

移动化你的 Web 服务

本文章翻译自《[Mobilising your Web Service](#)》

内容

[\[隐藏\]](#)

- 1_关于本文
- 2_为什么要对网站实施移动化
- 3_不同的移动浏览器和终端
 - 3.1_版面与大小
 - 3.2_浏览器功能
 - 3.3_带宽和数据流量服务计划
- 4_侦测客户端浏览器
 - 4.1_是否移动浏览器
 - 4.2_移动终端的类型
- 5_选择一种移动内容的提供方法
 - 5.1_方法 1. 设计一个简单的可缩放版面
 - 5.2_方法 2. 在移动和桌面版式间切换
 - 5.2.1_选择不同的 CSS
 - 5.2.2_隐藏并不阻止下载
 - 5.3_方法 3. 侦测后重定向
 - 5.4_一个一般的移动网站
 - 5.5_RSS
 - 5.6_终端特定网站
- 6_针对移动用户选择所显示内容
 - 6.1_了解你的用户
 - 6.2_省略还是添加?
 - 6.3_设计过程中时时考虑移动
- 7_使用转换编码器
- 8_如果使用内容管理系统
 - 8.1_Joomla
 - 8.2_Drupal
 - 8.3_Wordpress
- 9_范例
 - 9.1_移动化的 web 网站
 - 9.2_代码片段

[9.2.1_W3C 学校](#)

[9.2.2_JavaScript](#)

[9.2.3_ASP.NET](#)

[10_更多信息](#)

关于本文

本文讲述对网站实施移动化的一些方法。本文并非简明教程，而是工作文档，也提供针对该主题的其他资源的链接。

为什么要对网站实施移动化

今天，互联网用户越来越多地从各种不同的位置享用万维网。同一个用户经常会通过宽带连接从桌面电脑或通过智能手机访问某项服务或某个网站。对这些用户来说，很重要的一点是，不管屏幕大小和输入方式的不同，其享有的服务应具有协调一致的用户体验。

另外，那些只通过或基本上只用移动终端访问互联网的用户数量在不断增加中。移动终端不仅是另一种媒体浏览工具，它还具有更多功能。大多数支持互联网的手机都配有相机，它们还被用作个人数字助理，并将诸如日历和名片夹数据库等与消息服务相整合。最重要的是，只需按键，移动电话就可立即拨号通话。通过连网的 **widgets** 还可利用终端上的位置信息，大大有助于向用户提供目标化本地服务。

不同的移动浏览器和终端

今天的终端有大有小、有简有繁。这意味着，不仅用户数巨大，而且用户各异。最简单的终端只能在其小屏幕上显示基于文本的 **html** 页面，而最先进的终端其浏览器与桌面机并无二致，除了物理尺寸。

请至[诺基亚论坛网站](#)查找所有诺基亚终端及其规范。

版面与大小

最常用的便携式屏幕解析度是 **QVGA**（240 X 320 像素）。越来越多的终端使用更高的解析度，但考虑到移动性，终端屏幕的物理大小是有限度的。所以，即使解析度更高，显示更清晰，屏幕尺寸却不能大过可携带的尺寸。事实上，在常用的 **QVGA** 解析度中也包含了大量的物理尺寸。这里有一篇很好的[文章](#)介绍了不同显示屏的解析度流行情况。

为应对复杂性，终端可以是纵向或横向显示的，也可以同时支持这两种显示模式。这意味着，终端或者具有一种固定的版面格式，如只支持横向显示模式的[诺基亚 E63](#)，或者只支持纵向显示模式，如[诺基亚 6212 classic](#)。另一种选项则是让终端能在横向和纵向显示模式间切换，既可以由用户手动控制，也可以自动实现。用户控制动作可以如滑移[诺基亚 N95](#)的屏幕，或如打开[诺基亚 E90 通讯器](#)。后一种动作更具挑战性，因为要从 **QVGA** 解析度切换到通讯器终端特有的 **800x352** 像素。自动切换意味着终端内的传感器会感知其在用户手中的持有位置，并会相应切换显示屏的方向，如[诺基亚 5800 XpressMusic](#) 或 [诺基亚 N85](#)。

