

Rapid Web Service Integration in Rails

XML-RPC & SOAP
using
Action Web Service & SOAP4R

Rapid Integration

- Need to extend our Ad server to integrate with some partner apps in days
- Technologies in use:
 - Our Ad Server in PHP/MySQL/Linux
 - 3rd Party apps in Java
 - Apps will not be co-hosted
 - Web Services using SOAP or XML-RPC an obvious choice

But...

- SOAP & XML-RPC in PHP is very painful
(anything in PHP can be painful...)
duck for abuse from the PHP fan boys
- Database level integration acceptable
- Enter Ruby on Rails & Action Web Services
 - From zero to hero in 48 hours

So...

- We needed 4 new APIs, here's the simplest:
 - Opt In Manager – stop hassling people who have said enough
 - Mark a MSISDN as opted in or out of an advertising campaign
 - 3 methods (OptIn, OptOut, NoCall) each accepts a single parameter, the MSISDN

define the public api

```
$ script/generate web_service OptMgr OptIn OptOut NoCall
$ vi app/apis/opt_mgr_api.rb
$ cat app/apis/opt_mgr_api.rb
class OptMgrApi < ActionWebService::API::Base
  api_method :OptIn, :expects => [[:msisdn => :string]],
             :returns => [[:success => :bool]]

  api_method :OptOut, :expects => [[:msisdn => :string]],
             :returns => [[:success => :bool]]

  api_method :NoCall, :expects => [[:msisdn => :string]],
             :returns => [[:nocall => :bool]]
end
$
```

Yes, that's all there is to define the public API!

implement the methods

```
class OptMgrController < ApplicationController
  wsd_service_name 'OptMgr'
  web_service_scaffold :invoke

  def OptIn(msisdn)
    # Set advertise_to = true, create object if required
    find_or_create(msisdn, true)
    return true
  end

  def OptOut(msisdn)
    ...
  end

  ...
end
```

implement the methods

```
class OptMgrController < ApplicationController
  wsd1_service_name 'OptMgr'
  web_service_scaffold :invoke

  def OptIn(msisdn)
    # Set advertise_to = true, create object if required
    find_or_create(msisdn, true)
    return true
  end

  def OptOut(msisdn)
    ...
  end
  ...
end
```

implement the methods

```
class OptMgrController < ApplicationController
  wsd1_service_name 'OptMgr'
  web_service_scaffold :invoke

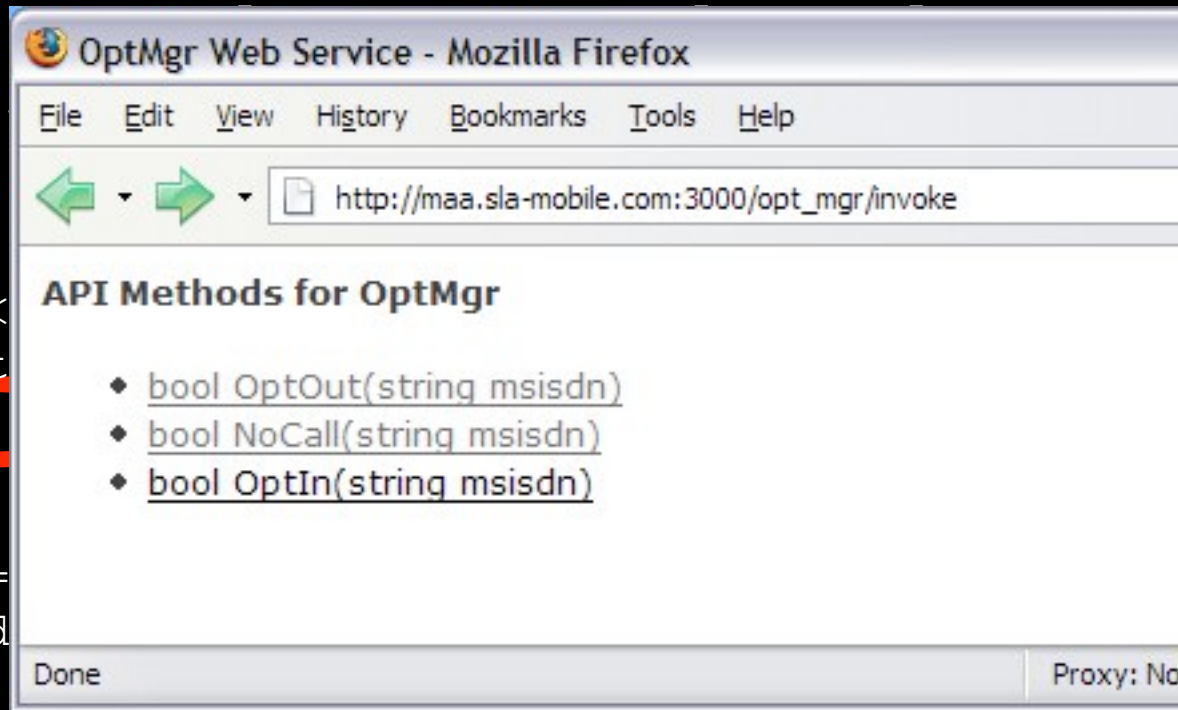
  def OptIn(msisdn)
    # Set advertise_to = true, create object if required
    find_or_create(msisdn, true)
    return true
  end

  def OptOut(msisdn)
    ...
  end
  ...
end
```

