

RIA's bouwen met Flex en REST

De andere gebruikerservaring

Rich Internet Applications (RIA's) zijn sterk in opkomst om een betere interactie en hogere gebruiksvriendelijkheid te kunnen bieden in bepaalde self-service toepassingen. Technologieën zoals DHTML en AJAX brengen enige verbetering voor de huidige in overvloed aanwezige HTML-toepassingen. Flex biedt echter een compleet andere gebruikerservaring, waarbij de interactie in een Flash-object plaatsvindt. In dit artikel wordt een introductie tot Flex gegeven, waarbij onder meer wordt ingegaan op de mogelijkheid om gegevens uit de database te gebruiken.

In het kader bij dit artikel (pagina 34 en 35) worden REST webservices beschreven die zeer goed gebruikt kunnen worden in combinatie met Flex voor het bouwen van *data(base)-driven* rich internet applications.

AJAX en DHTML zijn onvoldoende

DHTML (Dynamic HTML) is een oude techniek om HTML-toepassingen aantrekkelijker en interactiever te maken. Bij DHTML wordt de HTML via JavaScript dynamisch aangepast door het onderliggende Document Object Model (DOM) van de HTML-pagina te manipuleren. Een heel eenvoudig voorbeeld is het wijzigen van een afbeelding wanneer de gebruiker er met de muis over beweegt. Maar DHTML kan bijvoorbeeld ook gebruikt worden om een aantal invoervelden toe te voegen aan een online formulier wanneer een bepaalde checkbox aangevinkt wordt.

Afbeelding 1. Het bekendste voorbeeld van de toepassing van AJAX is het automatisch tonen of beperken van waarden in zoekvelden op basis van wat de gebruiker intypt

AJAX (Asynchronous JavaScript & XML) is een nieuwere en momenteel zeer populaire technologie. AJAX maakt gebruik van het 'XMLHttpRequest' in JavaScript om (asynchroon, de gebruiker hoeft hier niet op te wachten) nieuwe informatie (in XML-formaat) op te halen zonder de pagina te hoeven verversen. Op basis van deze informatie kan de HTML-pagina via JavaScript aangepast worden. Het bekendste voorbeeld van de toepassing van AJAX is het automatisch tonen of beperken van waarden in zoekvelden op basis van wat de gebruiker intypt, zoals op www.telefoongids.nl (zie afbeelding 1).

Hoewel AJAX en DHTML bijdragen aan een betere interactie en hogere gebruiksvriendelijkheid blijft het hier nog steeds om HTML pagina's gaan, met alle bijhorende beperkingen. Flex biedt een compleet andere ervaring aan gebruikers. De interactie vindt bij Flex plaats in een Flash object (movie) dat ingebed is in een HTML pagina of direct wordt uitgevoerd. Met Flex kunnen eenvoudige grafische applicaties gemaakt worden die een hoge mate van interactiviteit bieden (ondermeer Drag & Drop). De benodigde Flash-player is gratis en vrijwel op iedere computer al geïnstalleerd. Bovendien garandeert Flash cross-platform dezelfde Look & Feel; ontwikkelaars hebben niet te maken met afwijkingen in HTML en JavaScript tussen Firefox, Internet Explorer 6, Internet Explorer 7 en Apple- en Linux-browsers. Overigens kan Flex met AJAX communiceren via de Flex-AJAX Bridge: 'labs.adobe.com/wiki/index.php/Flex_Framework:FABridge'.

Wat is Flex?

Flex is een product, of beter gezegd een verzameling van producten van Adobe. Oorspronkelijk was Flex een product van Macromedia, dat eind 2005 (voor 3,4 miljard dollar) is overgenomen door Adobe. Een Flex-applicatie wordt gecompileerd naar een Flash (movie) file (.swf) en gerund in een Flash-player.

Flash

De meeste lezers zullen Flash kennen van introductiepagina's en banners op websites. Maar Flash wordt ook gebruikt in



Afbeelding 2. Op deze website worden nieuwsberichten uit een content management systeem in een Flash-movie getoond.