Browser capabilities 浏览器功能

浏览器功能也各不相同。如前文所述，支持 **web** 的终端其功能繁简各异。对网站开发者来说这意味着以下两点：

不要期望移动浏览器能具有桌面浏览器的同样功能。

不要假设所有移动终端都只需要文本 **html** 页面。

这意味着，当为移动用户设计服务或网页时，最好总是能提供至少两种选项：针对功能手机的简单文本版，以及针对高端终端的较高级型。

带宽和数据流量服务计划

终端所带来的限制不仅仅是图型和富内容的数量，还有网络。即使终端本身支持显示 Flash 和流媒体以及动态版面、渐变效果和所有显示特效 — 用户可能并没有数据包月计划或者他/她目前正处于漫游状态。

许多地区甚至并不提供数据包月服务。在这种情况下许多用户会选择关闭图像和 Flash 内容的显示，除非通过无线局域网接入点进行浏览。

侦测客户端浏览器

所以，鉴于前文所讨论的所有问题，向使用移动电话而非个人电脑访问你的网站的用户提供卓越的可用性并选择正确的内容向其显示确实是一项挑战。解决方案是，虽然不算简单，但至少可行 — 浏览器和终端侦测。

是否移动浏览器



最简单的方法是使用各浏览器所提供的 [user agent string](#)。它告诉 web 服务：用户使用的是哪一种浏览器，并在大多数情况下也告知：用户使用的是哪一种设备。可惜的是，[user agent string](#) 中并没有一种标准方法会告诉“这是一台手机”，所以还需要一些额外工作。侦测功能可通过服务器端脚本实现，比如 [PHP](#)，也可以通过客户端 [JavaScript](#) 实现。这里是一个可以侦测出所有诺基亚终端的 JavaScript 范例。它利用了 [user agent](#) 中所包含的“nokia”。

```
function isItNokia(){  
    if (/^(Series60|Nokia|SymbianOS)/i.test(navigator.userAgent)){  
        return true;  
    } else {  
        return false;  
    }  
}
```

