Menambah Iklan di dalam Aplikasi (In-App Advertising)

Artikel ini akan menjelaskan kepada rekan developer sekalian bagaimana cara untuk menambah iklan dengan perantara Inneractive di dalam aplikasi Series 40, Web Apps khususnya. Sebuah template project yang sudah terdapat in-app ads akan bersama-sama dibuat dan versi final dapat di download pada akhir artikel ini.

Note: This is an entry in the Asha Touch Competition 2012Q3

Permulaan

Iklan yang berada dalam sebuah aplikasi adalah sebuah cara yang sangat populer untuk mendapatkan penghasilan dari aplikasi tersebut. Cara ini dilakukan dengan cara mendistribusikan aplikasi itu secara gratis, dan pembuat aplikasi tersebut mendapatkan penghasilan dari user yang mengklik iklannya. Iklan yang muncul akan diambil melalui internet dan ditampilkan ke user ketika user menggunakan aplikasi kita. Maka dari itu, In-App Ads ini sangat cocok untuk Web Apps karena Web Apps itu sendiri untuk dapat berjalan membutuhkan koneksi internet, secara otomatis In-App Ads nya juga akan ter-load.

Nokia Asha Web Apps memilki API dari InnerActive dan Google Adsense dalam kasus tertentu.

Penempatan iklan dalam aplikasi Web Apps juga sangat penting. Iklan harus ditempatkan di tempat yang terlihat, tetapi tidak mengganggu kenyamanan user dalam menggunaan aplikasi itu sendiri. Di dalam artikel ini, kami akan menunjukkan bagaimana cara membuat sebuah template aplikasi berisi In-App Ads yang ditempatkan di tempat yang sangat pas, seperti yang dianjurkan oleh Inneractive itu sendiri. Kita juga akan membahas bagaimana cara memberikan informasi tentang aplikasi anda, agar iklan yang ditampilkan sesuai dengan aplikasi anda.

Template Basic Project dengan Ads

Buka Nokia WDE, dan buat sebuah project baru (File -> New -> Series 40 Web App (wgt)) dan pilih yang “Basic Project”. Beri nama project nya “Ads Template”. Kita akan memodifikasi project ini untuk mengubahnya menjadi template dasar dalam membuat aplikasi yang ada iklan di dalamnya.

Inneractive Ads

Untuk mendaftarkan diri anda di Inneractive Klik link ini. Setelah anda menerima email konfirmasi, login ke situs inner-active dan klik **Add App**. Isi semua informasi yang dibutuhkan dan klik **Create**. Anda akan mendapatkan sebuah *App Id*. Code ini digunakan untuk mengidentifikasi ads request dari aplikasi anda.

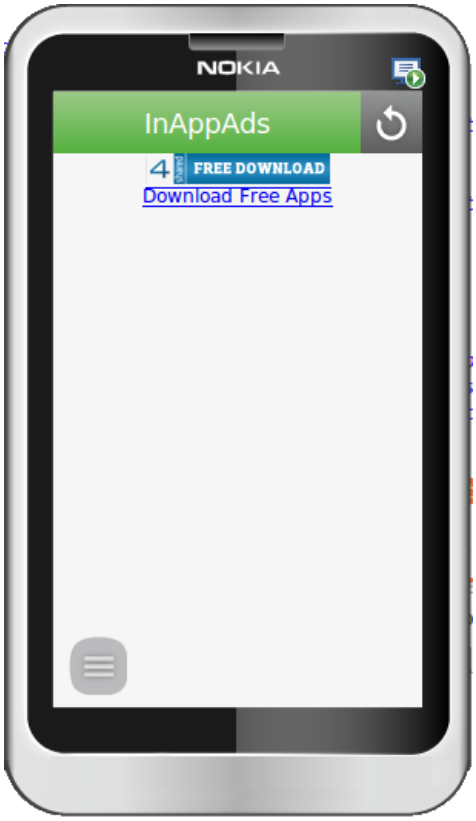
Untuk menambahkan iklan dari Inneractive ke dalam aplikasi anda, buka **index.html** di WDE anda dan tambahkan kode berikut di setelah elemen `div` yang memiliki kelas `ui-content`.

```
<div id="ads" class="ui-ads">
 <div id="inneractiveads"></div>
        <script language="javascript" type="text/javascript" src="http://cdn2.inner-
active.mobi/wv-js/inneractive_nokia_S40.js"></script>
 <script language="javascript" type="text/javascript">
  inneractive.parameters.divID = "inneractiveads";
  inneractive.parameters.appID = "MASUKKAN_APP_ID_ANDA_DI_SINI";
  inneractive.displayInneractiveAd();
 </script>
</div>
```