YouTube voor het afspelen van videoclipen. Interessant vanuit een database-achtergrond is de mogelijkheid om onderdelen van een Flash-movie dynamisch te maken op basis van informatie in de database. Dit is bijvoorbeeld toegepast op de website van Hogeschool Windesheim (www.windesheim.nl) waar nieuwsberichten uit het op de Oracle XML-database gebaseerde Tangelo content management systeem (www.tangelo.nl) in een Flash movie getoond worden (zie afbeelding 2).

Onderdelen van Flex

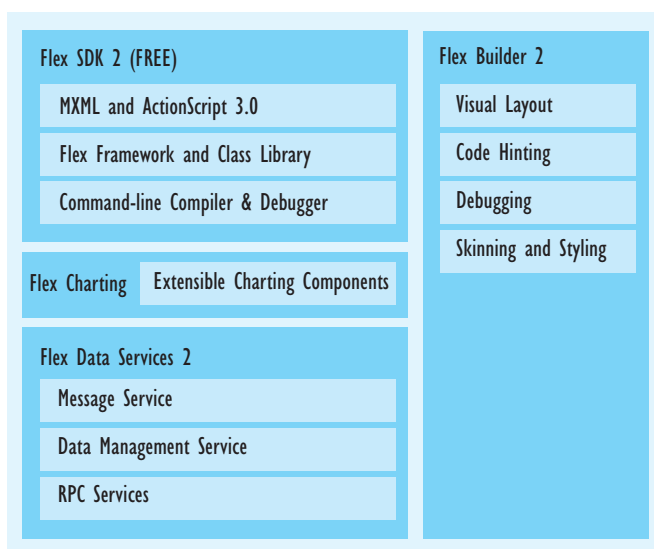
Afbeelding 3 toont de onderdelen van de Flex-productsuite en hun samenhang.

Flex framework en SDK

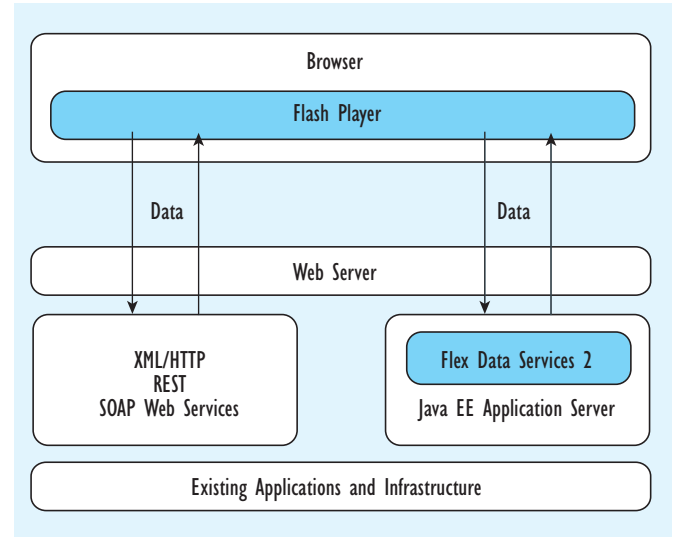
Het Flex-framework is een component-based development framework. De Flex 2 SDK (Software Development Kit) bevat de Flex class library en de MXML (Macromedia Flex Markup Language) en ActionScript 3.0 compilers.

Flex Charting 2

Flex Charting is een verzameling (library) van componenten voor het ontwikkelen van (interactieve) grafieken en diagrammen.



Afbeelding 3. De onderdelen van de Flex-productsuite en hun onderlinge samenhang.



Afbeelding 4. Naast Flex Data Services kan gebruik gemaakt worden van XML over HTTP, REST en SOAP.

Flex Data Services 2

Flex Data Services biedt functionaliteit voor het integreren van Flex met bestaande services, bijvoorbeeld publish/subscribe messaging, het real-time streamen van data en een directe communicatie met bestaande server-side Java-objecten. Flex Data Services integreert uitstekend met een J2EE-applicatie-server.

