

Progetto: **Piattaforma client-server per un provider di servizi mobile computing**  
Proponente: **Rino Goller / Metacortex s.r.l.**

---

# **Procedimento di sviluppo di un servizio MetaPlanet**

---

Versione **1**  
Data **15 December 2011**  
Autori **Rino Goller, Maurizio Barazzuol, Katuscia Reversi, Carlo Mirabassi,  
Cristiano Carlevaro.**

Abstract **Manuale d'installazione di un servizio per la piattaforma Metaplanet**

---

**Bando 5/2009: Interventi attraverso partnership tra imprese e organismi di ricerca  
per lo sviluppo della ricerca e dell'innovazione**

Legge provinciale n. 6 del 1999

## Indice

<b>INTRODUZIONE.....</b>	<b>3</b>
<b>ARCHITETTURA DI UN SERVIZIO.....</b>	<b>3</b>
<b>PROCEDURA DI CREAZIONE DI UN SERVIZIO.....</b>	<b>3</b>
<b>X CONFIGURATION PAGE.....</b>	<b>4</b>
<b>Comando definizione XML.....</b>	<b>4</b>
<b>Tag Service.....</b>	<b>4</b>
<b>Tag Screen.....</b>	<b>5</b>
<b>Tag navigation.....</b>	<b>5</b>
<b>Tag check.....</b>	<b>6</b>
<b>LAYOUT PAGE.....</b>	<b>6</b>
<b>Form.....</b>	<b>6</b>
<b>Panel.....</b>	<b>7</b>
<b>Label.....</b>	<b>7</b>
<b>Button.....</b>	<b>8</b>
<b>MENÙ PAGE.....</b>	<b>9</b>
<b>Menu.....</b>	<b>9</b>
<b>Item.....</b>	<b>9</b>
<b>PROCEDURA INSTALLAZIONE SERVIZI.....</b>	<b>9</b>
<b>CONCLUSIONE .....</b>	<b>9</b>

## **INTRODUZIONE**

Il nostro progetto di una piattaforma open source per diversi sistemi mobili prevede che da essa vengano erogati un certo numero di servizi. La nostra piattaforma è stata programmata nel linguaggio Java. Il codice della piattaforma in quanto open source sarà reso disponibile in rete e migliorabile da qualsiasi utente. La nostra idea è inoltre quella di dare la possibilità agli utenti di poter creare dei servizi da aggiungere alla nostra piattaforma direttamente senza passare dall'analisi del codice oltre che alla modifica della piattaforma stessa.

### **Architettura di un servizio**

La nostra piattaforma consta di diversi servizi, ma abbiamo dato la possibilità di poterne creare di nuovi. Questo permette di avere una nostra piattaforma aperta a più sviluppatori possibile. La scelta fra i vari linguaggi è ricaduta sul XML e sul JSP in quanto molto semplici da apprendere e di un uso ormai sterminato.

Il servizio per la nostra piattaforma dovrà essere una raccolta di pagine XML e JSP racchiuse in una cartella. La cartella dovrà avere il nome del servizio ed inoltre all'interno della cartella ci dovrà essere un XML di configurazione nominato come il servizio, che permetta alla nostra piattaforma di riconoscere il servizio come tale.

### **Procedura di creazione di un servizio**

Per la creazione di un servizio MetaPlanet sono necessari almeno tre tipi di file.

Per prima cosa la collezione dei file deve essere all'interno di una cartella il cui nome corrisponde a quello del servizio. Bisogna avere un file di configurazione scritto in linguaggio XML che permetta al servizio di configurarsi all'interno della piattaforma. Inoltre bisogna avere i file XML di definizione delle pagine visualizzate. La logica che gestisce la navigazione è integrata nelle pagine JSP che non devono necessariamente trovarsi nella cartella del servizio sul server Metacortex, ma possono stare su un altro server come ad esempio quello dello sviluppatore o di un cliente.