Kode tersebut me-load sebuah file JavaScript yang di dalamnya ada fungsi untuk me-load iklan Inneractive. Parameter `divID` adalah ID dari elemen tempat di mana iklan akan ditampilkan. Parameter`appID` adalah ID yang anda dapatkan setelah membuat aplikasi di situs Inneractive. Nah, fungsi `displayInneractiveAd()` dipanggil lalu iklannya akan diambil dari internet dan ditampilkan di aplikasi anda.

Untuk mengetes aplikasi anda, tekan **CTRL+R**. Hasilnya kurang lebih seperti gambar di bawah ini :

(C) Copyright Nokia 2014. All rights reserved.



nneractive memiliki panduan resmi / Official guide mengenai pemasangan iklan di Nokia Asha Web Apps di sini. Di sana mungkin terdapat informasi terbaru tentang cara memasang iklan di Nokia Asha Web Apps.

Penyedia iklan yang lain

Penyedia iklan yang lain meliputi Google Adsense dan Vserv.

Google Adsense menggunakan tag `<iframe>` , di mana tag tersebut tidak didukung di Nokia Asha Web Apps. Lagipula, Google Adsense for Mobile itu ekslusif untuk website mobile. Jika aplikasi anda tersedia sebagai sebuah website, maka tidak ada masalah, tetapi jika aplikasi anda digunakan secara independent dari sebuah website dan aplikasi anda di hosting secara menyeluruh di Nokia Store, maka anda tidak dapat menggunakan Adsense.

Hal yang sama berlaku pada ads untuk mobile sites dari Vserv, iklan-iklan dari Vserv tidak akan berfungsi di Nokia Asha Web Apps karena Vserv mengecek website yang menggunakan iklan mereka. Karena sebuah Nokia Asha Web Apps tidak punya URL yang pasti, maka tidak dimungkinkan menggunakan Vserv. Ada kemungkinan untuk me-request iklan dari sebuah Application Zone, tetapi ketika saya (penulis asli) mencobanya, tidak ada iklan yang ditampilkan.

Ada banyak pilihan untuk pemasangan iklan di Series 40 Web App, tetapi sejauh yang saya tau, Inneractive lah yang paling mudah digunakan ketika saya menulis artikel ini

Penempatan Iklan (Positioning)

Pada dasarnya, ada dua pilihan dalam meletakkan iklan: Menyatu dengan konten aplikasi anda, atau secara terpisah. Kita akan mencoba melakukan kedua pilihan tersebut, tapi pertama-tama mari kita buat sebuah view system yang dapat memudahkan anda ketika memilih kedua pilihan tersebut.

Membuat Aplikasi dengan Mulitple Views

Pertama-tama, buat sebuah file CSS baru dan beri nama **multi_view.css** di folder s40-theme/css dan import atau masukkan file css tersebut di file **index.html.** Kita akan menggunakan file ini untuk memberi style pada kelas-kelas yang akan kita buat.

Masukkan kode di bawah ini pada file **multi_view.css**

```
.ui-ads {
 width: 100%;
 text-align: center;
 height: 80px;
}
```


(C) Copyright Nokia 2014. All rights reserved.

```
.view {
 overflow: scroll;
 display: inline-block;
 vertical-align: top;
}

.ui-content {
 overflow: hidden;
        width: 1000px;
}


html, body {
 overflow: hidden;
}
```

Kelas view adalah kelas yang terpenting. Kelas ini akan digunakan untuk menandai bahwa sebuah `<div>` adalah sebuah view di aplikasi kita. Anda mungkin bertanya-tanya kenapa `ui-content` memiliki lebar 1000px. Alasannya adalah karena kita akan meletakkan view kita secara bersampingan (side-by-side) dan menggunakan sebuah transisi untuk membuat view tersebut menjadi lebih halus, dan lebar dari `ui-content` harus lebih besar dari jumlah lebar semua view kita.

