



the  
**POWER**  
of  
**JAVA™**



JavaOne  
Part of the Network for Enterprise Software

# Code Examples Using Java™ ME Technology and New Web 2.0 Services (Beyond Google Maps)

**Hinkmond Wong**

Sr. Staff Engineer  
Sun Microsystems, Inc.  
<https://j2me-cdc.dev.java.net/>

TS-1302

# Goal

## What You Will Learn

Learn how to use Web 2.0 services with Java™ Platform, Micro Edition technology-enabled cell phones by seeing code examples and stepping through the lines of code.

# Updates

<https://j2me-cdc.dev.java.net/servlets/ProjectDocumentList>  
Look in: conferences->javaone->2006

# Agenda

Introduction to Web 2.0

Overview of Java ME Technology

Bridging Web 2.0 and Java ME Technology

Code Example One: Amazon Searches

Code Example Two: Flickr

Code Example Three: Your Own Service

Demo

Summary

# Agenda

## Introduction to Web 2.0

Overview of Java ME Technology

Bridging Web 2.0 and Java ME Technology

Code Example One: Amazon Searches

Code Example Two: Flickr

Code Example Three: Your Own Service

Demo

Summary

# Introduction to Web 2.0

Moving to semantic-based Internet services

- **The Web as a Platform:** Programming to the Internet, not to a computer system.
- **Collective Services:** Using various Web services together. Mash-Ups.
- **Shareable Data:** Sharing XML data among many services
- **Dynamic Updates:** Software is updated whenever it is used.

# Introduction to Web 2.0 (Cont.)

Moving to semantic-based Internet services

- **Lightweight Programming:** Example: **AJAX** – **Asynchronous JavaScript™** technology **And XML**
  - XHTML/CSS—standards based markup
  - DOM—Document Object Model dynamic display
  - XMLHttpRequest—asynchronous data retrieval
  - JavaScript technology— for Web page programming
- **Cross-Device Compatibility:** Not targeted for one platform
- **Better User Experience:** Combine data from many sites. Example: Map of houses for sale

# Web 2.0

Browser, RSS  
Reader, cell  
phone, etc.



# Introduction to Web 2.0

Moving to semantic-based Internet services

- **The Web as a Platform:** Programming to the Internet, not to a computer system
- **Collective Services:** Using various Web services together—mash-ups
- **Shareable Data:** Sharing XML data among many services
- **Dynamic Updates:** Software is updated whenever it is used

# Introduction to Web 2.0 (Cont.)

## REST Protocol

- **REST**—Representational State Transfer [Roy Fielding]
- Software architecture defined:
  - Components
  - Connectors
  - Data
- Constrained in their relationships in order to achieve a desired set of architectural properties

# Introduction to Web 2.0 (Cont.)

## REST Protocol

- To create a REST service, you need to answer the following questions, and you should answer them in this order:
  - What are the URIs?
  - What's the format?
  - What methods are supported at each URI?
  - What status codes could be returned?
- Example:  

```
http://xml.amazon.com/onca/xml2?t=webservices-20&dev-  
=D1UCR04XBIF4A6&page=1&+”f=xml&mode=books&type=lite+”  
&KeywordSearch='hinkmond'
```

# Introduction to Web 2.0 (Cont.)

## AJAX Example: HTML Page (UI Dynamic Update)

```
<form name="autofillform" action="autocomplete" method="get">
  <table border="0" cellpadding="5" cellspacing="0">
    <tr>
      <td><b>Employee Name:</b></td>
      <td>
        <input type="text" id="complete-field" size="20"
          autocomplete="off"
          onkeyup="doCompletion();" />
      </td>
      <td align="left">
        <input id="submit_btn" type="Submit" value="Lookup Employee" />
      </td>
    </tr>
    <tr><td id="auto-row" colspan="2">&nbsp;</td></tr>
  </table>
</form>
<div style="position: absolute; top:170px;left:140px" id="menu-
popup">
  <table id="completeTable" border="1" bordercolor="black"
cellpadding="0" cellspacing="0" />
</div>
```

