

Archived:Getting Started With WidSets-Part2



Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template `{{ReviewForRemovalFromArchive|user=~~~~~|write your reason here}}`.

The article is believed to be still valid for the original topic scope.

Developing Widgets

Widgets are Web applications that can run independently from the browser application. In other words, widgets are Web pages, normally designed for a specific purpose. Widgets utilize standard based Internet technologies such as **HTML, CSS, JavaScript**, and even AJAX.

Some Important Terms

The following definitions explain several important terms as used in the context of Widgets developments:

- **Mobile widgets:** Widgets that run on Nokia mobile devices;
- **Web widgets:** Mobile widgets that run on the Nokia Web Run-Time platform;
- **Nokia Web Run-Time (WRT):** A technology that enables mobile widgets to run on Nokia mobile devices;
- **Widget engine:** A system component that provides programming interfaces to mobile widgets.

Prerequisite Knowledge

HTML markup and JavaScript are essential programming languages for the design and implementation of every widget. In addition, the Document Object Model (DOM) and Cascading Style Sheets (CSS) are also substantial technologies for controlling the style and layout of a widget. Nokia WRT supports the following Internet technology standards:

- HTML 4.01, XHTML 1.0 (basic and mobile profile)
- CSS
- JavaScript 1.5 (ECMA-262 3rd Edition)
- DOM

Widget Structure

Web widgets, unlike traditional Web applications, have their own structure and require certain components to be stored locally on a mobile device. The essential architectural differences between a widget and a normal Web application are the manifest file and the inclusion of only one HTML file. These two files are required in the widget installation package and will be stored locally on a device after a widget installation.

The following are typical widget component files and descriptions of their use:

- **Info.plist file**
- **.html file**
- **Icon.png**
- **.css file**
- **.js file**

Info.plist file

This is a mandatory component of a widget. It is in XML format, containing the property and configuration information of a widget. It has a defined structure. It contains following elements:

- **Info:** It contains general data about the widget such as version and encoding conventions.
- **Parameters:** It defines the parameters that can be used by the application.
- **Resources:** It contains the resources such as images, the script file and style sheets.
- **Layout:** It allow to create views using parameters in the xml file(minimized and maximized views)

.html file

This is the only HTML page of a widget, and a mandatory component in the widget package. The name of the HTML file must be predefined within the Info.plist. This file is a standard HTML file that mainly contains information for structuring a widget.

icon.png file

Installed widgets can be viewed on a mobile device screen (Application Menu) through a custom icon. An icon image must be in Portable Network Graphics (.png) file format. The size of the icon is recommended to be 88 by 88 pixels. If the icon.png file is omitted from the widget installation package, the widget will appear with a default S60 application icon.

.css file

The CSS file contains information for controlling the style and layout of a widget's contents. It is strongly recommended to keep the style information separated from the data (markup) elements. This is not a mandatory component of a widget; style information can be embedded within the HTML file, thereby allowing the CSS file to be omitted. A widget can have as many individual CSS files as needed.

.js file

The JS file contains JavaScript codes to perform widget UI interaction, communication operations, construction of UI elements dynamically, etc. JavaScript codes are best kept in a separate file. This is not a mandatory component of a widget; JavaScript codes can be embedded within the HTML file, thereby allowing the JS file to be omitted. A widget can have as many individual JS files as needed.

Installation Package

Once all the component file of the widget are created, the installation package should be produced to make the widget ready for installation on a device. The procedure is quite simple. To create an installation package for the Mobile widget, simply select a folder put all the component files within the folder, and then right click to zip and create the zip file. After creating the ZIP archive, manually rename the file extension from .zip to .wgz.

your widget is now ready to be installed...

Reference Link

[Archived:Getting Started With WidSets-Part1](#)

[Archived:Getting started with WidSets - Part3](#) --ck.umraliya 16:46, 5 September 2008 (EEST)