## Configuration Page

Per la creazione di un servizio MetaPlanet bisogna innanzitutto creare un XML chiamato "Configuration page" che non è nient'altro che un classico file .xml all'interno del quale i tag sono stati precedentemente definiti all'interno della piattaforma e più precisamente corrispondono ad un comando preciso. Il codice ha al suo interno un insieme di tag che permettono di definire i comportamenti del servizio. Questi tag verranno tradotti in comandi dalla nostra piattaforma. I tag inoltre possiedono una gerarchia di utilizzo ovvero certi tag vanno definiti all'interno di altri. Ogni tag comprende inoltre una lista di attributi che permettono l'esecuzione di un certo numero di comandi. I tag permessi dal nostro XML sono:

- Comando definizione XML
- Tag service
- Tag screen
- Tag navigation
- Tag check

### Comando definizione XML

Per definire che il foglio di lavoro sia del tipo XML bisogna intestarlo con la seguente riga di codice:

```
<?xml version="1.0" encoding="UTF-8"?>
```

### Tag Service

Per definire un servizio bisogna racchiudere il codice XML all'interno del seguente TAG:

```
<SERVICE>
```

```
    codice xml
```

```
</SERVICE>
```

Questo tag può essere definito solo una volta all'interno del nostro xml e possiede la gerarchia più alta, ovvero contiene tutti gli altri tag. Service possiede vari attributi quali:

- "name": è l'attributo che corrisponde al nome del servizio che dovrà

essere univoco per il corretto funzionamento

- “url”: questo attributo definisce il path del servizio.

Riportiamo un esempio di utilizzo qui a seguito:

```
<SERVICE name="meteo" url="ottieni_meteo">
```

### Tag Screen

Il tag screen rappresenta la definizione di una schermata del servizio. All'interno del servizio bisognerà elencare tutte le schermate disponibili al suo interno. Questo tag possiede una gerarchia di secondo grado ovvero è contenuto nel tag service. Vi possono essere più screen in quanto rappresentano le possibili pagine da visualizzare all'interno del servizio. Screen comprende inoltre i seguenti parametri :

- “name”: l'attributo name definisce il nome della schermata
- “xml”: l'attributo xml definisce il path al quale fa riferimento il file .xml di definizione della schermata.

A seguito verrà riportato un esempio:

```
<SCREEN name="richiestaPermessiStrade" xml="main.xml"/>
```

### Tag navigation

Il tag navigation indica il percorso che avviene nelle schermate. Al suo interno sono contenuti i possibili percorsi che si possono prendere all'interno del servizio. La sua gerarchia è di secondo livello in quanto è contenuto solo dal tag service. Al suo interno può contenere vengono i tag check che verranno controllati in ordine di inserimento. Alla prima condizione valida il sistema andrà alla pagina indicata dall'attributo “goto”. A seguito riportiamo un esempio del codice:

```
<NAVIGATION name="esplora_risorse">
  <CHECK condition="isNotEmpty" goto="invia" />
  <CHECK goto="esplora"/>
</NAVIGATION>
```

Questo tag possiede i seguenti attributi:

- “name”: corrisponde al nome di definizione della navigazione

## Tag check

E' il tag dove è contenuto il possibile percorso. Questo tag possiede una gerarchia di terzo livello in quanto è contenuto sia all'interno del tag service che del tag navigation. Vi possono essere più tag di questo tipo. Check possiede più attributi:

- "name": descrive il nome del check
- "condition": indica la condizione con la quale è possibile interrogare il servizio
- "goto": indica la pagina xml che deve essere visualizzata nel caso in cui la condizione sia verificata

Le condizioni possibili dell'attributo "condition" sono :

- isEmpty( è vuoto)
- isEmpty(not è vuoto)
- equals(uguale)

A seguito riportiamo un esempio di codice:

```
<CHECK condition="isEmpty" goto="invia" />
```

## Layout Page

Essendo XML un metalinguaggio molto semplice ed intuitivo abbiamo pensato che per rendere più semplice la creazione di un nostro servizio e accessibile anche ad un non esperto di Java, potevamo utilizzarlo anche per la definizione del layout delle pagine.