# Introduction to Web 2.0 (Cont.)

## AJAX Example: JavaScript Technology

```
function getXHR(url) {
    if (window.XMLHttpRequest) {
        return new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
}

function doCompletion() {
    var url = "autocomplete?action=complete&id=" +
    encodeURIComponent(target.value);
    var req = getXHR(url);
    req.onreadystatechange = processRequest;
    req.open("GET", url, true);
    req.send(null);
}
```

# Introduction to Web 2.0 (Cont.)

## AJAX Example: Servlet

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws IOException, ServletException {
    String targetId = request.getParameter("id");
    Iterator it = employees.keySet().iterator();
    while (it.hasNext()) {
        EmployeeBean e = (EmployeeBean)employees.get((String)it.next());
        if ((e.getFirstName().toLowerCase().startsWith(targetId) ||
            e.getLastName().toLowerCase().startsWith(targetId)) && !targetId.equals("")) {
            sb.append("<employee>");
            sb.append("<id>" + e.getId() + "</id>");
            sb.append("<firstName>" + e.getFirstName() + "</firstName>");
            sb.append("<lastName>" + e.getLastName() + "</lastName>");
            sb.append("</employee>");
            namesAdded = true;
        }
    }
    if (namesAdded) {
        response.setContentType("text/xml");
        response.setHeader("Cache-Control", "no-cache");
        response.getWriter().write("<employees>" + sb.toString() + "</employees>");
    }
    response.setStatus(HttpServletResponse.SC_NO_CONTENT);
}
```

# Introduction to Web 2.0 (Cont.)

## AJAX Example: JavaScript Client Callback