Belangrijk om op te merken is dat Flex Data Services niet nodig is om dynamische Flash-objecten (movies) te maken. Data kunnen ook zonder Flex Data Services opgehaald worden. Afbeelding 4 toont de verschillende mogelijkheden. Naast Flex Data Services kan gebruik gemaakt worden van XML over HTTP, REST (zie het kader op pagina 34 en 35) en SOAP.

Flex Builder 2

Flex Builder is een op Eclipse gebaseerd Integrated Development Environment (IDE) voor het ontwikkelen van Flex-applicaties. Deze plug-in bevat onder andere code-editors, een designertool, projectmanagementtools en een geïntegreerde debugger. In dit artikel wordt niet verder ingegaan op Flex Charting, Flex Data Services en Flex Builder. Wel wordt in meer detail gekeken naar MXML, de basis van Flex. Meer informatie over Flex kan gevonden worden op: 'www.adobe.com/products/flex/'.

MXML en ActionScript 3.0

MXML (Macromedia Flex Markup Language) is een op XML-gebaseerde programmeertaal voor het definiëren van grafische presentaties, zoals grafieken en diagrammen. Deze kunnen van de grond af aan opgebouwd worden, maar er kan ook gebruik gemaakt worden van een uitgebreide verzameling van meegeleverde basiscomponenten in de Flex class library. Het objectge-

oriënteerde ActionScript 3.0 wordt gebruikt om interactie toe te voegen.

Voorbeeld

In het volgende codevoorbeeld (in feite een 'mini' Flex-applicatie) wordt de HTTP service 'hs' [2] aangeroepen op het moment dat de creatie van het component voltooid is [1]. De HTTP-service is in dit voorbeeld het ophalen van een statische XML file (catalog.xml), maar zal in de praktijk in de regel de aanroep zijn van een 'echte' HTTP-service (zie ook de uitleg van REST in het kader op pagina 34 en 35).

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="absolute" creationComplete="hs.send()" [1]
  <mx:HTTPService id="hs" url="xml/catalog.xml"/> [2]

  <mx:Panel x="0" y="0" width="100%" height="100%" layout="vertical"
    horizontalAlign="center">
    <mx:TileList width="100%" height="100%"
      dataProvider="{hs.lastResult.catalog.product}" [3]
      itemRenderer="ImageRenderer"> [4]
    </mx:TileList>
  </mx:Panel>
</mx:Application>
```

Telefoons.mxml

De XML-file die via de HTTP-service 'hs' wordt opgehaald bevat informatie over mobiele telefoontoestellen.

```
<?xml version="1.0" encoding="utf-8"?>
<catalog>
  <product productId="1">
    <name>Nokia 6010</name> [b]
    <description>
      Easy to use without sacrificing style, the Nokia 6010 phone
      offers functional
      voice communication supported by text messaging, multimedia
      messaging, mobile
      internet, games and more
    </description>
    <price>99.99</price>
    <image>assets/pic/Nokia_6010.gif</image> [a]
    <series>6000</series>
    <triband>>false</triband>
    <camera>>false</camera>
    <video>>false</video>
    <highlight1>MMS</highlight1>
    <highlight2>Large color display</highlight2>
  </product>
  ...
</catalog>
```

catalog.xml

In codevoorbeeld 'Telefoons.mxml' wordt een TileList (een aantal achter elkaar geplaatste rechthoekige 'tegeltjes') dynamisch opgebouwd op basis van de inhoud van de opgehaalde XML-file

[3]. Voor ieder 'product'-element in 'catalog.xml' wordt 'ImageRenderer' aangeroepen [4]. De code van deze component ziet er als volgt uit:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:VBox xmlns:mx="http://www.adobe.com/2006/mxml" width="150"
  height="240"
  horizontalAlign="center" verticalAlign="top">
  <mx:Image source="{data.image}"/> [a]
  <mx:Label text="{data.name}" fontWeight="bold"/> [b]
</mx:VBox>
```

ImageRenderer.mxml

Het 'tegeltje' dat door 'ImageRenderer' wordt gecreëerd is 150 bij 240 pixels groot en toont de afbeelding [a] en naam [b] uit het 'product' element (in 'catalog.xml'). Afbeelding 5 toont het resultaat.