Satu hal lagi, karena Web Apps dapat berjalan pada dua tipe layar, portrait (240px) dan landscape (320px), maka sangat perlu untuk membuat kelas baru lagi untuk mengindikasikan ukuran view kita. Template “Basic App” sudah menyediakan dua file CSS yang berbeda yang dapat berjalan bergantung pada ukuran layar HP/device. Kedua file tersebut adalah **single_landscape.css** dan **single_potrait.css**, keduanya berada di folder s40-theme/css. Buka kedia file tersebut.

Tambahkan kode berikut ke **single_landscape.css**

```
.full-width {
 width: 320px;
}
```

Tambahkan kode berikut ke **single_portrait.css**.

```
.full-width {
 width: 240px;
}
```

Dengan begini, kelas **full-width** akan membuat elemen kita mengambil alih semua layar yang ada (full screen). Ganti code pada ui-content div pada file **index.html** dengan kode di bawah ini. Mohon diteliti kalau di dalam elemen div tersebut tidak boleh ada spasi atau line breaks antara `</div>` dan `<div>` yang baru. Satu lagi, tambahkan id `#header` pada elemen yang memuat judul aplikasi anda.

```
<div id="content" class="ui-content">
  <div class="view full-width">
   <h1>View 1</h1>
  <input type="button" name="button1" class="ui-button" value="Go to View 2" />
  </div><div class="view full-width">
   <h1> View 2</h1>
  <input type="button" name="button1" class="ui-button" value="Go to View 1" />
  </div>
</div>
```

HTML ini membuat dua view yang memenuhi seluruh lebar dari device/perangkat anda. Jika anda menjalankan aplikasinya

(C) Copyright Nokia 2014. All rights reserved.

sekarang, hanya view pertama yang akan terlihat. Untuk membuat view kedua menjadi terlihat, kita dapat menggunakan properti `margin-left` pada `<div>` yang memiliki id `#content` . Jika kita set menjadi -100%, view yang kedua akan terlihat. Maka kita perlu menambah 2 kelas lagi di 2 file css.

Tambahkan kode berikut ini ke file **single_portrait.css**:

```
.view1 {
 margin-left: 0px;
 -webkit-transition: margin-left 0.5s linear;
}

.view2 {
 margin-left: -240px;
 -webkit-transition: margin-left 0.5s linear;
}
```

Tambahkan kode berikut ini ke file **single_landscape.css**:

```
.view1 {
 margin-left: 0px;
 -webkit-transition: margin-left 0.5s linear;
}

.view2 {
 margin-left: -320px;
 -webkit-transition: margin-left 0.5s linear;
}
```

Langkah terakhir adalah membuat agar tombolnya bekerja. Kita akan menggunakan fungsi dari Library MWL. Fungsi yang akan kita gunakan adalah `mwl.switchClass`. Di button yang pertama, tambahkan kode berikut:

```
onclick="mwl.scrollTo('#header');mwl.switchClass('#content', 'view1', 'view2');"
```

Tambahkan kode berikut pada button yang kedua:

```
onclick="mwl.scrollTo('#header');mwl.switchClass('#content', 'view2', 'view1');"
```

Pada kode tersebut, perubahan pada properti `margin-left` akan terlihat lebih halus karena ada transisinya, memberikan efek yang indah pada pergantian view dari satu ke dua atau sebaliknya. Di bawah ini adalah gambar-gambar ketika aplikasi ini dijalankan di potrait full touch dan landscape simulator.

(C) Copyright Nokia 2014. All rights reserved.

Tampilan awal ketika dites pada sebuah perangkat full touch device.

Tampilan awal ketika dites pada sebuah perangkat Non-Qwerty.