```
function postProcess(responseXML) {
    clearTable();
    var employees = responseXML.getElementsByTagName("employees")[0];
    if (employees.childNodes.length > 0) {
        completeTable.setAttribute("bordercolor", "black");
        completeTable.setAttribute("border", "1");
    } else {
        clearTable();
    }
    for (loop = 0; loop < employees.childNodes.length; loop++) {
        var employee = employees.childNodes[loop];
        var firstName = employee.getElementsByTagName("firstName")[0];
        var lastName = employee.getElementsByTagName("lastName")[0];
        var employeeId = employee.getElementsByTagName("id")[0];
        appendEmployee(firstName.childNodes[0].nodeValue,
            lastName.childNodes[0].nodeValue, employeeId.childNodes[0].nodeValue);
    }
}
```

# Agenda

Introduction to Web 2.0

**Overview of Java ME Technology**

Bridging Web 2.0 and Java ME Technology

Code Example One: Amazon Searches

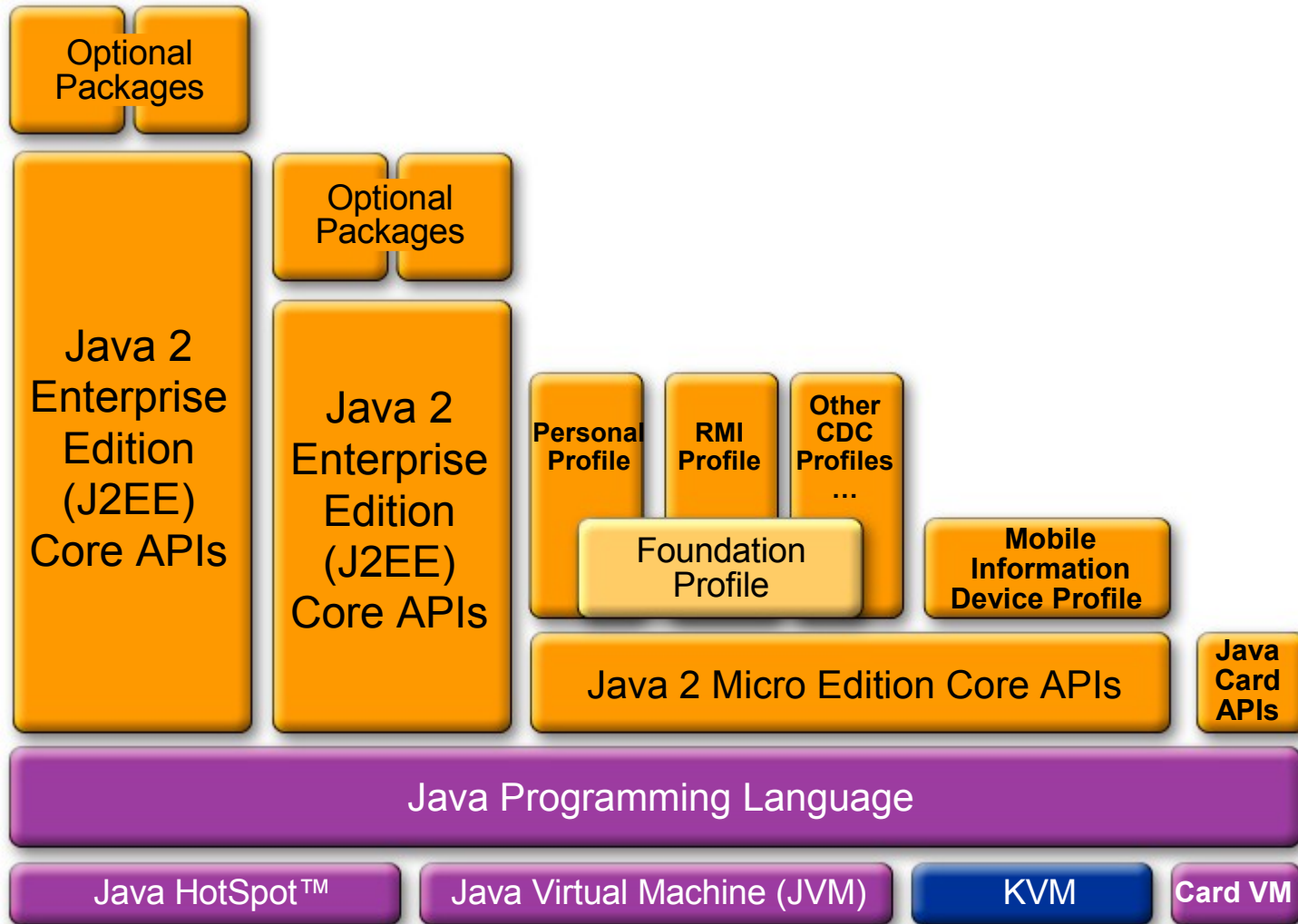
Code Example Two: Flickr

Code Example Three: Your Own Service

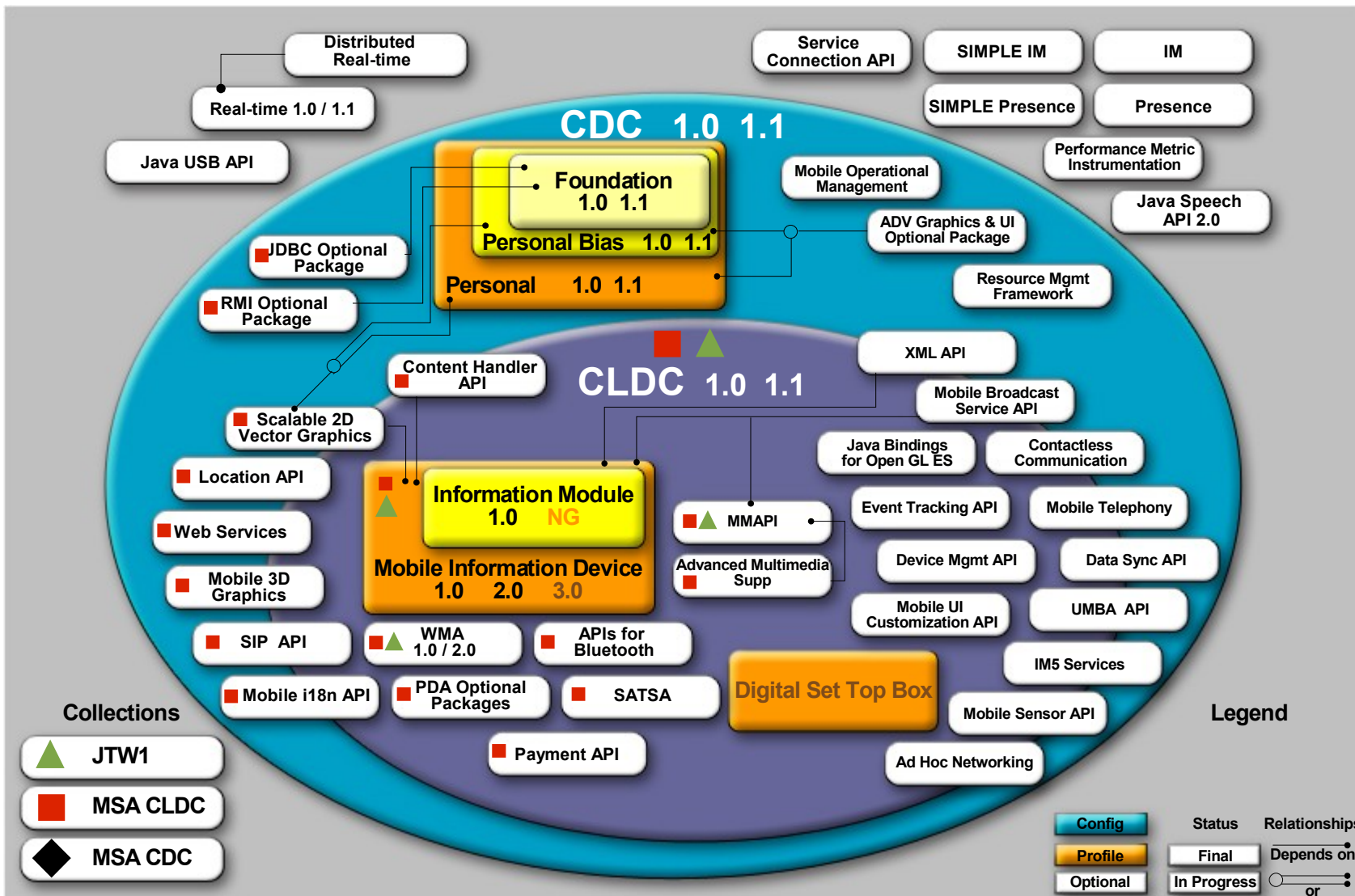
Demo

Summary

# Java Platform, Micro Edition (Java ME)



# Java Platform, Micro Edition (Java ME)



# Overview of Java ME Technology

## Mapping the Layers

- **Configurations:** CLDC and CDC—Virtual Machines and base libraries
- **Profiles:** MIDP and Foundation Profile—Adding more specific APIs
- **Optional Packages:** Platform extensions.
- **Java Community Process<sup>SM</sup>:** Specifications developed by expert groups with representation from different companies and individuals

# Overview of Java ME Technology

## JSR 290

- Integrate Java ME and Web Technologies
  - Web Formats (XHTML, SVG\* Tiny, ESMP\*, CSS\* Basic)
  - Java ME / Browser interactions
  - Java ME Web Engines integration
  - Browser integration of Java ME components
- Integrate the work of professionals with different skills:
  - Various programming skills (Java technology or ESMP)
  - Various graphic art skills

# Overview of Java ME Technology

## JSR 290: What is CDF?

- Compound Document Format
- W3C Working Group
  - Nokia, Vodafone, Sun, France Telecom, Sony Ericsson, RIM, IBM, Adobe, Opera, Iikivo, Fuchsia Design, Infraware, Streamazzo, Mozilla, Justsystem, Helsinki University of Technology, Volantis, Expway, Infraware
- Defining rules for combining XML UI markups
  - XHTML Basic, SVG, CSS Basic, ESMP
- Goal is to facilitate rich multimedia content defined using multiple markup languages (+ ECMAScript)

# Overview of Java ME Technology

## JSR 290: What is the Value Proposition of CDF?

- Adds layout capabilities to combined markup
- Adaptability to screen size and form factor
- Mixed content
  - Interaction rules for combined markup
  - Animation and interactivity as a UI around other content
- Unified and consistent DOM API model, animation model and event model—simple
- Leverages body of experience, content, & tools

# Agenda

Introduction to Web 2.0

Overview of Java ME Technology

**Bridging Web 2.0 and Java ME Technology**

Code Example One: Amazon Searches

Code Example Two: Flickr

Code Example Three: Your Own Service

Demo

Summary

# Bridging Web 2.0 and Java ME Technology

## Java Platform for Small Devices

- **XHTML/CSS** – XML data can also be used by Java ME (JSR 172 – Web Services)
- **DOM** – Future Java Specification Requests will address DOM better (JSR 290 – CDF). For now, use MIDP MIDlets as the User Interface
- **XMLHttpRequest** – Using Generic Connection Framework HttpURLConnection with XML parsing
- **JavaScript technology** – Only found in Opera Mini Browser. Instead use Java ME CLDC/MIDP which is more robust and allows object-oriented design
- JSR 279: Service Connection API for Java ME (**SOA**)
- JSR 280: **XML** for Java ME

# XML Parsing Http Request Code Sample