有关使用 JavaScript 侦测终端的更多范例：[\[1\]](#)。

这当然是一个非常轻量级的方法，只适用于还没有更强大更广泛的终端侦测方法时。

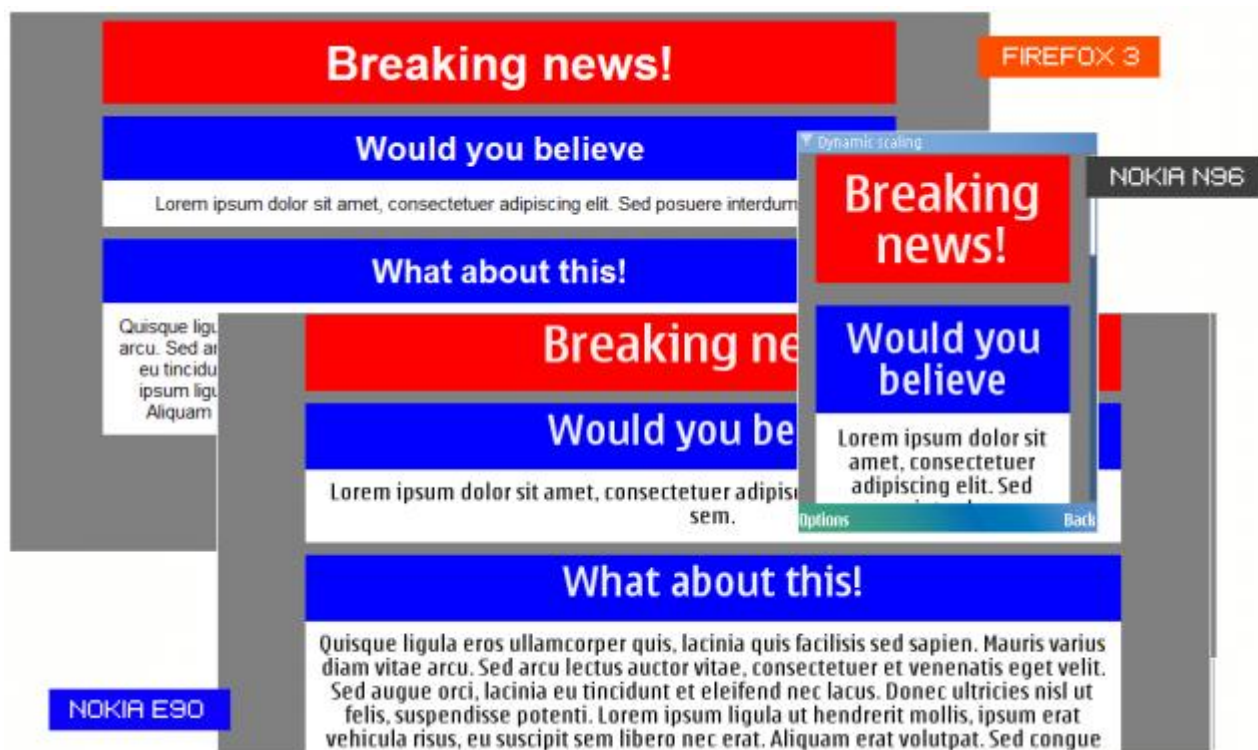
移动终端的类型

下一步就是检测访问者所使用的终端类型。这方面有几种解决方案，文档《[在 Web 服务中侦测移动终端](#)》中对此有详细介绍，因此本文不再细述。

选择一种移动内容的提供方法

现在，针对移动的内容已经就绪，我们甚至已知道用户的终端类型，是时候决定如何向移动浏览器提供经过调整的内容了。简而言之，各种手段在此被分成三种不同的方法。现实中，内容提供方法当然取决于所使用的技术，而且可能是几种方法的组合，或者全然不同。虽然如此，下列方法可以提供一个通用思路。

方法 1. 设计一个简单的可缩放版面



最简单的方法之一，而且适用于非常简单的网页，是在页面上使用相对定位和缩放。比如，内容中不使用绝对像素宽度，而是定义相对宽度，并让高度随内容而缩放。下文范例中，用一些颜色来强调，字体的高度是用毫米定义，而每一元素则给出了一个相对宽度。这意味着页面可以缩放到全屏显示，至于屏幕中的一毫米是否就真是一毫米则由终端和浏览器供应商定义。比如，在诺基亚 S60 5th edition 终端中，10mm 被解释为 5mm。

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>Dynamic scaling</title>
```

```
<style type="text/css">
```

```
* {  
  font-size:10mm;  
}  
body {  
  font-family:sans-serif;  
  text-align:center;  
  background-color:gray;  
}  
h1, h2, p, a {  
  width:80%;  
  display:block;  
  font-weight:bold;  
  padding:10px;  
  margin:0 auto;  
  color:white;  
  font-size:70%;  
}  
h1, p {  
  margin-bottom:10px;  
}  
h1 {  
  background-color:red;  
  font-size:100%;  
}  
h2 {  
  background-color:blue;  
}  
p {  
  background-color:white;  
  color:black;  
  font-size:40%;  
  font-weight:normal;  
}  
a {  
  width:50%;  
  background-color:green;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1 name="top">Breaking news!</h1>
```

```
<h2>Would you believe</h2>
```

```
<p>
```

```
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

```
  Sed posuere interdum sem.
```

```
</p>
```

```
<h2>What about this!</h2>
```

```
<p>
```

```
  Quisque ligula eros ullamcorper quis, lacinia quis
```

```
  facilisis sed sapien. Mauris varius diam vitae
```

arcu. Sed arcu lectus auctor vitae, consectetur et venenatis eget velit. Sed augue orci, lacinia eu tincidunt et eleifend nec lacus. Donec ultricies nisl ut felis, suspendisse potenti. Lorem ipsum ligula ut hendrerit mollis, ipsum erat vehicula risus, eu suscipit sem libero nec erat. Aliquam erat volutpat. Sed congue augue vitae neque. Nulla consectetur porttitor pede.