Iklan / Ads menyatu dengan Konten

Untuk membuat ads menyatu dengan konten, cukup dengan menaruh `<div>` sesuai dengan tempat yang anda inginkan. Pilihan yang sering digunakan adalah meletakkannya di dasar layar setelah konten, atau di atas layar sebelum konten. Kedua pilihan ini dapat dicapai secara mudah dengan view system yang barusan kita buat.

Untuk menaruh ads di atas, tinggal copy dan paste kode ads yang pada bagian awal telah kita buat, tepat sebelum element `#content` dengan id content. Dengan cara ini, pada view manapun anda berada, ads tersebut akan selalu terlihat.

Meletakkan ads di bawah konten sedikit disesalkan. Adsnya memang kelihatan tapi kontennya yang terlihat lebih dulu, dan si user bisa saja tidak melihat ads tersebut karena konten yang terlalu panjang. Karena itu, ini bukanlah posisi yang disarankan.

Iklan / Ads yang terpisah dengan Konten

Ini adalah peletakkan yang direkomendasikan oleh Inneractive. Intinya adalah meletakkan ads diantara view-view pada aplikasi anda, hati-hati untuk tidak terlalu banyak meletakkan iklan (user tidak akan senang dengan hal ini).

Catatan: View yang ada ads di dalamnya harus memiliki tema / komposisi warna yang sama dengan aplikasi anda. Hal ini agar ads tersebut terlihat natural atau terlihat seolah-olah bagian dari aplikasi anda.

Kita akan membuat ads memenuhi seluruh layar dalam dua fase. Fase pertama iklan akan diperlihatkan tanpa transisi. Di fase kedua kita akan menambahkan transisi yang halus untuk memperlihatkan ads nya.

Untuk membuat hanya ads saja yang muncul di layar, kita akan mengatur tinggi dari elemen dengan id `#content`menjadi 0 (kosong), membuatnya menjadi tidak terlihat. Lalu kita ubah elemen dengan kelas `ui-ads` menjadi terlihat. Buka file **multi_view.css** dan tambahkan kelas-kelas berikut :

```
.ui-ads-hide {
 display: none;
}

.ui-ads-show {
 display: block;
}

.ui-content-hide {
 height: 0px;
}
```


(C) Copyright Nokia 2014. All rights reserved.

```
.ui-content-show {
 height: auto;
}
```

Kelas-kelas tersebut mengatur tampak atau tidaknya si konten dan si ads/iklan. Nah, sekarang tambahkan `.ui-ads-hide` ke `ui-ads` . Jika anda menginginkan untuk menampilkan iklan ketika tombol pertama di klik, ganti event-handler `onClick` dengan yang ini :

```
onclick="mwl.switchClass('#content', 'ui-content-show', 'ui-content-hide');
mwl.switchClass('#ads', 'ui-ads-hide', 'ui-ads-show');mwl.switchClass('#content',
'view1', 'view2');"
```

Untuk menambahkan ad untuk setiap tombol, tambahkan saja kode di bawah ini pada handler `onClick` :

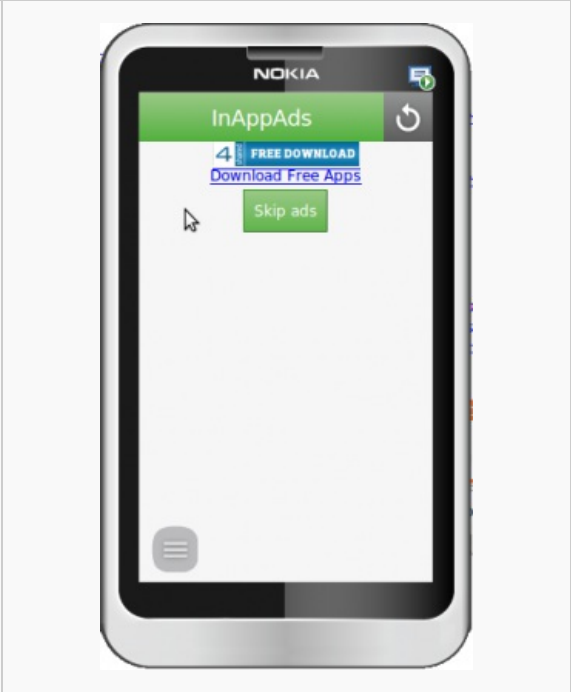
```
mwl.switchClass('#content', 'ui-content-show', 'ui-content-hide');
mwl.switchClass('#ads', 'ui-ads-hide', 'ui-ads-show');
```

