Nokia Asha web apps - FAQ

This article provides a guide to Nokia Asha web apps. The article answers various frequently asked questions on how to use Nokia Asha web apps. Series 40 web apps offer an easy entry point to mobile app development where one can utilize their web developer skills to create content with HTML, CSS, and JavaScript code.

Getting started

17 Apr 2011

How do I get started?

- Visit getting started page on the Nokia Developer website and watch the web apps training videos.

What do I need to develop and test Nokia Asha web apps ?

- Download and install Nokia Asha Web App Tools 3.0 Beta, this tool provides for the development and simulation of web apps.
- Download and install this apps that enables deployment of apps from Nokia Asha Web App Tools to a device
  - Nokia Asha Platform: Use phone's browser to install USB Launcher app.
  - Series 40 platform: Bluetooth Launcher app.
- Read the tutorial and watch the videos on the Getting started with Asha web apps page on the Nokia Developer website.
- Create your first app.
- Alternatively, you can also try the Xpress Web App Builder tool.
- Watch the videos on the Getting started with Xpress Web App Builder page on the Nokia Developer website.

Support facilities

Where can I find documentation?

Nokia Asha Web Apps Library

Please see also Nokia Asha web app wiki articles

Where can I find example apps?

- Code examples on the Nokia Developer website
- Nokia Asha Web App Tools provides fully working web app project templates to help you get started.

How do I use the support channels?

- Nokia Developer staff and our friendly community members provide help on the Nokia Developer discussion boards.
- The Nokia Developer technical support channel provides professional support with a good service level agreement.

How do I report bugs?

- Use the Nokia Developer bug reporting channel.

What devices support Nokia Asha web apps?

A complete list of the devices that can run web apps is found in the Nokia Developer Devices section.

During the 3.0 beta phase the ability to capture images and video will only be available on the Nokia Asha 501. You can target all other features at Nokia Asha phones with Nokia Asha web app tools 3.0.

What Nokia Browser version I should use?

- Phones that have Xpress browser as default browser, automatic update mechanism keeps your browser up to date.
- For Nokia Asha phones that, don't have Xpress Browser pre-installed, visit http://download.browser.ovi.com/ to download latest browser.

Development

What are new features in Nokia Asha Web App Tools version 3.0 Beta

- Simulator support for Nokia Asha software platform 1.0 devices.

(C) Copyright Nokia 2014. All rights reserved.

- New HTML attributes "accept" and "capture" with the supported values "image/*", "video/*" and "camera" respectively.
- New sample applications: "MediaCaptureSample" to support the Camera API and "BackKeySample" to demonstrate "mwl.addNavBackListener" events.

Release Notes

What are new features in Nokia Asha web app version 2.3

- UI Designer in WDE
- Web App USB deployment from a PC running Microsoft Windows
- Automatic reloading of the simulator for locally previewed web apps as code changes are saved.
- Additional project templates
- Support for fixed and scrollable regions in web apps by the use of fixed headers and inline scrollable areas.

Release Notes

What changed in new Nokia Asha Web Apps Runtime Proxy server release?

Please refer to Developer guidelines for new Nokia Asha web apps runtime proxy server release

How can I start using new features in use in my Web App?

If you desire to use new features in your web app, you must update Web App Version feature tag in config.xml. Feature tag defines the minimum runtime version Nokia Browser needs to support to run the Web Application.

```
<feature name="nokia://s40.nokia.com/SAWRT/3.0" required="true" />
```

Note that during the beta release of Nokia Asha web apps Version 3.0, support for media capture is only available on smartphones based on Nokia Asha software platform 1.0.

In addition a new version of Nokia Asha Web App tools is needed to support new features. You can download Nokia Asha Web App Tools here.

How can I make sure that my Web App works in different devices?

Nokia Asha software platform 1.0. based devices

Screen width matches Series 40 full touch devices (240px) ensuring compatibility of the web apps. Special care must be taken to implement support for hardware back key for Nokia Asha software platform 1.0. based devices. This is critical to offer best possible UX to end users. Font spacing is slightly different in Nokia Asha 501, thus avoid having tight fitting text inside fixed size elements.

Series 40 Full Touch Devices

Nokia Asha Web Apps are always presented in portrait mode in Full Touch devices. The screen width of full touch device matches the width commonly used in Series 40 devices (240 px). To make sure that your web app works in a full touch device, verify that UI layout is not hard coded on fixed height of 320 pixels. No other actions are needed.