</p>

Back to top

</body>

</html>

将CSS数据嵌入到HTML中只是为了考虑下载时间和用户体验。独立样式表得在载入html之后才能载入，对于较慢的连接这意味着页面最初显示时没有任何样式，这将使用户感到困惑。这个范例十分基础，但可以用图像和诸如边界半径、下拉阴影等CSS3效果加以扩展。对不支持这些特效的终端来说，这些都是透明的，并且会被忽略。

这类方法的优点是无需任何终端或浏览器侦测，但在桌面端则可能显得不够活泼。尽管如此，这种方法对一般的移动页面可算是不错的选择。如果页面是在服务器端用比如PHP创建的，则有更多的方法能确保内容符合给定的终端。

方法 2. 在移动和桌面版式间切换



如果移动和桌面版本之间要求更大的差异性，但内容却大致相同，最好的办法就是各自使用不同的样式。

有几种实现方法，文档《[移动化](#)》中讲述了一些方法。

选择不同的 CSS

有一种可动态切换样式表的选项。一种方法就是使用两个相互独立的 `css` 文件，并在侦测出移动浏览器之后通过 `JavaScript` 指向移动版本。范例如下：

首先，在 `html` 文件的 `head` 区中定义了两个样式表。两者都被给予 `title` 属性，以彼此区分。

```
<head>
...
<link rel="stylesheet" type="text/css" href="desktop.css" id="desktop" />
<link rel="stylesheet" type="text/css" href="mobile.css" id="mobile" />
...
</head>
```

然后，用到了前文所讲述的终端侦测方式，一个 `JavaScript` 函数会根据 `title` 属性切换样式表。

```
function switchCSS(media){
  var obj = media == 'desktop'? 'mobile': 'desktop';
  document.getElementById(obj).disabled = true;
}
```

在该页面的移动版本中，网站中某些不适应移动终端显示的内容可被切换，也可被完全略去。而且，确保字体足够大也是一种良好的设计思路。

如果终端侦测算法足够智能，并使用了诸如 [WURFL](#) 或 [Handset Detection](#) 这样的服务，那么网站还会知道该终端是否支持触摸。在这种情况下，应确保链接和其它可点击元素足够大，彼此间隔也足够，以便手指操作，这也不失为一种良好的策略。

隐藏并不阻止下载

要牢记的是，如果该页面具有大型图像、`flash` 和其它元素，如果只是将它们隐藏起来，它们仍会被载入到终端中。因此，在大多数情况下，最好依靠服务器端脚本程序首先根据已侦测到的终端选择好内容，然后根据某个恰当的模板提供合适的内容。

方法 3. 侦测后重定向

有时候，分别为移动用户和桌面用户维护彼此独立的网站也是一种好方法。这样做很有效，比如，如果移动网站所提供的服务与桌面版本的差异较大。还有一种可能，网站所有者希望使用不同的域名，比如针对桌面浏览器的 `.com` 或 `.org`，以及针对移动的 `.mob`。

经常访问的用户可以得到指点，将该网站加入书签，而终端侦测算法只用于确保首次访问用户不致迷失方向。在重定向到某个移动网站时，询问用户是否真希望如此，甚至使用 `cookies` 记住该用户的决定，这也不失为一种良好策略。事实上，如果移动网站很小很直观，一些用户甚至在移动终端上也宁愿访问桌面网站。

一个一般的移动网站

即使将用户重定向或指引其到 `.mobi` 或 `m.site.com`，在多数情况下侦测所使用终端和浏览器仍不失为一个好主意。正如前文所解释的，终端和用户类型多种多样，很难用一个网站服务每一位用户。基于文本的网站并非好主意，但如果网站的内容量比较大，那么使用统一费率的用户及不使用包月数据流量服务的 `2G` 功能手机用户对下月账单会惊诧莫名。