Sekarang, ads akan tampil ketika user mengklik button. Tetapi, tidak ada cara bagi user untuk meng-skip ad tersebut dan melanjutkan ke view selanjutnya. Mari kita buat tombol skip, tambahkan kode berikut di akhir elemen dengan id `ads`

```
<input type="button" name="button1" class="ui-button" value="Skip ads"
onclick="mwl.switchClass('#content', 'ui-content-hide', 'ui-content-show');
mwl.switchClass('#ads', 'ui-ads-show', 'ui-ads-hide');"/>
```

Hasilnya :

View Pertama



View Iklan / Ad

(C) Copyright Nokia 2014. All rights reserved.

View kedua

Kode di atas bekerja dengan baik, tapi transisi dari satu view ke view yang lain sangat kasar. Kita akan menambahkan transisi yang smooth/lembut, seperti transisi yang sudah kita buat sebelumnya. Ini akan membutuhkan lebih banyak kode karena kita perlu menentukan secara pasti tinggi dari sebuah view yang kita buat. Buka file **multi_view.css** dan tambahkan kode di bawah ini:

```
.view1-height {
 height: 450px;
 -webkit-transition: height 0.5s linear;
}

.view2-height {
 height: 300px;
 -webkit-transition: height 0.5s linear;
}
```

Sekarang, tambahkan kelas `ui-ads-hide` ke elemen dengan id `ads` dan wrap/bungkus elemen dengan id `content` dengan sebuah elemen div baru, seperti contoh di bawah ini.

```
<div id="content_wrapper" class="full-width view1-height">
        <div id="content" class="ui-content">
  <div class="view full-width">
    <h1>View 1</h1>
    <h1>View 1</h1>
    <h1>View 1</h1>
    <h1>View 1</h1>
    <h1>View 1</h1>
   <h1>View 1</h1>
    <h1>View 1</h1>
    <h1>View 1</h1>
    <h1>View 1</h1>
    <h1>View 1</h1>
    <h1>View 1</h1>
    <h1>View 1</h1>
    <h1>View 1</h1>
```


(C) Copyright Nokia 2014. All rights reserved.

```
    <h1>View 1</h1>
    <h1>View 1</h1>
    <input type="button" name="button1" class="ui-button" value="Go to View 2"
onclick="" />
         </div><div class="view full-width">
    <h1> View 2</h1>
   <input type="button" name="button1" class="ui-button" value="Go to View 1" onclick=""
/>
         </div>
        </div>
</div>
```

Ganti button di elemen `#ads` dengan `div`, seperti di bawah ini :

```
<div id="ads" class="ui-ads ui-ads-hide">
 <div id="inneractiveads"></div>
 <script language="javascript" type="text/javascript">
  inneractive.parameters.divID = "inneractiveads";
  inneractive.parameters.appID = "Montagner_IsItDown_Nokia";
  inneractive.displayInneractiveAd();
 </script>
 <div id="btnAdSkip">

 </div>
</div>
```

Langkah terakhir, buat sebuah file javascript bernama **ads.js** dengan isi di bawah ini : (jangan lupa untuk import file ini di **index.html**)

```
function set_ad_return(cl) {
 var btn = document.getElementById("btnAdSkip");
 btn.innerHTML = "<input type=\"button\" id=\"btnAdSkip\" class=\"ui-button\"
value=\"Skip ads\" onclick=\"mwl.switchClass('#ads', 'ui-ads-show', 'ui-ads-hide');
mwl.switchClass('#content_wrapper', 'ui-content-hide', '" + cl + "');
mwl.timer('reset_ad_return', 1000, 1, 'mwl.switchClass(\'#btnAdSkip\', \'ui-ads-show\',
\'ui-ads-hide\'); inneractive.displayInneractiveAd();');\" />";
 mwl.switchClass('#btnAdSkip', 'ui-ads-hide', 'ui-ads-show');
}
```