Toepassingen

In het bovenstaande voorbeeld hebben we kunnen zien hoe eenvoudig het is om een dynamische grafische toepassing te maken met Flex. De volgende stap is het toevoegen van interactie, bijvoorbeeld het interactief opvragen van gegevens of het beïnvloeden van de presentatie. Zoals al eerder opgemerkt wordt hiervoor gebruik gemaakt van ActionScript 3.0. Flex biedt unieke mogelijkheden voor bepaalde self-service oplossingen. Adobe zelf noemt de volgende toepassingen:

- product configurators
- expressive data dashboards
- guided selling applications
- collaborative enterprise applications

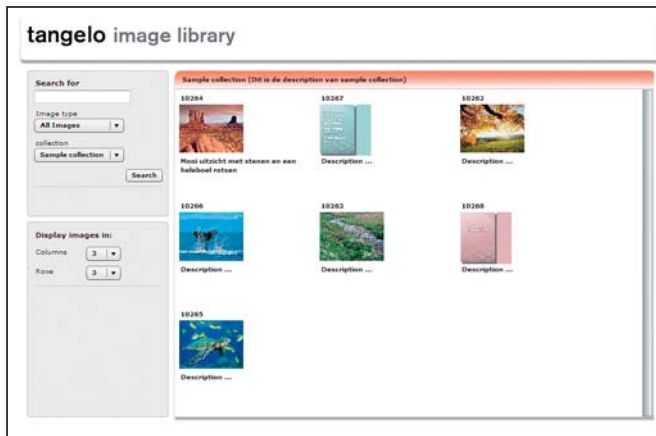


Afbeelding 5. Het 'tegeltje' dat door 'ImageRenderer' wordt gecreëerd toont de afbeelding en naam uit het 'product'-element.

Advertentie

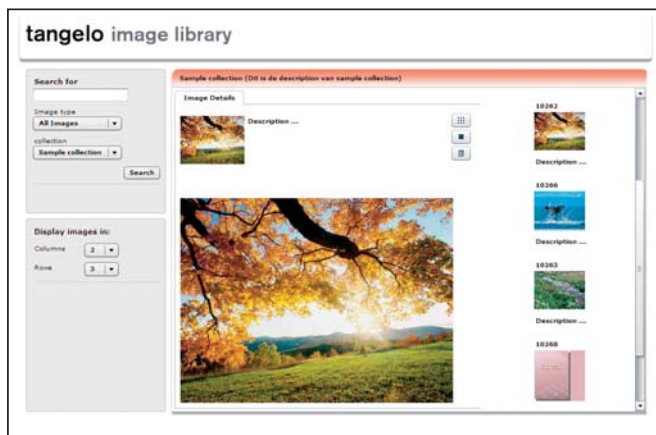
Tangelo Image Library

Een goede fit voor het gebruik van Flex is het browsen door een Image Library. Hierbij wordt gebruik gemaakt van de in het kader beschreven REST-webservices. Afbeelding 6 toont de afbeeldingen in een gekozen collectie.



Afbeelding 6. Een goede fit voor het gebruik van Flex is het browsen door een Image Library.

Door op een afbeelding te klikken wordt deze in het groot getoond en worden de overige afbeeldingen in een 'scrollbare' lijst ernaast getoond (zie afbeelding 7). Voor deze lijst wordt gebruik gemaakt van een standaardcomponent uit de Flex-classlibrary. De eenvoud waarmee dit gerealiseerd wordt, is een goede illustratie van de kracht van Flex.



Afbeelding 7. Voor de 'scrollbare' lijst wordt gebruik gemaakt van een standaardcomponent uit de Flex-class library.

Conclusie

Flex is een krachtig raamwerk voor het ontwikkelen van een nieuwe generatie rich internet applications. De acceptatie van Flash is groot, hoewel bij sommige bedrijven en organisaties Flash-objecten uit veiligheidsoverwegingen nog 'tegengehouden' worden. Het gemak en de open manier (zie REST-webservices

in het kader bij dit artikel op pagina 34 en 35) waarmee data kan worden benaderd draagt sterk bij aan de aantrekkelijkheid van deze technologie. Voor bepaalde self-service toepassingen, waarbij een grafische presentatie of vergaande interactie gewenst is, kan met Flex een unieke oplossing geboden worden.

