

Mobile and Wireless Systems Programming

Networking



Introduction

- The J2ME Generic Connection Framework
- Wireless Messaging API
- Platform related functions
- Specific Networking APIs

- 2 Classes (1 in MIDP 1.0) : Connector, PushRegistry
- 16 Interfaces (7 in MIDP 1.0) : CommConnection, Connection, ContentConnection, Datagram, DatagramConnection, HttpConnection, HttpsConnection, InputConnection, OutputConnection, SecureConnection, SecurityInfo, ServerSocketConnection, SocketConnection, StreamConnection, StreamConnectionNotifier, UDPDatagramConnection
- 1 Exception : ConnectionNotFoundException

Methods

- Creating new Connection objects
- static Connection open(String url)
- static Connection open(String url, int mode)
- static Connection open(String url, int mode, boolean timeouts)
- static Data(In/Out)putStream
openData(In/Out)putStream(String name)
- static (In/Out)putStream open(In/Out)putStream(String name)

Methods

- **url** : should conform to the URL format as described in RFC 2396
- **protocol** ://[host] :[port]
- **Example** :
 - `HttpConnection c = (HttpConnection)Connector.open("http ://www.univ-pau.fr") ;`
 - `SocketConnection sc = (SocketConnection)Connector.open("socket ://localhost :9000") ;`
 - `Connection fc = Connector.open("file ://images/img.png")`
- **Must cast the returned Connection**
- **mode** :
 - READ
 - WRITE
 - READ_WRITE
- **timeouts** :InterruptedIOException when it detects a timeout condition

- most used/useful Interface
- perform HTTP POST and GET requests from a MIDlet
- a lot of methods :
 - `setRequestMethod(HttpConnection.POST)`
 - `getHost()`
 - `getPort()`
 - `getProtocol()`
 - ...

Download an image

```
public void load(String url) throws IOException {
    HttpURLConnection hc = (HttpURLConnection) Connector.open(url);
    byte[] img;

    try {
        InputStream in = hc.openInputStream();
        try {
            img = new byte[in.available()];
            in.read(img);
        } finally {
            in.close();
        }
    } finally {
        hc.close();
    }
}
```

Communicate with a Servlet

```
HttpConnection hc = null;
InputStream is = null ;
OutputStream os = null ;
try {
    hc = (HttpConnection) Connector.open ( url , Connector.READ_WRITE) ;
    hc.setRequestMethod (HttpConnection.POST) ;
    hc.setRequestProperty ("User-Agent" , " Profile/MIDP-1.0 Configuration/CLDC" );
    hc.setRequestProperty ("Content-Language" , "fr-FR" );
    hc.setRequestProperty ("Content-Type" , "application/x-www-form-urlencoded" );

    os = hc.openOutputStream ( ) ;
    byte [] msg = messageToSend.getBytes ( ) ;
    os.write(msg) ;
    is = new DataInputStream (hc.openInputStream ( ) ) ;
    int ch ;
    while ((ch = is.read()) != -1) {
        serverResponse = serverResponse + (char) ch ;
    }
} finally {
    if ( hc != null ) hc.close() ;
    if ( is != null ) is.close() ;
    if ( os != null ) os.close ( ) ;
}
}
```


- The MIDlet register for inbound connections
- The AMS listens for inbound notification requests
 - static long registerAlarm(String midlet, long time)
 - static void registerConnection(String connection, String midlet, String filter)
- MIDlet-Push-<n> : <ConnectionURL>, <MIDletClassName>, <AllowedSender>
- Example :
 - MIDlet-Push-1 : socket :// :79, com.sun.example.SampleChat, *
 - MIDlet-Push-2 : datagram :// :50000, com.sun.example.SampleChat, *
- MIDlet-Permissions : javax.microedition.io.PushRegistry,
- Since MIDP 2.0

- WMA 1.1 (JSR 120)
- javax.wireless.messaging
- Based on the GCF
 - Message
 - MessageConnection
 - MessageListener

```
MessageConnection mc = (MessageConnection)  
Connector.open("sms ://+336XXXXXXXXX");
```

Send SMS

```
public void sendTextMessage(MessageConnection mc, String msg, String url){
    try{
        TextMessage tmsg =
            (TextMessage)mc.newMessage
            (MessageConnection.TEXT_MESSAGE);
        if (url!= null)
            tmsg.setAddress(url);
        tmsg.setPayloadText(msg);
        mc.send(tmsg);
    }catch(Exception e){
        // Handle the exception...
        System.out.println("sendTextMessage_ " + e);
    }
}
```

- WMA 2.0 (JSR 205)
- Multipart Message (Subject, multiple addresses, larger message...)
 - MultipartMessage
 - MessagePart
 - SizeExceededException

Send MMS

```
public void sendMultipart(String address, String subject,
                          MessagePart[] parts)
    throws IOException, InterruptedException {

    String cs = "mms://+" + address;

    MessageConnection mc = (MessageConnection) Connector.open(cs);
    MultipartMessage mm = (MultipartMessage)
        mc.newMessage(MessageConnection.MULTIPART_MESSAGE);
    mm.setSubject(subject);
    for (int i = 0; i < parts.length; i++)
        mm.addMessagePart(parts[i]);
    mc.send(mm);
}
```

- javax.microedition.midlet.MIDlet
- MIDP 2.0

```
public final boolean platformRequest(String URL)
```

- Launch browser : "http ://www.univ-pau.fr"
- Perform a call : "tel :00336XXXXXXXXX"

Conclusion

- SIP, Bluetooth and Obex, XML and Web Services
- HTTP and HTTPs are widely used with 3G networks
- Security constraints
- Huge fragmentation issues
- For more information :
 - WTK Networking examples (HTTP, Socket...)
 - WMA :
<http://developers.sun.com/mobility/midp/articles/wma2/#0>
 - WMA :
<http://www.ibm.com/developerworks/wireless/library/wi-extendj2me/>
 - GCF :
<http://www.devx.com/getHelpOn/10MinuteSolution/16646/1763/pa>
 - GCF : <http://on-java.com/pub/a/onjava/2001/12/05/wirelessjava.html?page=1>
 - GCF :