Architecture

How do Nokia Xpress browser / Nokia Asha web apps differ from conventional browser and web app execution environments

Nokia Xpress Browser is a distributed user agent. The browser client renders content compressed and formatted by the Nokia Xpress Proxy.

For developers this means that JavaScript is executed in the proxy rather than on the client.

This helps provide a rich web app experience in very affordable hardware that would have challenges running a full JavaScript stack.

Please read the Architecture and behavior overview document for more information.

Can I host my application code on my own server?

To some extent, yes. You can place JavaScript resources on your server and load them into your web app.

(C) Copyright Nokia 2014. All rights reserved.

What is the biggest or recommended file size for Nokia Asha web apps?

There is a limit to the size of .wgt files that can be uploaded, it's currently set to 500kb.

Debugging and testing

How can I debug or log my application?

Web Apps Simulator (one of the Nokia Asha Web App Tools]) provides server side JavaScript logging with console.log() calls. Additionally, you can use the Simulator Web Inspector functionality to inspect e.g. the DOM tree or the consumed resources. Full Web Inspector functionality is available in the local preview state of the simulator, while n cloud preview mode, you can inspect the data transfer as it would happen between the proxy and client in real target devices.

Please see the simulator user guide for more information.

Nokia Asha Web Apps doesn't support logging or debugging in target hardware.

However, you can use a 3rd party logging solution such as log4javascript to flush logs to your own server script. Please refer to this example for implementing a logging solution.

Tools

What's the difference between local preview and cloud preview states in Web App Simulator?

Local preview is client side execution of your project's Javascript and HTML through an emulator - much like running your web app in a normal desktop browser. Local preview is useful because it allows you to debug JavaScript, and therefore helps you to spot possible errors more easily (it is also much faster for basic testing and works even if you don't have an Internet connection).

Care must be taken because local preview is the *least accurate* representation of how your web app will actually run on a device. It will allow you to use HTML and JavaScript that is not supported on the real proxy browser. It should not be used to verify the Nokia Asha web app user experience.

Cloud preview (by contrast) runs most of the app JavaScript and HTML in the server-side proxy browser. It reflects the actual supported feature set of HTML/JavaScript on the device and also the user experience that comes from proxy-based app execution.

How to launch a web app from command line

Please note that this is not an officially supported feature and may not be available in future releases. You can launch a Web App into the simulator from the command line using following syntax:

```
C:\Program Files (x86)\Nokia Asha Web App Tools 3.0.0\Web App Simulator>WebSDKSimulator.exe
-f [full path of config.xml] -url [short url as appearing in WDE console].
```

JavaScript

What is MWL (Mobile Web Library)?

MWL is a JavaScript library provided by Nokia for Nokia Asha web apps.

Direct MWL calls are the only JavaScript that will be executed on the client, thus you will find it useful to implement functionality such as view switching or timer triggered operations.

- the MWL is automatically injected to all web apps - you don't need to add it to your code.
- provides functions to modify the CSS properties of DOM elements, set locally executed timers, and break out of the web app context to remote URLs.

Get an MWL overview and full API documentation for Nokia Asha web apps in the MWL API reference.

Can I use jQuery or other popular libraries?

In many cases, yes.

However keep in mind that JavaScript is executed on the server side and remember the limitations with functionality, such as timers.

Can I use any off the shelf JavaScript?

Most off the shelf JavaScript will function, but keep in mind that any non-MWL call will be executed in the proxy and execution

(C) Copyright Nokia 2014. All rights reserved.

loops will cause a client-server roundtrip.

Some important omissions

- **JavaScript timers** (setTimeout(), setInterval()) **are not supported**,since they do not fit into the distributed architecture of Nokia Xpress Browser / Nokia Asha web apps.
- - Use the MWL timer() function to work around this.

How to execute MWL calls locally on the client?

