OCCI User Interface (XHTML5)

1. Overview

1. XHTML5 is an XML-based “concrete syntax” of the HTML5 “abstract language” (the “World Wide Web's markup language”). For the purposes of this rendering specification we mandate XHTML5 over HTML5 as it is valid to present HTML5 that is not well-formed. It is served using the `application/xhtml+xml` Internet media type.HTML5

2. XHTML5

2. XHTML5 is used to describe a user-friendly rendering of the object, optionally with images, icons, javascript, etc. embedded. It is also marked up with RDFa attributes so as it is also machine readable. The use of RDFa to demarcate semantic content is used throughout the XHTML5 renderings where ever appropriate. This combination of RDFa and XHTML5 is a W3C Recommendation.

3. Pages are typically either lists of resources (e.g. `<ul>`, `<ol>`, `<tr>` etc.) or a rendering of an individual resource. To declare an OCCI attribute (e.g. `compute.memory`), the RDFa `property` attribute is used. Links in the `<head>` are specified using the HTML5 `<link>` element. Links in the `<body>` are specified using the HTML5 `<a>` element. Properties and links that are defined to be read-only will appear in the head or body of the appropriate document. To signal to clients that a property is read-write (i.e. modifiable), the set of modifiable properties must be encapsulated in a form *<!--| there is some work remaining to be specified here as all specifications are somewhat deficient - discuss at confcall |-->*.

2.1. Examples

4. What follows is an example representation of an OCCI compute resource rendered as XHTML5. The resource contains read-only and read-write attributes as well as link attributes of the resource.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>OCCI: My Virtual Machine</title>
    <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
    <!-- Here links to related resources are inserted -->

  </head>

  <body xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:occi="http://purl.org/occi/">

    <!--
    Here we add all properties related to the compute resource
    Properties that are read/write are enclosed by the <FORM> element
    Properties that are read-only are enclosed by any valid HTML5
      element EXCEPT <FORM>
    Whether a property is read/write or read-only is an implementation
      detail settled by the provider
    -->


    <!-- Here a set of read-only properties are exposed using RDFa
         vocabularly to signal semantics -->
    <h1 property="occi:title">My Virtual Machine</h1>
    <span property="occi:summary">A simple sample virtual machine</span>
```

```
        Console:
        <a href="/myvm/console.png" rel="http://purl.org/occi/compute#console"
           type="image/png"><img src="/myvm/console-thumb.png" /></a>
        <a href="ssh://myvm:22"
           rel="http://purl.org/occi/compute#console">SSH</a>

        <!-- Here a set of read/write properties are exposed via a HTML5 form. -->
        <!-- The HTML5 FORM supports all the basic HTTP verbs required by REST -->
        Attributes:
        <form method="POST" action="/myvm">
          <ul>
            <li>ID: <input type="text" name="id"
                value="urn:uuid:164a3064-1176-4de3-b24e-e16dc8c2d4ed" />
            </li>
            <li>Cores:
                <select name="compute.cores">
                  <option>1</option><option selected>2</option><option>3</option>
                </select>
            </li>
            <li>Memory: <span property="occi:compute.memory">2048</span>
            </li>
          </ul>
          <input type="submit" value="Save" />
        </form>

        <!-- OCCI actions are equivocal and map to HTML5 forms -->
        Actions:
        <form class="action" method="POST" action="/myvm;start">
          <input type="image" src="occi-start.png" alt="Start" />
        </form>
        <form class="action" method="POST" action="/myvm;stop">
          <input type="image" src="occi-stop.png" alt="Stop" />
        </form>
        <form class="action" method="POST" action="/myvm;restart">
          <input type="image" src="occi-restart.png" alt="Restart" />
        </form>
        <form class="action" method="POST" action="/myvm;resize">
          <select name="compute.cores">
            <option>1</option><option>2</option><option>4</option>
          </select>
          <input type="text" name="compute.memory" value="2048" />
          <input type="image" src="occi-resize.png" alt="Resize" >
        </form>
        <form class="action" method="DELETE" action="/myvm">
          <input type="image" src="occi-delete.png" alt="Delete" />
        </form>

        Download
        <a href="/myvm.ovf" rel="alternate" type="application/ovf+xml">
          <img src="ovf-logo.png" alt="OVF" />
        </a>
        <a href="/myvm.xen" rel="alternate" type="application/xen+xml">
          <img src="xen-logo.png" alt="Xen" />
        </a>