因为最后决定在于用户自己，所以最好不要做太多的假设，而应该向他们提供各种选项。尽管开发者应该牢记用户通常也不希望面对太多的选择。好的解决方案是：执行一种静默且精准的终端侦测，只在无法确定时提供选择。也可以在网站顶端提供指向较

简洁或较繁复版本的链接。有时，甚至笔记本电脑用户也更喜欢移动版本，如果他们是通过手机在[EDGE](#)网络中漫游接入网络的话。

RSS

[RSS](#)阅读器正成为智能手机中的标准项。因此，为移动用户提供对你的 feeds 的链接是个好主意。而有时，提供某个网站的移动版本的最简便方法就是将移动用户直接导向 [RSS feeds](#)。所有最新的移动 RSS 阅读器都能定期地下载内容，如果不需要持续在线，这将节省用户成本，延长电池使用寿命。

终端特定网站

结合智能内容选择的终极性终端侦测有可能被在线游戏店铺使用。当你首先准确地检测出终端的制造商和型号时，你就可以向用户提供绝对适合其终端的预选软件。这可能极大地促进销售，因为用户不必在大量无法运行的应用中费心费力进行搜索。

针对移动用户选择所显示内容

正如前文所述，有几种方法可用于提供适用于移动浏览器的内容。无论你决定采用轻量级方案并摒弃所有图像，抑或严重依赖终端侦测并根据各终端款型行事，你还是需要确定实际的内容和展示格式。一种方法是：让某家专业咨询公司开展市场调研。或者你自己跑去询问自己的用户：他们希望在网站上看到什么。如果你考虑实施一项针对移动用户的问卷调查，请记住：只有很少的终端适合输入较长的回答文本，所以请将问卷做得尽可能简短。

明白这些道理之后，无论以你自己的商业情报，还是通过雇佣服务，你都能更方便地提供能留住回头客的内容。

了解你的用户

终端侦测的一种可能用途是静静检测每一个访问你网站的浏览器。许多内容管理系统已提供这类功能，你自己用 [PHP](#) 实现也不难。你只需将数据保存到服务器端。当你分析这些数据时会了解到：使用移动终端的用户有多少、终端类型是什么、你网站上的哪些部分是他们访问最多的。

这种数据首先可用于理解初始内容，但慢慢地，当你开始推广一项针对移动而优化的 [web](#) 服务时，能了解清楚用何种方法来进一步开发自己的网站，这点更显重要。

省略还是添加？

有关移动网站的一个常见误解是：它们只是[WAP](#)页面，是真正网站的一个子集。这当然也对。但这样的话，你就失去了为你的移动用户提供适合其网络化生活的机会。一个简单的 [WAP](#) 网站已可提供一些特定服务，如安全的运营商账单业务和分离分发[DRM](#) 保护。当你知道正在浏览你网站的特定终端的能力时，大量的潜在业务已经初露端倪。[SMS](#) 支付即是其中之一，直接向手机发送特价券、或为电话号码提供一个可点击的链接等功能则比比皆是。当你的用户使用 [GPS](#) 照相终端时，组织竞赛及获取用户生成的内容将越发有趣。

设计过程中时时考虑移动

当你设计一项新服务或一个新网站时，你应该同时设计桌面网站和移动网站。这不仅因为：从头开始功能设计比事后弥补更方便，而且也因为：你还可以在自己的移动网站上重用桌面组件而额外受益。假设你计划在自己的所有页面上都增加一个侧面导航 [widget](#)，为什么不将它设计成针对移动环境的 240x320 像素、字体足够大、并便于缩放呢？这样的话，无需任何修改，你的侧面导航部分就能作为你的移动首页。其它元素也是如此——多考虑移动性，从而帮助自己省时省钱。如果你的网站组件可以逐个显示而非同时出现，且贴合移动电话显示屏，那你就已经拥有了该网站的移动版本。

