



Here we describe the Application Programming Interfaces (API) of Mobile Web Server.

Contents

- 1 REST APIs
 - ◆ 1.1 API XML schemas
 - ◆ 1.2 Presence REST API (since MWS 1.3)
 - ◆ 1.3 Messaging REST API (since MWS 1.3)
 - ◆ 1.4 Availability REST API (since MWS 1.5)
 - ◆ 1.5 Add Meeting REST API (since MWS 1.5, changed in MWS 1.5.1)
- 2 RSS feeds (since MWS 1.2, content feed added in MWS 1.5)
 - ◆ 2.1 Feeds hosted mymobilesite.net gateway
 - ◆ 2.2 Feeds hosted on mobile sites
- 3 Badge
- 4 DIY
- 5 Comments

REST APIs

We have implemented a REST API framework and following APIs. REST APIs follow the mobile site authentication and authorization scheme as such.

NOTES Allowing anonymous access to one of the presence resources causes the authentication for the other two resources to stop working in some configurations. The authentication should still always work when connection filtering is on or if the user first logs in through the MWS web pages.

API XML schemas

```
http://mymobilesite.net/schema/presence/1.0.0/presence.xsd
http://mymobilesite.net/schema/presence/1.0.0/availability.xsd
http://mymobilesite.net/schema/presence/1.0.0/devicestate.xsd
http://mymobilesite.net/schema/presence/1.1.0/location.xsd
http://mymobilesite.net/schema/contact_me/1.0.0/sms_message.xsd
http://mymobilesite.net/schema/calendar/1.2.0/availability.xsd
http://mymobilesite.net/schema/calendar/1.2.0/addmeeting.xsd
http://mymobilesite.net/schema/calendar/1.2.1/meeting.xsd
```

Presence REST API (since MWS 1.3)

The Presence web application provides a REST API for accessing terminal presence information in XML format. This REST API provides the following four resources (where domain_name is for example ?somedomain.mymobilesite.net?):

```
http://<domain_name>/rest/presence
http://<domain_name>/rest/presence/availability
http://<domain_name>/rest/presence/devicestate
http://<domain_name>/rest/presence/location
```

Mobile_Web_Server_APIs

- Presence resource

The presence resource contains links to the other two resources which contain the actual presence availability and device state information. If user has access to only one of these resources then only that resource is listed in the XML response. Requesting the presence resource will return HTTP_UNAUTHORIZED If user doesn't have access to either resource. This resource allows only the HTTP GET method. XML schema:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="presence">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="availability">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:anyURI">
                <xs:attribute name="xlink" type="xs:anyURI" />
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="devicestate">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:anyURI">
                <xs:attribute name="xlink" type="xs:anyURI" />
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="location">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:anyURI">
                <xs:attribute name="xlink" type="xs:anyURI" />
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

- Availability resource

The availability resource contains the same information that is listed in the main view under the availability information title. Requesting the availability resource will return HTTP_UNAUTHORIZED If user doesn't have access to availability information. This resource allows only the HTTP GET method. The details of availability information are:

- Profile is a string with a set of possible values defined as an enumeration.
- Calendar_availability and Call state are Boolean values.
- Reserved_for_secs is an integer value. If currently available this value is 0.

XML schema:

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

Mobile_Web_Server_APIs

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="availability">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="profile">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="general" />
              <xs:enumeration value="silent" />
              <xs:enumeration value="meeting" />
              <xs:enumeration value="outdoor" />
              <xs:enumeration value="pager" />
              <xs:enumeration value="offline" />
              <xs:enumeration value="drive" />
              <xs:enumeration value="custom" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="calendar_availability" type="xs:boolean" />
        <xs:element name="reserved_for_secs">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="0" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="call_state" type="xs:boolean" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

- Device state resource

The Device state resource contains the same information that is listed in the main view under the device state title. Requesting the device state resource will return HTTP_UNAUTHORIZED If user doesn't have access to device state information. This resource allows only the HTTP GET method. The details of device state information are:

- Battery level is an integer value between 0 and 100.
- Localtime is presented as XML dateTime field.
- Connection_type is a string with a set of possible values defined as an enumeration.