```
SAXParserFactory saxpf =
    SAXParserFactory.newInstance();
SAXParser saxp = saxpf.newSAXParser();

// HTTP Request of XML data
try {...
StreamConnection httpconn = (StreamConnection)
    Connector.open("http://localhost/test.xml");
...} catch...

InputStream is = httpconn.openInputStream();
InputSource inputsrc = new InputSource(is);

saxp.parse(is, new MyHandler(this));
```

# XML Parsing Http Request Code Sample (Cont.)

```
class MyHandler extends DefaultHandler {
    private Stack tags = new Stack();

    public void startDocument() throws SAXException {}

    public void startElement(String uri, String
        localname, String qName, Attributes attr)
        throws SAXException {
        if(qName.equals("blah")) {
            // Do stuff here
        }
        tags.push(qName);
    }
}
```

# XML Parsing Http Request Code Sample (Cont.)

```
public void characters(...) {
    // Process characters here...
}

public void endElement(...) {
}

public void endDocument() throws SAXException {
    // Action after everything parsed
}
```

# Agenda

Introduction to Web 2.0

Overview of Java ME Technology

Bridging Web 2.0 and Java ME Technology

Code Example One: Amazon Searches

Code Example Two: Flickr

Code Example Three: Your Own Service

Demo

Summary

# Amazon Searching

Java ME tech-  
enabled cell  
phone



# Amazon Searching

Code sample using Amazon as a Web Service

- **REST** – Representational State Transfer
- **Contacting the Amazon server** – Via URL, which results in a HTTP Request  
`http://xml.amazon.com/onca/xml2...`
- **Forming the URL** – Embed the command parameters
- **XML Data is returned** – Parse the data using Java ME technology

# Amazon Http Request Code Sample

```

...
SAXParser saxp = saxpfactory.newSAXParser();

String url = "http://xml.amazon.com/onca/xml2"
+"?t=webservices-20&dev-t=D1UCR04XBIF4A6&page=1&"
+"f=xml&mode=books&type=lite"
+"&KeywordSearch='hinkmond'";

// HTTP Request of XML data
...
StreamConnection httpconn = (StreamConnection)
    Connector.open(url);
...

saxp.parse(is, new MyHandler(this));

```

# Amazon Http Request Code Sample (Cont.)

```
// Dynamically update User Interface in a MIDlet
try {
    // Assume: Parsed return data in returnStr
    textbox = new javax.microedition.lcdui.TextBox
        ("Book data = "+returnStr, 40, 0);

    textbox.addCommand(exitCommand);
    textbox.addCommand(anotherSrchCommand);
    textbox.setCommandListener(this);
    // This updates the screen without refreshing
    display.setCurrent(textbox);
} catch (Exception e) {
    e.printStackTrace();
}
```

# Amazon Http Request Xlet Code Sample

...