Fungsi ini membuat sebuah **"Skip Ads"** button di elemen dengan id `ads` yang mengarahkan anda ke view yang sesuai. Ketika button di klik, aplikasi kita akan me-request ads baru, maka akan ada ads yang berbeda. Inneractive menyarankan bahwa ads ditampilkan sesudah user menggunakan aplikasi. Di dalam games, ini berarti ads harus diletakkan ketika user telah mencoba bermain game anda. Di Aplikasi biasa, ads ditampilkan ketika user sudah selesai melakukan sesuatu dengan aplikasi anda. Kita akan mencoba menambahkan ads hanya pada transisi dari view dua ke view satu

Tambahkan kode berikut ke button pertama :

```
onclick="mwl.switchClass('#content_wrapper', 'view1-height', 'view2-height');
mwl.scrollTo('#header'); mwl.switchClass('#content', 'view1', 'view2'); "
```

Tambahkan kode berikut ke button ke dua :

```
onclick="mwl.switchClass('#content_wrapper', 'view2-height', 'ui-content-hide');
mwl.switchClass('#ads', 'ui-ads-hide', 'ui-ads-show'); mwl.switchClass('#content',
```


(C) Copyright Nokia 2014. All rights reserved.

```
'view2', 'view1');mwl.timer('ad_skip', 5000, 1, 'set_ad_return(\'view1-height\');');
```

Perbedaan utamanya adalah kali ini kita menambahkan timer untuk membuat tombol "Skip Ads" akan muncul setelah lima detik. Hal ini membuat user melihat ads/iklannya sebentar sebelum dia dapat melanjutkan kegiatannya. Di bawah ini adalah contoh video dari aplikasi yang telah kita buat di Nokia C3-00. Kualitasnya tidak terlalu baik, tapi video tersebut menggambarkan bahwa hasilnya sangat baik, bahkan pada perangkan lama seperti C3-00.

Template yang dijalankan di Nokia C3-00.

Tips dari Inneractive

Inneractive memiliki sebuah halaman website dengan tips dan contoh tentang peletakan iklan.

Lebih men-detailkan Ads anda

Langkah terakhir dalam artikel ini adalah menyediakan informasi ekstra yang dikirimkan ke Inneractive agar ads yang di dapat lebih relevan terhadap penggunanya. Dengan menggunakan namespace `inneractive.parameters`

Parameter yang ada:

- Umur User. Buat properti `inneractive.parameters.age`.
- Jenis Kelamin User. Buat properti `inneractive.parameters.gender`.
- Keyword/Kata Kunci - dipisahkan oleh koma. Buat properti `inneractive.parameters.keywords`.
- Kategori Aplikasi. Buat properti `inneractive.parameters.category`
- Lokasi Pengguna - Dipisahkan oleh koma. Ini terdiri dari Negara, Wilayah dan kota. Negara dapat diambil dari properti `navigator.homeCountry` . Lihat artikel ini untuk informasi lebih lanjut tentang properti ini

Buat properti ini `inneractive.parameters.location` untuk mengirim informasi negara.

GPS Location

Web Apps dapat mengetahui posisi device menggunakan API geolocation W3C's. Di bawah ini adalah contoh untuk membuat `inneractive.parameters.GPSLocation` dengan posisi device yang didapat dari API ini. Parameter ini akan dikirim di request selanjutnya ketika posisi sudah di dapat.

```
function setGPSPosition(position) {
      inneractive.parameters.GPSLocation = position.coords.longitude + "," +
position.coords.latitude;
}

navigator.geolocation.getCurrentPosition(setGPSLocation);
```


(C) Copyright Nokia 2014. All rights reserved.

Kesimpulan

Artikel ini menunjukkan sebuah cara untuk membuat aplikasi Web Series 40 yang memanfaatkan API In-App Advertising untuk memonetize aplikasi anda. Contoh Kode : File:Template.wgt.

(C) Copyright Nokia 2014. All rights reserved.