

Contents

- 1 Introdução
- 2 Desenvolvimento da Aplicação
 - ◆ 2.1 Plataforma Java
 - ◆ 2.2 Estilo de Codificação
 - ◆ 2.3 Hierarquia de Pacotes
- 3 Semelhanças e diferenças na API
 - ◆ 3.1 Instaladores
 - ◆ 3.2 Assinando aplicações
 - ◆ 3.3 Ferramentas
- 4 Permissão de acesso aos Recursos
- 5 Armazenamento de dados no dispositivo

Introdução

Android é uma plataforma de software para dispositivos móveis que inclui o sistema operacional, *middleware* e algumas aplicações chaves. O ambiente de desenvolvimento (SDK) da plataforma Android provê as ferramentas e APIs necessárias para iniciar o desenvolvimento de aplicações na plataforma Android usando a linguagem de programação Java. Este artigo tem por auxiliar o processo de portar uma aplicação desenvolvida para Android para *Java ME S60 5th Edition*.





Desenvolvimento da Aplicação

Android possui uma máquina virtual chamada Dalvik[1] e uma API própria semelhante a Java mas totalmente implementada pela *Google* e pela *Open Handset Alliance*[2]. Sendo assim, objetiva-se mostrar como efetuar o port dessas API para as JSRs desenvolvidas pela Sun microsystems para Java ME.

Plataforma Java

Android executa a versão 1.5 do java. Nesta versão há suporte para diferentes funcionalidades, maioria das quais também é suportadas nos dispositivos S60 5ª Edição.

Estilo de Codificação

Geralmente uma aplicação *Android* possui muitos arquivos XML responsáveis por vários tipos de configuração. Tal característica é incomum nas aplicações Java Me. Em uma aplicação *Android*, no desenvolvimento da interface do usuário, arquivos XML são utilizados para fazer a ligação entre a lógica da aplicação e os componentes gráficos. Por outro lado, em uma aplicação Java ME a referência aos componentes são realizados, em geral, diretamente código-fonte.

Hierarquia de Pacotes

Os detalhes sobre as classes da plataforma Android e da API da Java ME podem ser obtidos na documentação referente a ambas plataformas ([3] e [4] respectivamente). Geralmente ambas APIs possuem bastante semelhanças no modo como a lógica de negócio é desenvolvida, deste modo portar o código de uma plataforma para outra torna-se muito mais acessível.

Semelhanças e diferenças na API

A API de Java para *Android* possui os seguintes pacotes.

```
# java.math - Big numbers, rounding, precision
# java.net - Network I/O, URLs, sockets
# java.nio - File and channel I/O
# java.security - Authorization, certificates, public keys
# java.sql - Database interfaces
# java.text - Formatting, natural language, collation
# java.util (including java.util.concurrent) - Lists, maps, sets, arrays, collections
# javax.crypto - Ciphers, public keys
# javax.net - Socket factories, SSL
# javax.security (except javax.security.auth.kerberos, javax.security.auth.spi, and javax.security.sasl)
# javax.sound - Music and sound effects
# java.io - File and stream I/O
# java.lang (except java.lang.management) - Language and exception
# javax.xml.parsers - XML parsing
# org.w3c.dom (but not sub-packages) - DOM nodes and elements
# org.xml.sax - Simple API for XML
```

De maneira semelhante a Java ME alguns recursos não são incluídos da API de Java para *Android*, visto que são considerados as limitações de processamento dos dispositivos. Tais pacotes são apresentados abaixo.

```
# java.applet
# java.awt
# java.beans
# java.lang.management
# java.rmi
# javax.accessibility
# javax.activity
# javax.imageio
# javax.management
# javax.naming
# javax.print
# javax.rmi
# javax.security.auth.kerberos
# javax.security.auth.spi
# javax.security.sasl
# javax.swing
```

Instaladores

Nokia - Jad, Jar

Android - .apk(using Ant[5])

Assinando aplicações

Suporte Nokia: VeriSign, thawte, Java Verifed Certificate etc

Suporte Android: por chave privada (ferramentas como Keytool and Jarsigner)

Ferramentas

- Wireless tool kit(WTK)
- Nokia S60 SDK

Permissão de acesso aos Recursos

Na plataforma Android, o desenvolvedor precisa configurar um conjunto de arquivos *manifest* no projeto de forma a possibilitar o acesso do usuário a alguns recursos restritos do dispositivo. Numa aplicação Java ME, por outro lado, tal acesso é garantido de duas formas, ou a partir da assinatura da aplicação ou, durante a execução, a aplicação irá solicitar acesso aos recursos diretamente ao usuário, que poderá permitir ou não sua execução.

Armazenamento de dados no dispositivo

Java ME : RMS

Android : banco de dados SQLite[6].

Este artigo é uma tradução do artigo em inglês [Porting Android \(Java\) applications to Java ME on S60 5th Edition](#).