Archived:Example Image Service in WidSets

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.

This example shows how to use image service to fetch images from the Internet and scale them to fit to the mobile device's screen.

The easy way

This example uses *getPicture()* to automatically fetch the image.

image_service.he

```
class
{
  const int CMD_BACK    = 1;

  MenuItem BACK        = new MenuItem(CMD_BACK, "Back");


  void startWidget()
  {
    setMinimizedView(createMinimizedView("viewMini", getStyle("default")));
  }


  Shell openWidget()
  {
    Picture pict = getPicture("http://dev.widsets.com/images/widsets_dev_logo.gif");

    return new Shell(pict);
  }


  MenuItem getSoftKey(Shell shell, Component focused, int key)
  {
    if (key == SOFTKEY_BACK) {
      return BACK;
    }
    return null;
  }


  void actionPerformed(Shell shell, Component source, int action)
  {
    if (action == CMD_BACK) {
      popShell(shell);
    }
  }

} //class
```

You may use the same *widget.xml* with this.

A more complicated example

This example calls the image service with the *geturl* action in the traditional way and declares success and failure callbacks.

image_service.he

```
class
{
  void startWidget()
  {
    setMinimizedView(createMinimizedView("viewMini", getStyle("default")));
  }


  Shell openWidget()
  {
    fetch();

    return null;
  }


  void fetch()
  {
    int x, int y = getScreenSize();

    Value arg = [
      "url" => "http://dev.widsets.com/images/widsets_dev_logo.gif",
      "boundx" => x - 40,
      "boundy" => y - 40,
```


(C) Copyright Nokia 2013. All rights reserved.

```
      "format" => "png8"
    ];

    call(null, "imageService", "geturl", arg, ok, nok);
  }


  void ok(Object state, Value ret)
  {
    Image img = getImage(ByteArray(ret));

    //Bubble is the easiest way to show images. Normally,
    //one would instantiate a new Picture component and
    //place the Image inside it and then place the Picture
    //component in a flow.

    setBubble(img, "Success");
  }


  void nok(Object state, String error)
  {
    setBubble(null, "Error getting image: "+error);
  }

} //class
```

widget.xml

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

<widget spec_version="2.0">
  <info>
    <name>example_image_service</name>
    <version>1.0</version>
    <author>example</author>
    <clientversion>1.0</clientversion>
    <shortdescription>Image Service Example</shortdescription>
    <longdescription>Image Service Example</longdescription>
    <tags>example imageservice</tags>
  </info>

  <parameters>
    <parameter name="widgetname">Image Service</parameter>
  </parameters>

  <services>
    <service type="image" id="imageService"/>
  </services>

  <resources>
    <code src="image_service.he"/>

    <stylesheet>
      mini {
        background: solid white;
        color: black;
        align: vcenter hcenter;
      }

      maxi {
        background: solid white;
        padding: 5 5 5 5;
      }

      flow {
        background: solid white;
      }
    </stylesheet>
  </resources>

  <layout minimizedheight="2em">
    <view id="viewMini" class="mini">
      <label class="mini">${widgetname}</label>
    </view>
  </layout>

</widget>
```

Cache

Note: Images and other http resources may be cached by the servers to minimize traffic to destination servers. If you are fetching a Web page or an image that has an URL but changes often, do as follows:

Instead of *www.site.com/image.jpg* ask for *www.site.com/image.jpg* + "*?*" + *currentTimeMillis()*.
The following request will go through the cache but most Web servers will still return *image.jpg*.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.