Rule of thumb: Tie MWL calls directly to DOM event handlers

- Calls wrapped in custom JavaScript functions will always step to the server side.

```
<head>
<script>
function showBox(){
   mwl.show('#box');
}
</script>
</head>
<body>

<!-- This click handler will execute locally -->
<a onclick="mwl.show('#box');">Right way</a>

<!-- This click handler will cause a server roundtrip,-->
<!-- since it's calling a non-MWL function-->
<a onclick="showBox();">Wrong way</a>

<div class="mybox" id="box"></div>
</body>
```

You can chain MWL and other JavaScript calls to, for example, modify displayable elements locally before stepping to the server side:

```
<!--Chains locally executed MWL CSS operations and-->
<!--a locally executed timer that triggers a server-side operated call (myajaxcall)-->

<a onclick="mwl.show('#el1');mwl.hide('#el2');mwl.timer('tmr', 10, 1,
'myajaxcall(\'http://api.com/get\')');">Chain action</a>
```

You can **generate DOM elements with MWL event handlers in JavaScript** also - this can be useful when generating new page structures from downloaded content, when you don't want to deal with massive raw HTML string operations.

```
var myEl = document.createElement('img');
myEl.setAttribute('onclick', 'mwl.toggleClass(\'#progress_anim\',
\'progress_visible\');API.run(' + i + ');');
```

Pay extra attention on quotation marks, when attaching MWL event handlers in JavaScript.

```
document.getElementById("btn").innerHTML = "<div onclick=\'mwl.toggle(\"#tg1\");return
false;\'>wrong</div>";
document.getElementById("btn2").innerHTML = "<div onclick=\"mwl.toggle(\'#tg1\');return
false;\">right</div>";
```


(C) Copyright Nokia 2014. All rights reserved.

Why is my code causing round trips to the server when JavaScript is executed?

See the answers above.

- JavaScript wrapped in non-MWL calls will always step to the server for execution.
- One good approach is to build a logic that downloads and generates fairly large DOM structures:
- - The DOM structure should then be split to visible and hidden parts that can be toggled between with direct locally executed MWL calls.
- - The RSS reader template in Nokia Web Tools is a good example of this approach.

Can I use alert() ?

`alert()` is not supported in Nokia Asha Web Apps. Use `console.log()` instead.

Can I use "`javascript: ...`" to invoke JavaScript functions from anchor tags?

No. The construct "`javascript: ...`" is not supported on the proxy server. If used, it may result in a new web page being opened in the Xpress browser and as a result the web app will no longer behave as expected. Where JavaScript function need to be invoked in an anchor tag it should be done using `<a onclick="javascript function">`.

How do I add a content refresh option to my web app?

If you wish to add an ad-hoc refresh capability to your web app, it should be done as part of application logic e.g. `<a onclick="refreshPageContent();">`. The more conventional options of `window.location.reload();` or the syntax `<a href="javascript:refreshPageContent();"` should be avoided as they cause the web app context to be lost, which results in the web app no longer behaving as expected.

APIs and features

How do I use timers in Nokia Asha web apps?

Standard JavaScript timers do now work in distributed environment of web apps. To overcome this issue, when timer features are need, use the MWL timer() and stopTimer() methods. Note that when using the timer() and stopTimer() methods that inline coding should be used: these methods will not trigger from a subroutine attached to the event. Typically this means your code should be similar to:

```
function callFunc() {
// Process event here...
}
</script>

</head> <body onload="mwl.timer
  ('timer1', 5000, 0, 'callFunc()');">
```

For more information, see timer() documentation.

Does Nokia Nokia Asha web apps support HTML5

Platform is primarily based on HTML 4.0, but subset of HTML5 related technologies are supported:

- Media Capture API
- Subset of CSS3 Transitions
- Geolocation API

What CSS3 features are supported?

Animated 2D transitions of width, height, and margin properties are supported with the -webkit -prefix. Use MWL calls such as addClass to run animations on the client side.

More information is available in the Animated transitions documentation.

Rounded corners, gradients, and other CSS3 visual effects are not supported in current version of Nokia Asha web apps.

What URI schemes are supported?

(C) Copyright Nokia 2014. All rights reserved.

http, https, tel and sms.

Can I call, or send sms or email?

Initiating calls with tel URI scheme is supported.

```
<a href="tel:+3581234567p123">Dial</a>

<a href="tel:+3581234567p123">Dial - then pause before sending a new number</a>

<a href="tel:+3581234567w123">Dial - then ask the user if he wants to send a new
number</a>
```

SMS sending is supported from Nokia Asha Web Apps 1.5 onward.

```
<a href="sms:+35801234567?body=hello%20world">SMS</a>
```

For more information see SMS URI scheme in Nokia Asha web apps

Email creation is not supported.

How do I playback or stream video or audio?

Current version of web apps enables you to launch video and audio playback using the native player on the device.

However, the use case flow shows a screen prompting the user to start playback of the media file.