Leon Derks is software engineer bij Cumquat Information Technology en houdt zich bezig met het ontwikkelen van enterprise Java-oplossingen op het Oracle-platform. Leon is voor opmerkingen en vragen naar aanleiding van dit artikel bereikbaar via leon.derks@cumquat.nl.

Erwin Groenendal (erwin.groenendal@cumquat.nl) is senior solutions architect bij Cumquat Information Technology (www.cumquat.nl).

REST-webservices

In dit kader wordt een korte achtergrond en uitleg gegeven van REST. REST-webservices zijn bij uitstek geschikt om te gebruiken in combinatie met Flex.

Eerste generatie

De eerste generatie webservices behelst het op een zeer eenvoudige manier uitwisselen van XML-data over HTTP (HyperText Transfer Protocol). Met een HTTP *get request* kan informatie worden opgehaald ¹ en met een HTTP *post request* kan informatie worden toegezonden. In feite wordt in deze eerste webservices een Remote Procedure Call (RPC) gedaan over HTTP (als netwerkprotocol) met XML als 'informatiedrager' (payload) voor parameters en return-waarden. Aan de services worden geen voorwaarden gesteld en er worden geen aanvullende (op HTTP- en XML-) standaarden gebruikt. Uiteraard moeten er wel afspraken worden gemaakt over de informatie in de uitgewisselde XML, de vocabulary, zodat de consumer en provider van de webservice elkaar kunnen begrijpen (semantiek). Deze eerste generatie webservices wordt vaak aangeduid met de term 'XML over HTTP'.

SOAP

Na de eerste generatie webservices zijn er diverse aanvullende standaarden gekomen voor Business-to-Business (B2B) toepassingen en voor gebruik van webservices in Service-Oriented Architectures (SOA). De bekendste standaard is SOAP (Simple Object Access Protocol).

Eén van de belangrijkste concepten van SOAP is het gebruik van een envelope met standaard adresseergegevens in het XML-bericht. Binnen de envelope bevindt zich de body van het bericht met de informatie die uitgewisseld wordt. Op basis van de envelope kan een SOAP-server een aanroep van een webservice automatisch doorsturen naar- en laten afhandelen door bijvoorbeeld een Java-servlet.

De SOAP-webservice, inclusief de XML-informatie in de body, wordt beschreven in WSDL (Web Services Description Language). Op basis van een WSDL-beschrijving kan code gegenereerd worden op eenvoudig een webservice aan te kunnen roepen uit bijvoorbeeld C#. In de praktijk is echter gebleken dat de manier waarop in .NET en J2EE waarden in XML worden omgezet en uit XML worden gehaald vaak voor interoperabiliteitsproblemen zorgt.

¹ Een HTTP *get request* kan rechtstreeks vanuit een browser gedaan kan worden. Bijvoorbeeld 'http://www.trynt.com/movie-imdb-api/v2/?t=gattaca' voor het ophalen van informatie over films, in dit geval 'Gattaca', uit de Internet Movie Database (IMDB).

In reactie op deze problemen zijn er weer aanvullende standaarden ontwikkeld door de Web Services Interoperability Organization (WS-I, 'www.ws-i.org').

Hoewel SOAP voor veel toepassingen onmisbare functionaliteit standaardiseert is SOAP voor bepaalde oplossingen te complex (en duur). Een zeer gemakkelijke 'discussie' tussen een ontwikkelaar en een 'SOAP guy' over de complexiteit van SOAP is te lezen op de weblog van Pete Lacey: 'wanderingbarque.com/nonintersecting/2006/11/15/the-s-stands-for-simple'.

Concepten

REST is voor bepaalde toepassingen een eenvoudiger (en goedkoper) alternatief voor SOAP. De toenemende aandacht voor REST kan gezien worden als een reactie op de complexiteit van SOAP en een terugkeer naar eenvoudige 'XML over HTTP'-webservices. REST voegt echter ook functionaliteit toe.

