

Mobile Page

A mobile page is a screen of a browser. It is composed of Header/Footer bar and Content UX components. Links are used to transit pages.

Notes

Browser has a different history management method (pushstate). It is recommended that history.js be used to support all the browsers.

Importing the history.js makes the back and forward buttons of the browser to work normally.

history.js download

Declare it above jquery.mobile-1.0b3.min.js

```
<!-- History.js -->
```

```
<script type="text/javascript" src="/js/egovframework/mbl/cmm/jquery.history.js"></script>
```

pushstate enabled browsers : Chrome(5.0), Firefox(4.0), Opera(11.50), safari(5.0)

- Older browsers should import import istory.js to each page.

※ **When the browser can support pushstate, you can remove the script.**

Base Code

The mobile pages are based on HTML5 and CSS3. The document should start with HTML5 **<!DOCTYPE html>**. Declare the following under <head>.

```
<!DOCTYPE html>
<html>
  <head>
    <title>eGovFrame</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
    <link rel="stylesheet" href="/css/egovframework/mbl/cmm/jquery.mobile-1.0b3.min.css"/>
    <link rel="stylesheet" href="/css/egovframework/mbl/cmm/EgovMobile.css" />
    <script src="/js/egovframework/mbl/cmm/jquery-1.6.2.min.js"></script>
    <script src="/js/egovframework/mbl/cmm/jquery.mobile-1.0b3.min.js"></script>
    <script src="/js/egovframework/mbl/cmm/EgovMobile.js"></script>
  </head>
  <body>
    ...
  </body>
</html>
```

Components

Use HTML5 **data-*** property to use [header](#), content and [footer](#) in <body> to organize the mobile page components.

```
<body>
  <!-- Mobile pagestart -->
  <div data-role="page">
    <!-- header start -->
    <div data-role="header">
      <h1>header Title</h1>
    </div>
    <!-- header end -->

    <!-- content start -->
    <div data-role="content">
      <p> This is the default page screen of the e-government mobile standard framework.</p>
    </div>
    <!-- content end -->

    <!-- footer start -->
    <div data-role="footer">
      <h4>footer title</h4>
    </div>
    <!-- footer end -->

  </div>
  <!-- Mobile pageend -->
</body>
```

Page Title

When Ajax is used to transit pages, only the content of data-role="page" is called. Therefore you cannot modify <title> provided by default. By designating **data-title="[title]"** in <data-role="page"> tag, you can designate the title to the activated page

```
<div data-role="page" id="foo" data-title="Page Foo">
<div
```

Page Loading

The mobile e-government standard framework recommends the **domain internal link** rather than the page internal link.

Prefetch Page

When the domain internal link is used, screen transition and data loading is slower than the internal link. Therefore you should declare **data-prefetch**. If **data-prefetch** is declared in the domain internal link, the pagecreate event takes place to read the page where data-prefetch is declared. And the target page is loaded in the background. Quick page transition is allowed as the page is loaded in advance before connection.

```
<a href="/guide/components/pageStructure/page.do" data-prefetch> ... </a>
```

Caching Page

In the mobile e-government page, when a page is transited, the previous page DOM is deleted by default. If you want keep it, you can store it in domCache. But it is recommended that you store it only when necessary, as an error could occur in page transition and memory if the DOM is large.

< When declared in mobileinit >

```
$.mobile.page.prototype.options.domCache = true;
```

<When declared in the page >

```
<div data-role="page" id="cacheMe" data-dom-cache="true" >
```

Ajax-based Hash URL Control

If Ajax is used, you can control the appearance of hash url. If you need hash url, declare **\$.mobile.pushStateEnabled=true**. If you don't, declare **\$.mobile.pushStateEnabled=false**.

*- default 는 \$.mobile.pushStateEnabled=true

< Declare in mobileinit>

```
$.mobile.pushStateEnabled=true;
```

Page Navigation

Mobile page navigation uses Ajax by default. This is to give an optimized screen transition effect to the mobile device. You can change this by setting the options.

Navigation within Page

You can use this method when multiple pages are declared in a HTML. This is a basic method of composing a mobile page. Ajax is used. Apply the id value of the div tag declared as a page to the link(#pageId).

* If multiple pages are declared in a HTML, the top page is recognized as the first screen.

```
<!-- main page -->
```

```
<div id="main" data-role="page">
```

```
  <a href="#view" data-role="button">Writing</a>
```

```
<p>Main Page.</p></div>
```

```
<!-- view page -->
```

```
<div id="view" data-role="page">
  <p>View detail page</p>
</div>
```

Navigation outside page

To navigate outside the page, Ajax is used. HTML is called by Ajax and its data-role=="page" area is read and added to the DOM element of the HTML page (DOM is managed in the similar way of navigation within page).

* Since the page area of HTML called by Ajax is read, the Javascript and CSS used by the called page should exist in the HTML page that called.

* Not to use Ajax, you can make reference to [Link](#) for modification.

main.html

```
<div id="main" data-role="page">
  <a href="create.html" data-role="button">Writing</a>
<p>Main Page.</p></div>
```

...

create.html

```
<div id="create" data-role="page">
  <p>Writing page</p>
</div>
```

...