Example:

```
<a href="#" onclick="mwl.loadURL('http://www.my.example.video/video.mp4');">Play
video</a>
```

Find information on the supported formats and codecs from web app compatible device specs at Nokia Developer

Embedding video files is not possible in the first version of web apps.

Functionality on the simulator differs from real devices, therefore it's recommended highly that multimedia functionality is tested on a device.

Can I create custom items for the Nokia Asha web app options menu?

Not in the current version. You need to create your own menu with HTML. From version 2.3 onwards, fixed and scrollable region support enables you to make stationary menus.

Can I tie an event handler to the hardware back key on phones based on Nokia Asha software platform 1.0 ?

Yes, you can and it is highly encouraged to do so. `addNavBackListener()` method enables you to implement standard back stepping support in Nokia Asha web apps.

addNavBackListener() documentation

How can I close my app from JavaScript code?

window.close() and the other options you might normally use aren't supported in the current release.

User can exit the web app using built in exit command (Series 40 platform). In Nokia Asha software platform 1.0 based devices web app can be closed by swiping or by pressing hardware back key in main screen.

How can I detect resolution or other client side parameters?

The wiki article Coding Nokia Asha web apps for screen orientation describes how to detect and react to landscape and portrait

(C) Copyright Nokia 2014. All rights reserved.

screen formats.

Can I discover device features, such as whether my web app is running on a Touch and Type device?

You can discover the characteristics and feature of the device upon which a web app is running by looking at the http request headers sent by your web app and received by a remote web server.

We have set up simple php scripts to get text and JSON output of your user agent string:

```
*http://logme.mobi/js.text
```

- http://logme.mobi/js.json

The server side scripts for the above are fairly simple and can be freely copied if required: Apache server rewrites:

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} /js.*
RewriteRule js.(.*) /index.php?output=$1 [L]
</IfModule>
```

PHP:

```
<?php
// Read a GET parameter to determine the output format, if none specified, nothing is
output
if (isset($_GET['output']) && $_GET['output'] != '') {
  $show_markup = $_GET['output'];
}

// Preparing an Array with the User-Agent string. Might add more parameters in the
future
if (isset($headers['User-Agent'])) {
   $api_values['User-Agent'] = $headers['User-Agent'];
}
switch ( $show_markup ) {
  case 'json':
  header('Content-type: application/json');
  echo json_encode($api_values);
  break;

  case 'xml':
  // not implemented
  break;

  case 'text':
  header('Content-type: text/plain');
  echo $api_values['User-Agent'];
  break;
}
?>
```

This is example JavaScript code to call the above mentioned URLs at logme.mobi and obtaining the real user agent string of the Nokia Xpress Browser client:

(C) Copyright Nokia 2014. All rights reserved.

```
function handleResponse(content){
   document.getElementById("ua").innerHTML = content;
}

function getLogMe(){
   var xhr;
   var myurl = "http://logme.mobi/js.text";
   xhr = new XMLHttpRequest();

   xhr.open("GET", myurl, true);
   xhr.setRequestHeader("Cache-Control", "no-cache");
   xhr.setRequestHeader("Pragma", "no-cache");
   xhr.onreadystatechange = function(){
      if (xhr.readyState == 4) {
         if (xhr.status == 200) {
           if (xhr.responseText != null) {
               // Work with the response
               handleResponse(xhr.responseText);
           }else{
            // something is wrong
            return '';
           }
      }else{
      // something is wrong
      return '';
      }
   }
}
//send the request
xhr.send(null);
}
```

This would give you the real user agent string to play with, after which you can determine what device the browser is running on.

Can I store data locally on the client?

Local storage isn't supported in web apps, however you can use `widget.preferences` to store data in cloud.

You can use the preferences attribute of W3C Widget Interface to store application instance specific persistent data in the Nokia Xpress proxy. For details see: Using widget.preferences in Nokia Asha Web Apps

Do you provide an API to query user's location?

W3C Geolocation API `getCurrentPosition()` is supported in Nokia Asha Web Apps Version 1.5 onwards.

Please note that anonymous functions cannot be passed to getCurrentPosition() as arguments.

```
function getPosition()
{
 if(navigator.geolocation){
    navigator.geolocation.getCurrentPosition(geoSuccess, geoFailure);
    }
  else{

// geolocation API is not available, handle the case here
  }

}
```


(C) Copyright Nokia 2014. All rights reserved.