Quindi per definire il layout della pagina bisognerà scegliere fra i componenti che verranno messi a disposizione. Il creatore del servizio dovrà creare per ogni pagina visualizzata un file XML che ne definisca la visualizzazione utente. Per semplicità sono stati creati da noi componenti preimpostati richiamabili dall'utente utilizzando i tag all'interno del XML. I possibili tag utilizzabili dagli sviluppatori di servizi MetaPlanet sono:

- FORM
- PANEL
- LABEL
- BUTTON

## Form

Tutta la definizione della pagina deve essere all'interno di questo tag ovvero possiede la gerarchia più alta. Anche per questo tag è possibile definire dei parametri fra i quali:

- “\_” : definisce l'etichetta della form.
- “layout” : definisce il metodo di visualizzazione della pagina. Può essere di diversi tipi ma per ora sono stati solamente implementati i layout cascade( a cascata)e table.

A seguito riportiamo un esempio:

```
<FORM _="Servizio 1" layout="cascade" >
  codice
</FORM>
```

## Panel

Questo tag definisce i pannelli contenuti all'interno della pagina e possiede una gerarchia di secondo grado ovvero è contenuto nel tag form. Come per ogni tag anche questo viene definito con vari attributi:

- “name” : definisce il nome del pannello
- “layout” : definisce il layout del pannello ovvero il posizionamento degli elementi al suo interno
- “direction” : definisce la direzione di inserimento degli elementi. I possibili parametri sono “vertical” o “horizontal”. Questo attributo viene definito solo per il layout cascade.

A seguito riportiamo un esempio di codice:

```
<PANEL layout="table" direction="vertical" >
  <LABEL _="Label1.1" name="nome1" cellX="1" cellY="0" cellWidth=""
  cellWeigth="" />
  <LABEL _="Label1.2" name="descrizione1" cellX="1" cellY="1"
  cellWidth="" cellWeigth="" />
  <LABEL _="Label1.3" name="dimensione1" cellX="2" cellY="1"
  cellWidth="" cellWeigth="" />
</PANEL>
```

## Label

Con Label è possibile inserire delle etichette. Come ogni tag possiede la sua gerarchia che è di terzo livello ovvero è contenuto all'interno di altri due tag e possiede i suoi attributi di definizione:

- “\_”: definisce l'etichetta
- “name”: definisce il nome della label
- “posizioni”: possibilità di scegliere le posizioni dei label all'interno del pannello.

```
<LABEL _="Label1.1" name="nome1" cellX="1" cellY="0" cellWidth=""  
cellWeigth="" />
```

## Button

Per definire dei bottoni in una pagina del nostro servizio bisognerà utilizzare il tag button. Questo come gli altri tag ha una gerarchia che è di terzo livello e vari attributi di definizione quali:

- “\_”: definisce l'etichetta del bottone
- “name”: definisce il nome del bottone
- “service”: il parametro service definisce il nome del servizio da chiamare

A seguito riportiamo un esempio di codice:

```
<BUTTON _="Indietro" name="BACK" />
```



## Menù Page

Un servizio oltre che ad avere le pagine di definizione abbiamo pensato potesse avere dei Menù. Per rendere più semplice la procedura di creazione di un Menù abbiamo pensato di poter creare un xml che permetta di definire le caratteristiche del menù.