      </body>
    </html>
```

2.1.1. Collections

5. What follows is an example representation of an OCCI compute resource collection rendered as XHTML5. The resources of the collection are displayed with read-only attributes.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <title>OCCI: My Cloud</title>
  <meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
</head>

<body xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:occi="http://purl.org/occi/">
<h1>My Cloud</h1>

Compute Resources:
<table>
  <th>
    <td>Console</td>
    <td>ID</td>
    <td>Cores</td>
    <td>Memory</td>
    <td>Actions</td>
    <td>Formats</td>
  </th>
  <tr typeof="occi:compute" about="/myvm">
    <td>
      <a href="/myvm/console.png"
         rel="http://purl.org/occi/compute#console" type="image/png">
        <img src="/myvm/console-thumb.png" />
      </a>
      <a href="ssh://myvm:22"
         rel="http://purl.org/occi/compute#console">SSH</a>
    </td>
    <td property="occi:title">
      urn:uuid:164a3064-1176-4de3-b24e-e16dc8c2d4ed
    </td>
    <td>
      <span property="occi:title">
        My Virtual Machine
      </span><br/>
      <span property="occi:summary">
        A simple sample virtual machine
      </span><br/>
    </td>
    <td property="occi:compute.cores">2</td>
    <td property="occi:compute.memory">2048</td>
    <td>
      <form class="action" method="POST" action="/myvm;start">
        <input type="image" src="occi-start.png" alt="Start" />
      </form>
      <form class="action" method="POST" action="/myvm;stop">
        <input type="image" src="occi-stop.png" alt="Stop" />
      </form>
      <form class="action" method="POST" action="/myvm;restart">
```

```
            <input type="image" src="occi-restart.png" alt="Restart" />
          </form>
        </td>
        <td>
          <a href="/myvm.ovf" rel="alternate" type="application/ovf+xml">
            <img src="ovf-logo.png" alt="OVF" />
          </a>
          <a href="/myvm.xen" rel="alternate" type="application/xen+xml">
            <img src="xen-logo.png" alt="Xen" />
          </a>
        </td>
      </tr>
    </table>

    </body>
    </html>
```

3. Javascript

6. Asynchronous Javascript may be used to deliver interactivity without having to reload the page.

3.1. Example

7. In this example, a HTTP request for a resource is asked for and using the OCCI JavaScript client, demonstrates the ability of reading HTML5 headers.

```
var OCCIclient = {};

OCCIclient.URI = document.location.href;
OCCIclient.XHR = null;

if (window.XMLHttpRequest) {
  // code for Firefox, Mozilla, IE7, etc.
  OCCIclient.XHR = new XMLHttpRequest();
} else if (window.ActiveXObject) {
  // code for IE6, IE5
  OCCIclient.XHR = new ActiveXObject("Microsoft.XMLHTTP");
}

if (OCCIclient.XHR!=null) {
  OCCIclient.XHR.onreadystatechange = function() {
    if(OCCIclient.XHR.readyState == 4) {
      x = OCCIclient.XHR.getAllResponseHeaders();
      document.getElementById("header").innerHTML = x;
    }
  }
  OCCIclient.XHR.open("HEAD",OCCIclient.URI, true);
  OCCIclient.XHR.send();
} else {
  alert("Your browser does not support XMLHTTP.");
}
```

Bibliography

Normative References

[HTML5] *HTML 5*. http://www.w3.org/TR/html5/ [http://tools.ietf.org/html/rfc4287]. Ian Hickson. David Hyatt. World Wide Web Consortium (W3C) 2009-08-25.

[RDFa] *RDFa*. http://www.w3.org/TR/xhtml-rdfa-primer/ [http://tools.ietf.org/html/rfc4287]. Ben Adida, Creative Commons. World Wide Web Consortium (W3C) 2008-10-14.

Informative References

[http-article] *Is HTTP the HTTP of cloud computing?*. http://samj.net/2009/05/is-http-http-of-cloud-computing.html [http://samj.net/2009/05/is-atompub-already-http-of-cloud.html]. Sam Johnston. Australian Online Solutions 2009-05-25.

[HTML5-article] *Misunderstanding Markup: XHTML 2/HTML 5 Comic Strip*. http://www.smashingmagazine.com/2009/07/29/misunderstanding-markup-xhtml-2-comic-strip/. Jeremy Keith. Brad Colbow. 2009-07-29.