# PA International Web presence

PA International produces Geo-location software and data logging units (*devices*). The department for data logging and Geo-location is fully developed, but the department for data presentation and unit monitoring is falling behind.  
  
The PA International has requested a project with a new website and a web-service.

The web-service should be used by the devices and the website, as well as anyone who wishes to write a custom client. Custom clients are not a part of the project.

The system should be able to handle a large number of simultaneous devices (several thousand), as well as a large number of simultaneous web page and service users (several hundred).



## Devices

The devices register themselves and upload status and readings to the web-service.

The devices receive commands and configuration from the web-service.

## Web-service

The web-service must ensure that only authorized data access is permitted. The performance penalty of using HTTPS (SSL/TSL) as opposed to basic HTTP authentication should be considered, yet the actual implementation may be done without authentication or confidentiality concerns.

The web-service has four distinct types of requests:

1. Device registration requests
2. Device reading or status requests.
3. Device configuration and command requests
4. Service administration requests



### Type 1 (Device readings and status)

The web service should maintain three pieces of information about a device:

1. The location and a track of locations,
2. The status and a track of status.
3. A track of the readings.

This information must be exposed to interested parties (web site, custom clients, etc.)

The device information is exchanged in the XML language PAGeoLoggingML, which informally can be described as:

* The root element is *geoLogCollection*, and contains attribute *deviceID*, attribute *timestamp*, and zero or more geoLog elements
* *deviceID* uniquely identify the device and is of type URI.
* *timestamp* is of type DateTime and indicate the time of the reading or status update.
* The *geoLog* element may contain elements *location*, *status*, *readings* and *zone*
* *location* is defined in the KML XML Language (Keyhole Markup Language) and indicates the device location at time *timestamp*.
* *status* contains assorted status information for the device. Exact content will be defined later. Only present if this is a status update.
* *readings* contains a list of key-value-type pairs where the type is either a known MIME-type or defined in the Request header. Only present if this is a new reading.
* *zone* contains information about the zone that this device belongs to, if any. This information is generated by the web-service, and is not part of the device <-> web-service exchange.

An XML Schema formalization of this language must be created as part of the web-service API.

### Type 2 (Device commands and configuration)

Will be defined as needed.

### Type 3 (Service configuration)

The web-service should furthermore allow for the administration of devices as follow:

1. Which user/owner a device is associated with
2. Which zone a device is in, if any (only used for manual override).
3. Definition of zones.

Write actual XML language setup.

### API

The API for the web-service should be REST-like. E.g.

* A GET-request to /geolog/devices returns a list of all devices registered to the user
* A POST-request to /geolog/devices with a device ID and possible a timespan returns a collection of device data for that device.
* A PUT request to /geolog/devices registers a new device on the system
* A PUT request to /geolog/devices/readings adds a reading for a given device
* A detailed specification will be part of the web project

## Web site

The website should expose the information and capabilities of the web service in an easy to understand manner, using graphical aids where possible.

The use of Google Maps to illustrate the location of the devices, as well as the zones is a requirement.

The website must have an application look-and-feel and respond to changes in a device without user interaction.

Technologies  
Java Servlets, JSP, XML, XML Schema, JDOM, XSLT, JavaScript, HTML, CSS, REST, JQuery, Java