```
function geoSuccess(position){
/*
* handle success here
* use: position.coords.latitude, position.coords.longitude
*                            position.coords.accuracy and position.timeStamp
*/
}

function geoFailure(error){
/*
* handle failure here
* use: error.code and error.message
*/
}
```

A user agent IP lookup also returns the IP of the data center of Xpress browser proxy through which the network traffic is routed.

Do you support device APIs?

Yes W3C Geolocation API (version 1.5 onwards) and Media Capture API (v3.0 onwards, currently only supported in Nokia Asha 501)

Do you support Flash?

There is no Flash support in Nokia Asha web apps / Nokia Xpress Browser

Can Nokia Asha web apps integrate to device home screen?

Home screen integration isn't supported.

Can I use a map API in Nokia Asha web apps?

Yes. Any map service that outputs static images (png, gif, jpeg) should do.

Map Image API provides an easy to use REST interface for accessing, for example, map tiles.

Can I use an advertising API in Nokia Asha web apps

Yes, advertising solutions can be used, including Inneractive.

I need to integrate to Facebook and Twitter from my web app, how do I do it?

Current Nokia Asha web apps development offering doesn't contain frameworks for integrating to popular social networks. Thus, you need to implement integration points yourself or use third-party frameworks.

Do you have a recommended UI library?

The current version of Nokia Asha web apps doesn't include a full UI library.

- Nokia Developer Codeexamples and Nokia WebTools project templates and code snippets contain some HTML and CSS definitions to help build UI controls and layouts.
- [1] provides UI Graphics stencils

How can I customize the favicon that is displayed for my web app within Xpress Browser web apps menu?

To customize the favicon for your app, simply add the following line within index.html:

<link rel="icon" type="image/png" href="favicon.png">

The favicon image file should be 16x16 PNG format with a transparent background.

Porting

How do I port my existing web site?

(C) Copyright Nokia 2014. All rights reserved.

- Familiarise yourself with this FAQ, the Nokia Asha web app developer's library to understand the limitations in features such asJavaScript execution, CSS support, and event handling.
- Keep in mind that web apps run in affordable mobile phones with 240x320px screens in landscape or portrait resolution or in 240x400px screen in portrait mode.

How do I port my existing Symbian WRT widget or other packaged web app?

Some things to check and remember:

- Nokia Asha web apps W3C widget packaging format, which is different from Symbian WRT.
- As with porting between any other environments, API and feature differences exist.
- JavaScript (except direct MWL calls) execute on the server, thus many apps with heavy client side logic, DOM manipulation, etc. would benefit from redesign ofthe code to optimise the need for server round trips.

In many cases, it's worthwhile considering redesigning your application rather than attempting to port.

Other

How do I find out the version of Nokia Browser running on my device

Select Tools -> About in Nokia Browser Options menu

What is the user agent string for the Nokia Browser for Series 40 / Series 40 web apps clients

Example user agent strings:

- Nokia Browser for Series 40 Version 1.0
  - Mozilla/5.0 (Series40; NokiaX3-02/05.65; Profile/MIDP-2.1 Configuration/CLDC-1.1) Gecko/20100401 S40**NokiaBrowser**/1.0.0.9.17.
- Nokia Browser for Series 40 Version 1.5
  - Mozilla/5.0 (Series40; NokiaX3-02/06.00; Profile/MIDP-2.1 Configuration/CLDC-1.1) Gecko/20100401 **S40OviBrowser**/1.5.0.34.11

However, as the browser was previously known as Ovi Browser some implementation will report ‘OviBrowser’, as in the following example user agent string: Mozilla/5.0 (Series40; NokiaX3-02/05.65; Profile/MIDP-2.1 Configuration/CLDC-1.1) Gecko/20100401 S40**OviBrowser**/1.0.0.9.17

- Xpress Browser for Series 40 Version 2.3
  - Mozilla/5.0 (Series40; Nokia311/5.92; Profile/MIDP-2.1 Configuration/CLDC-1.1) Gecko/20100401 S40OviBrowser/2.3.0.0.49
- Xpress browser for Nokia Asha platform Version 3.0
  - Mozilla/5.0 (Series40; Nokia501/1.0; Profile/MIDP-2.1 Configuration/CLDC-1.1) Gecko/20100401 S40OviBrowser/3.0.0.0.67

How do I find out the firmware version of my device?

Dial *#0000# in the home screen of your device.

Publishing

Where can I publish my app?

You can publish your web apps through Nokia Store. For help see: Publishing your web app

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.