XML schema:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="devicestate">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="battery_level">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="0" />
              <xs:maxInclusive value="100" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="localtime" type="xs:dateTime" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Mobile_Web_Server_APIs

```
<xs:element name="connection_type">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Not connected" />
      <xs:enumeration value="Unknown" />
      <xs:enumeration value="CSD" />
      <xs:enumeration value="3G" />
      <xs:enumeration value="LAN" />
      <xs:enumeration value="CDMA2000" />
      <xs:enumeration value="GPRS" />
      <xs:enumeration value="HSCSD" />
      <xs:enumeration value="EGPRS" />
      <xs:enumeration value="WLAN" />
      <xs:enumeration value="Bluetooth" />
      <xs:enumeration value="Virtual" />
      <xs:enumeration value="Virtual VPN" />
      <xs:enumeration value="CSD (WCDMA)" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

- Location resource (updated in MWS 1.4)

The Location resource contains the same coordinate information that is listed in the Presence main view under the location title. In case of unauthorized access to location information, HTTP_UNAUTHORIZED HTTP status code is returned to the request origin. HTTP GET is the only allowed method to access the device state information via REST API. The details of device state information are:

- Latitude is a decimal value between -90 and 90.
- Longitude is a decimal value between -180 and 180.
- Accuracy is a positive integer value.
- Method is a string with a set of possible values defined as an enumeration.

XML schema:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="location">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="latitude">
          <xs:simpleType>
            <xs:restriction base="xs:decimal">
              <xs:minInclusive value="-90" />
              <xs:maxInclusive value="90" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="longitude">
          <xs:simpleType>
            <xs:restriction base="xs:decimal">
              <xs:minInclusive value="-180" />
              <xs:maxInclusive value="180" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Mobile_Web_Server_APIs

```
</xs:element>
<xs:element name="accuracy">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="method">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Bluetooth GPS" />
      <xs:enumeration value="Assisted GPS" />
      <xs:enumeration value="Integrated GPS" />
      <xs:enumeration value="Network-based" />
      <xs:enumeration value="None" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Messaging REST API (since MWS 1.3)

This application provides a REST API for 3rd party applications to send messages to the MWS admin's mobile terminal inbox using a HTTP POST message.

Requesting this REST resource will return HTTP_UNAUTHORIZED If user doesn't have access to send SMS messages. This resource allows only the POST method.

The REST resource is located at the url

```
?http://<domain_name>/rest/inbox/message?
```

(where domain_name is for example ?somedomain.mymobilesite.net?).

The body of the HTTP POST should contain an XML representation of the sms message to be sent. If the XML representation is missing or corrupt HTTP_NOT_FOUND (404) is returned.

XML Schema for the message:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="message">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:minLength value="0" />
        <xs:maxLength value="500" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```

Availability REST API (since MWS 1.5)

Availability REST API provides three different views into target S60 device's calendar availability information. The views are available in day, week and month formats starting from the UTC time specified in the request URI. The response body of a valid request lists the reservations made in device calendar, within given time frame, in XML format. Only time-consuming calendar entries (entries with duration specified) are considered as reservations and listed in the response body. The following chapters provide detailed description about each of the views. XML schema describing the structure of the documents returned by availability REST API is listed in the following:

```
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="entry">
    <xs:all minOccurs="1" maxOccurs="1">
      <xs:element minOccurs="1" maxOccurs="1" name="start_date" type="xs:dateTime" />
      <xs:element minOccurs="1" maxOccurs="1" name="end_date" type="xs:dateTime" />
    </xs:all>
  </xs:complexType>
  <xs:complexType name="reservations">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="reservation" type="entry" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

In the examples embedded in the subchapters below domain_name is e.g. alice.mymobilesite.net. Definition for unix_time_stamp is available at http://en.wikipedia.org/wiki/Unix_time.

- Day view

Day view presents the device calendar availability information within the next day (24hrs) since the timestamp specified in the request URL. Since the request contains no body, the only allowed HTTP method for requesting the availability information is GET. For successful API calls 200 OK HTTP status code and reason, accompanied with the XML formatted body, are returned to the request origin. In case of unauthorized or otherwise malformed request an HTTP status code and reason, best describing the reason for denial of service, are returned instead. In order to successfully complete the request, the request URI must conform to the following format:

```
http://domain_name/rest/calendar/availability/day/unix_time_stamp
```

- Week view

Week view presents the device calendar availability information within the next week since the timestamp specified in the request URL. Since the request contains no body, the only allowed HTTP method for requesting the availability information is GET. For successful API calls 200 OK HTTP status code and reason, accompanied with the XML formatted body, are returned to the request origin. In case of unauthorized or otherwise malformed request an HTTP status code and reason, best describing the reason for denial of service, are returned instead. In order to successfully complete the request, the request URI must conform to the following format:

```
http://domain_name/rest/calendar/availability/week/unix_time_stamp
```

- Month view

Mobile_Web_Server_APIs

Month view presents the device calendar availability information within the next month since the timestamp specified in the request URL. Since the request contains no body, the only allowed HTTP method for requesting the availability information is GET. For successful API calls 200 OK HTTP status code and reason, accompanied with the XML formatted body, are returned to the request origin. In case of unauthorized or otherwise malformed request an HTTP status code and reason, best describing the reason for denial of service, are returned instead. In order to successfully complete the request, the request URI must conform to the following format:

```
http://domain_name/rest/calendar/availability/month/unix_time_stamp
```

Add Meeting REST API (since MWS 1.5, changed in MWS 1.5.1)

Add meeting REST API makes it possible to propose new meeting type calendar entries in target S60 device's calendar application. If the user has the access right to create calendar meetings the new meeting is instantly created but otherwise a meeting proposal is displayed on the device UI and admin can accept or reject the request. The details of a new meeting instance are defined by the XML body of the request. Since request body is required, the only approved method for making a call to the API is HTTP POST. In case of a successful API call 200 Ok HTTP status code and XML describing the outcome of the proposal are returned to the request origin. Otherwise an HTTP status code and reason, best describing the reason for denial of service, are returned instead. In order to successfully complete a request to this API, the request URI must conform to the following format (where domain_name is e.g. alice.mymobilesite.net):

```
http://domain_name/rest/calendar/meeting
```

The schema describing the structure of the XML documents embedded in request bodies and approved by the API is listed in the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" >
  <xs:complexType name="repeat">
    <xs:all minOccurs="1" maxOccurs="1">
      <xs:element minOccurs="1" maxOccurs="1" name="repeat_type">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="daily"/>
            <xs:enumeration value="weekly"/>
            <xs:enumeration value="fortnightly"/>
            <xs:enumeration value="monthly"/>
            <xs:enumeration value="yearly"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element minOccurs="1" maxOccurs="1" name="repeat_end_date" type="xs:date"/>
    </xs:all>
  </xs:complexType>
  <xs:complexType name="entry">
    <xs:all minOccurs="1" maxOccurs="1">
      <xs:element minOccurs="0" maxOccurs="1" name="subject" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="location" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="from" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="start_datetime" type="xs:dateTime"/>
      <xs:element name="end_datetime" type="xs:dateTime"/>
      <xs:element minOccurs="0" maxOccurs="1" name="alarm" type="xs:dateTime"/>
      <xs:element minOccurs="0" maxOccurs="1" name="repeat" type="repeat"/>
      <xs:element minOccurs="0" maxOccurs="1" default="private" name="synchronization">
```

Mobile_Web_Server_APis

```
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="private"/>
    <xs:enumeration value="public"/>
    <xs:enumeration value="none"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:all>
</xs:complexType>
<xs:complexType name="meetings">
  <xs:sequence minOccurs="1" maxOccurs="unbounded">
    <xs:element name="meeting" type="entry"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

The schema describing the structure of the XML document returned on successful requests is listed in the following:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element minOccurs="1" maxOccurs="1" name="response">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="accepted" />
        <xs:enumeration value="declined" />
        <xs:enumeration value="timeout" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```

RSS feeds (since MWS 1.2, content feed added in MWS 1.5)

Feeds hosted mymobilesite.net gateway

Status feed

Mobile domain statuses can be followed via RSS from our gateway (like this [\[1\]](#) where id is the domain name). This is how our [widgets](#) work.

Content feeds

This feed is hosted on the gateway and generated based on the settings made by admin of the domain. Example feed URL is [\[2\]](#)

Feeds hosted on mobile sites

Blog entries and gallery files can be accessed via RSS as from the device (like this for Blog: [\[3\]](#))

Badge

The mobile site badge can be fetched from the gateway with a code like this:

```
<a target="_blank" href="http://demo.mymobilesite.net"><img border="0" name="thumbimg" id="thumb
```

Where *demo* is the actual domain name.

DIY

If there is no suitable API available, one can just implement the needed functionality (in Python). That is how the desktop widgets were developed, and there is an article available about the experience in Desktop widgets.

Comments

Please describe here your ideas for new APIs or changes/fixes to existing.