要记住的另一点是：移动受众和终端所具有的附加值，关于这方面请参阅前面的章节。只需稍加考虑，你就能从头开始通过利用用户的移动特性设计自己的服务。

使用转换编码器

获取移动版网站的另一种方法就是将自己的内容输入到一个转换编码器中。转换编码器是一种服务，它将输入其中的内容进行缩容和修改，使之更适应移动终端。转换编码器中也可以使用终端侦测以帮助确定正确的格式。其高级版本甚至能对视频进行降级。

使用转换编码器的好处是：内容会变得只有几千字节那么小，从而能更快地被移动终端加载。你既可以自行创建一个转换编码器，也可以使用某个商用代理服务器。最主要的是，内容在被下载到移动终端之前已经得到了缩减。

一些商用转换编码器服务的范例是 [Skweezer](#)、[Teashark](#) 和 [Opera Mini](#)。

如果使用内容管理系统

[CMS](#)是日益流行的网站维护方法。特别对于一些大型的组织机构。一些最为流行的内容管理系统都为自己开发了移动插件。这些插件身后的理念与本文所阐述的并无二致。下面列出了对当前一些可用插件和解决方案的链接：

Joomla

http://sourceforge.net/project/showfiles.php?group_id=252556

Drupal

<http://www.mobiledrupal.com>

Wordpress

<http://alexking.org/blog/2009/03/29/wordpress-mobile-edition-302>

范例

请随时访问下面的代码片段和范例网站链接。

Source: [Image:Mobileweb.zip](#)

资源: [Image:Mobileweb.zip](#)

移动化的 web 网站

About.com 上有移动网站十佳设计排行榜: <http://webtrends.about.com/od/mobileweb20/tp/Top-Mobile-Websites-.htm>

MobiThinking.com 上有[最佳和最差移动网站](#)排行榜，此外还有大量其它的[最佳实践白皮书](#)和[展示](#)目录。

代码片段

W3C 学校

<http://www.w3schools.com/>

JavaScript

<http://www.hand-interactive.com/resources/detect-mobile-javascript.htm>

ASP.NET

<http://www.asp.net/mobile/>

更多信息

http://www.forum.nokia.com/Resources_and_Information/Explore/Web_Technologies/

http://en.wikipedia.org/wiki/Mobile_browser

<http://www.w3.org/Mobile/>

--[Ruikku](#) 11:23, 21 April 2009 (EEST)

--[Ruikku](#) 2009年4月21日, 11:23

如何同时使用 Symbian C++和 QT S60 代码

本文章翻译自《[How to use Symbian C++ and Qt S60 Code together](#)》

ID	...	创建日期	2009年1月15日
平台	S60 3rd Edition FP1、S60 3rd Edition FP2、S60 5th Edition	真机测试	模拟器
分类	Qt for S60	子类	应用

关键词 (APIs、类、方法、函数) : Qt、Symbian C++

概述



本节演示如何同时使用 Symbian C++和 Qt for S60。本节内容即将更新。

预备知识

从 [Qt for S60 "Garden" pre-release](#) 安装 Qt for S60 Garden 版

点击此链接下载安装指南: [如何安装该文件包](#)。

参阅该文: [Qt for S60 入门](#)

下载 Qt 创建器 IDE: [QtIDEv4.5](#)

初步

如何启动 Symbian OS 编程? [入门](#)

如何开始使用 Symbian Platform_macros? [Platform_macros](#)
[转换描述符](#)

在 Qt S60 中如何使用 Symbian 描述符

[如何将 Tbuf 转换为 QString](#)

[如何将 HbufC 转换为 QString](#)

[如何将一个 Qt QString 转换为一个 S60 描述符](#)

注

新增了一些超链接和代码。



后继知识

在此使用了 symbian 描述符 Tbuf 和 HbufC，并将之转换为串。