```
SAXParser saxp = saxpfactory.newSAXParser();

String urlStr = "http://xml.amazon.com/onca/xml2"
+"?t=webservicess-20&dev-t=D1UCR04XBIF4A6&page=1&"
+"f=xml&mode=books&type=lite"
+"&KeywordSearch='hinkmond'";

// Open Xlet connection
url = new URL(urlStr);
InputStream inputStream = url.openStream();
InputStreamReader inputStreamReader = new
    InputStreamReader(inputStream);
BufferedReader bufferedReader = new
    BufferedReader(inputStreamReader);
saxp.parse(is, new MyHandler(this));
```

# Agenda

Introduction to Web 2.0

Overview of Java ME Technology

Bridging Web 2.0 and Java ME Technology

Code Example One: Amazon Searches

**Code Example Two: Flickr**

Code Example Three: Your Own Service

Demo

Summary

# Flickr Service

Java ME tech-  
enabled cell  
phone



# Flickr Service

## Code sample using Flickr as a Web Service

- **REST** – Representational State Transfer
- **Contacting the Flickr server** – Via URL, which results in a HTTP Request  
`http://www.flickr.com/services/rest/?api_key=...`
- **API Key Registration** – Must be a registered developer at their Web site
- **Forming the URL** – Embed the command parameters
- **XML Data is returned** – Parse the data using Java ME technology

# Flickr Http Request Code Sample

```
...
    SAXParser saxp = saxpfactory.newSAXParser();

    String url ="http://www.flickr.com/services/rest/?"
+"api_key=7f14d2ba4b1c973fab9d539c0a09f3a"
+"&method=flickr.test.echo&name=%22test%22";

    // HTTP Request of XML data
    ...
    StreamConnection httpconn = (StreamConnection)
        Connector.open(url);
    ...

    saxp.parse(is, new MyHandler(this));
```

# Agenda

Introduction to Web 2.0

Overview of Java ME Technology

Bridging Web 2.0 and Java ME Technology

Code Example One: Amazon Searches

Code Example Two: Flickr

**Code Example Three: Your Own Service**

Demo

Summary

# Your Web 2.0 Service

Java ME tech-  
enabled cell  
phone



# Your Web 2.0 Service

## Code sample using Your Own Web Service

- **Your Web Server** – May use REST along with Java EE (servlets), PHP, CGI-BIN script, etc.
- **Contacting your server** – Via URL, which results in a HTTP Request  
`http://www.yourserver.com/services/rest/?api_key=...`
- **API Key Registration** – May decide to have developers register themselves
- **Forming the URL** – Decide on parameters to use.
- **XML Data is returned** – Parse the data using Java ME technology

# Your Web Service Http Request Code Sample

```
...
    SAXParser saxp = saxpfactory.newSAXParser();

    String url ="http://yourserver.com/services/rest?"
+"api_key=0000000000000000000000000000000000000000"
+"&method=your.test.function&param1=%22test%22";

    // HTTP Request of XML data
    ...
    StreamConnection httpconn = (StreamConnection)
        Connector.open(url);
    ...

    saxp.parse(is, new MyHandler(this));
```

# Agenda

Introduction to Web 2.0

Overview of Java ME Technology

Bridging Web 2.0 and Java ME Technology

Code Example One: Amazon Searches

Code Example Two: Flickr

Code Example Three: Your Own Service

Demo

Summary

# DEMO

Code Examples Using Java ME Technology  
and Web 2.0

# Agenda

Introduction to Web 2.0

Overview of Java ME Technology

Bridging Web 2.0 and Java ME Technology

Code Example One: Amazon Searches

Code Example Two: Flickr

Code Example Three: Your Own Service

Demo

Summary

# Summary

- Web 2.0 is a new highly dynamic and popular model of programming
- Bridging Web 2.0 and Java ME Technology is simple
- Examined code examples: Amazon, Flickr, Your Own Web Services
- Web 2.0 services can easily be used with Java ME technology-enabled cell phones

# For More Information

## URLs

### AJAX

<http://en.wikipedia.org/wiki/AJAX>

### REST

<http://www.xfront.com/REST-Web-Services.html>

### Amazon

<http://www.amazon.com/webservices>

### Flickr

<http://www.flickr.com/services/>

### Google

<http://www.google.com/apis/maps/documentation/>

### Other Web 2.0 services

<http://www.programmableweb.com/howto>

# Q&A

**Hinkmond Wong**

Sun Microsystems, Inc.



the  
**POWER**  
of  
**JAVA™**



# Code Examples Using Java™ ME Technology and New Web 2.0 Services

**Hinkmond Wong**

Sr. Staff Engineer  
Sun Microsystems, Inc.  
<https://j2me-cdc.dev.java.net/>

TS-1302