REST staat voor Representational State Transfer en is gebaseerd op drie concepten:

- resource
- resource identifier
- representation

Resource

Een resource is een 'ding' waarvan je informatie wilt kunnen opvragen en dat je wilt kunnen 'manipuleren' vanuit een applicatie. Een voorbeeld is een afbeelding in een Image Library, maar ook een collectie van afbeeldingen is een resource. Alle gegevens waarmee je wilt kunnen werken in een applicatie moeten als resources benoemd worden.

Resource identifier

Elke resource wordt aangeduid met een unieke resource identifier, ook wel aangeduid met Uniform Resource Identifier (URI).

Representation

Door de URI van een resource aan te roepen (via een HTTP *get request*) wordt een representation van de resource verkregen. Meestal is deze representatie in XML, maar het kan ook in HTML, een audio- of video bestandsformaat zijn.

REST interface

Op een REST resource kunnen verschillende acties uitgevoerd worden. Hiervoor wordt gebruik gemaakt van HTTP,

echter niet beperkt tot alleen de HTTP get en post methoden. HTTP staat nog een aantal andere methoden toe. Twee hiervan, put en delete, worden benut door REST, zoals te zien is in tabel 1.

HTTP metho-	Functie	Database/"CRUD" equivalent
POST	Aanmaken van een nieuwe resource	Create
GET	Informatie opvragen over de resource	Retrieve (select)
PUT	Aanpassen van de resource	Update
DELETE	Verwijderen van de resource	Delete

Tabel 1.

In feite kan met REST remote een object gecreëerd worden waarvan de representational state (toestand) kan overgaan – transfer – naar een andere toestand (aangepast kan worden).

Zoals al eerder opgemerkt kan een HTTP get methode eenvoudig vanuit een browser gedaan worden. De overige methoden zullen vanuit een applicatie uitgevoerd moeten worden, bijvoorbeeld een Flex-applicatie.

Ontwerp

Bij het ontwikkelen van REST-webservices moet eerst worden onderkend wat de resources zijn, en met welke URI

deze resources geïdentificeerd worden. Vervolgens moet bepaald worden wat de representatie is van de resources. Zoals hiervoor beschreven, is dit in de regel in XML maar mogelijk ook in een ander bestandsformaat. Tenslotte moet vastgesteld worden welke acties op de resources zijn toegestaan. Is dit alleen informatie opvragen, of ook aanmaken, aanpassen en verwijderen? Tabel 2 toont welke resources, URI's, representations en acties ontworpen worden voor een Image Library.

Implementatie

Vervolgens moeten de ontworpen REST-webservices op 'een manier' geïmplementeerd worden, 'hoe' wordt niet voorgeschreven. Er kan bijvoorbeeld een controller servlet ontwikkeld worden die de meegegeven URI afbeeldt op de aanroep van een bepaalde ActionClass. De ActionClass voert de gevraagde actie uit. Zo zou de GET actie 'http://localhost:8888/rest/library/collection' de volgende representatie in XML terug kunnen geven:

```
<collections>
  <collection id="99">
    <url>http://localhost:8888/rest/library/collection/99</url>
    <name>Sample collection</name>
    <description>
      Dit is de description van sample collection
    </description>
  </collection>
  ...
</collections>
```

Resource	Template URI	Representation	Acties
Alle collecties	/collection	XML	GET PUT POST DELETE
Alle images uit een collectie	/collection/{collection_id}	XML	GET PUT POST DELETE
Thumbnail image	/images/thumb/{image_id}	image (jpeg)	GET
Small image	/images/small/{image_id}	image (jpeg)	GET
Medium image	/images/medium/{image_id}	image (jpeg)	GET
Large image	/images/large/{image_id}	image (jpeg)	GET
High resolution image	/images/high/{image_id}	image (jpeg)	GET
Alle thumbnail images	/images/thumb	XML	GET PUT POST DELETE

Tabel 2.