impleme

```
class OptMgrController <
  wsd1_service_name 'Opt
  web_service_scaffold :
  def OptIn(msisdn)
    # Set advertise_to =
    find_or_create(msisdn)
    return true
  end

  def OptOut(msisdn)
    ...
  end
  ...
end
```



impleme

```
class OptMgrController <
  wsd_service_name 'Opt
  web_service_scaffold :
end

def OptIn(msisdn)
  # Set advertise_to =
  find_or_create(msisdn)
  return true
end

def OptOut(msisdn)
  ...
end
...
end
```

OptMgr Web Service - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://maa.sla-mobile.com:3000/opt_mgr/invoke_method_params

Method Invocation Details for *OptMgr#NoCall*

Protocol:

SOAP (selected)
SOAP
XML-RPC

Msisdn (string)
447710836915

Invoke

[Back](#)

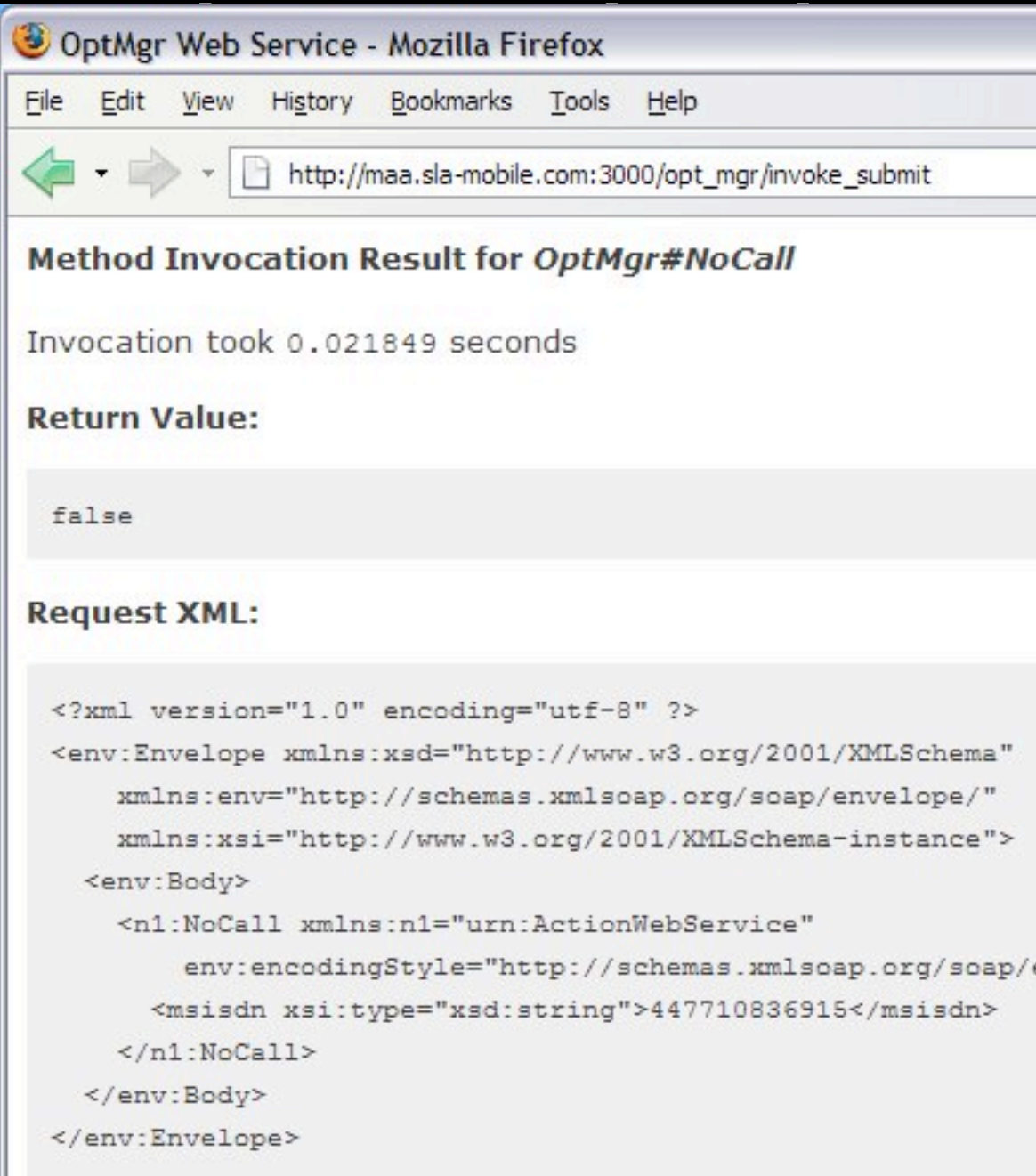
Done Proxy: No

impleme

```
class OptMgrController <
  wsd_service_name 'Opt
  web_service_scaffold :
end

def OptIn(msisdn)
  # Set advertise_to =
  find_or_create(msisdn)
  return true
end

def OptOut(msisdn)
  ...
end
...
end
```



The screenshot shows a Mozilla Firefox browser window titled "OptMgr Web Service - Mozilla Firefox". The address bar contains the URL "http://maa.sla-mobile.com:3000/opt_mgr/invoke_submit". The main content area displays the "Method Invocation Result for OptMgr#NoCall". It indicates that the invocation took 0.021849 seconds and the return value is "false". Below this, the "Request XML" is shown, which is a SOAP message with an envelope, body, and a NoCall action containing the msisdn value "447710836915".

OptMgr Web Service - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://maa.sla-mobile.com:3000/opt_mgr/invoke_submit

Method Invocation Result for *OptMgr#NoCall*

Invocation took 0.021849 seconds

Return Value:

false

Request XML:

```
<?xml version="1.0" encoding="utf-8" ?>
<env:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <n1:NoCall xmlns:n1="urn:ActionWebService"
      env:encodingStyle="http://schemas.xmlsoap.org/soap/
      <msisdn xsi:type="xsd:string">447710836915</msisdn>
    </n1:NoCall>
  </env:Body>
</env:Envelope>
```

the results

- Accelerated Integration
 - Multi Protocol, auto-generation of WSDL
- Our Ad server can now:
 - Check if a subscriber has opted out before delivering an advert (over SMS, Portal Banners etc)
 - Deliver Ads to an ODP environment

implementing an existing WSDL

- not quite to easy
- soap4r is useable, wsdl2ruby generates stubs
- some manual tweaking of the WSDL may be required (unsupported types)

Thank you for listening!

Simon McCartney

simon@mccartney.ie