiegészítések

- `is_amf?`
- `FaultObject.new`

Kiegészítések

- `is_amf?`
- `FaultObject.new`
- `scaffolding (generate)`

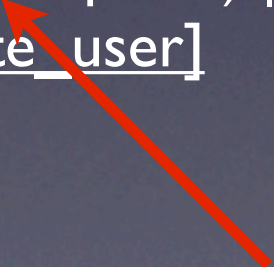
Kiegészítések

- `is_amf?`
- `FaultObject.new`
- `scaffolding (generate)`
- `session, setCredentials`

Teljesítmény

- xml

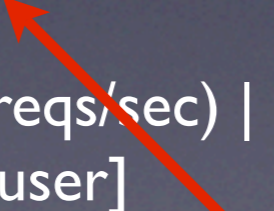
```
Processing XmlAlapuController#create_user (for 127.0.0.1 at 2009-02-23 17:05:40) [POST]
  Session ID: BAh7BiKZmxhc2hJQzonQWN0aW9uQ29udHJvbGxlcjo6Rmxhc2g6OkZsYXNo
SGFzaHsABjoKQHVzZWV7AA==--2bd730ceef283b07063ccf50028a2a41392011f3
  Parameters: {"user"=>{"username"=>"asd", "password"=>"asd", "email"=>"asd"},
"action"=>"create_user", "controller"=>"xml_alapu"}
  User Columns (0.149886) SHOW FIELDS FROM `users`
  SQL (0.000473) BEGIN
  User Create (0.000253) INSERT INTO `users` (`updated_at`, `username`, `password`, `email`,
`created_at`) VALUES('2009-02-23 16:05:40', 'asd', 'asd', 'asd', '2009-02-23 16:05:40')
  SQL (0.111076) COMMIT
Completed in 0.27093 (3 reqs/sec) | Rendering: 0.00034 (0%) | DB: 0.26169 (96%) | 200 OK [http://localhost/xml\_alapu/create\_user]
```



Teljesítmény

- Amf

```
Processing AmfAlapuController#create_user (for 127.0.0.1 at 2009-02-23 17:05:52) [POST]
  Session ID: BAh7BiKZmxhc2hJQzonQWN0aW9uQ29udHJvbGxlcjo6Rmxhc2g6OkZsYXNo
SGFzaHsABjoKQHVzZWV7AA==--2bd730ceef283b07063ccf50028a2a41392011f3
  Parameters: {"user"=>{"username"=>"asd", "password"=>"ads", "email"=>"asd"},
0=>{"user"=>{"username"=>"asd", "password"=>"ads", "email"=>"asd"}}, "action"=>"create_user",
"controller"=>"amf_alapu"}
  User Columns (0.002416) SHOW FIELDS FROM `users`
  SQL (0.001513) BEGIN
  User Create (0.000635) INSERT INTO `users` (`updated_at`, `username`, `password`, `email`,
`created_at`) VALUES('2009-02-23 16:05:52', 'asd', 'ads', 'asd', '2009-02-23 16:05:52')
  SQL (0.000719) COMMIT
Completed in 0.01405 (71 reqs/sec) | DB: 0.00528 (37%) | [http://localhostamf\_alapu/create\_user]
Sending data
Completed in 0.02217 (45 reqs/sec) | Rendering: 0.00034 (1%) | DB: 0.00000 (0%) | 200 OK [http://localhostamf\_alapu/create\_user]
```



Amik kimaradtak

- Socket
- SOAP és társai

Kérdések

Köszönöm a figyelmet