Per semplicità sono stati creati da noi componenti preimpostati richiamabili dall'utente utilizzando i tag all'interno del XML. I possibili tag utilizzabili dagli sviluppatori di servizi MetaPlanet sono:

- MENU
- ITEM

### Menu

Questo tag definisce la creazione di un Menù. Come ogni tag ha una gerarchia che è di primo livello e possiede i seguenti attributi:

- “\_” : definisce l'etichetta del menù
- “layout” : definisce la disposizione dei componenti all'interno del menù

### Item

Questo tag definisce il componente che verrà visualizzato nel menù. Item possiede una gerarchia di secondo livello e i seguenti attributi:

- “\_” : descrive il nome dell'etichetta
- “name” : definisce il nome del componente da inserire(servizio)
- “description” : definisce la descrizione del componente
- “icon” : definisce il path dell'icona corrispondente al componente

## Procedura Installazione servizi

Una volta creato il servizio così come precedentemente spiegato, la cartella dov'è racchiuso verrà inserita all'interno della raccolta dei servizi della nostra piattaforma sul nostro server. Per il funzionamento corretto la cartella, avente come nome il nome del servizio, verrà inserita all'interno del cartella WebPages del nostro server. Il servizio verrà reso disponibile non appena caricato.

## Tutorial “Hello World”.

Il procedimento illustrato di seguito vuole essere di aiuto per illustrare la procedura standard per la creazione di un semplice servizio compatibile con la piattaforma Metaplanet.

Andremo quindi a creare un applicazione che, alla sua apertura, visualizzi il classico “Hello World”.

Vedremo quindi l'uso di diversi tag xml creati ad hoc per la nostra piattaforma. Ripetiamo, nonostante sia stato detto all'inizio di questo capitolo, che è necessario creare file xml di configurazione, file xml che permettano al sistema di poter riconoscere il codice prodotto come un servizio valido.

Vediamo di seguito pochi semplici passi per poter creare il servizio:

1. Apertura dell'ambiente di sviluppo (ad es. Netbeans).
2. Dal menu creiamo un primo file e lo chiamiamo “helloworld.xml” e ci prepariamo ad editarlo.
3. Al suo interno specificheremo che “Hello World” è un servizio. Vediamo come:

```
<SERVICE name="helloworld">  
  <SCREEN name="helloworld" xml="main.xml"/>  
  <NAVIGATION name="helloworld">  
    <CHECK goto="helloworld"/>  
  </NAVIGATION>  
</SERVICE>
```

Analizzando il codice si capisce l'utilizzo dei Tag spiegati nelle sezioni precedenti di questa relazione. Per le definizioni rigorose e più dettagliate di ogni Tag invitiamo il lettore a tornare indietro di qualche pagina per poter comprendere meglio il ruolo di ciascun Tag.

4. Andiamo a creare ora il secondo file xml che chiameremo “main.xml” che contiene il servizio vero e proprio. Ovviamente più il servizio è complicato più dovranno essere incrementate le righe di codice xml e laddove sia richiesto anche il codice jsp con il quale generare la logica di controllo del servizio. Per realizzare il nostro “Hello World” non è necessario implementare codice Java quindi possiamo limitarci ad integrare codice xml.

```
<FORM _="Hello World Service" layout="cascade" >  
  <LABEL _="HELLO WORLD"/>
```

```
<BUTTON _="Indietro" name="BACK"/>  
</FORM>
```

5. L'ultimo passo per vedere attivo il proprio servizio è quello di renderlo disponibile nella lista servizi. Per poter effettuare questo tipo di operazione è necessario che lo sviluppatore invii a Metacortex i file xml comprese le icone e le varie pagine che vengono visualizzate nel servizio, mentre il codice che amministra il servizio e che raccoglie elabora e gestisce i dati inviati dall'utente per usufruire del servizio possono stare sul server dello sviluppatore. L'unico accorgimento del quale tener conto è che nell' xml che ci viene consegnato deve esserci specificato il server al quale il servizio fa riferimento.

## Conclusione

Questo documento vuole essere una sorta di guida per coloro che desiderano creare la loro applicazione. Ovviamente lo sviluppatore deve avere delle competenze di base che non possiamo e né vogliamo dare in questo documento poiché sarebbero superflue e poco inerenti con la linea guida di questa relazione.