

Aegis Web APIs

Developer's Guide

Document Number	EN-TN-0114
Revision Number	1.4
Document Issue Date	2013-08-19

4 Using the Web API

4.1 Common to All Aegis Web APIs

4.1.1 Requests

Use REST calls over HTTPS for all requests, using this general structure:

```
<METHOD> https://<Aegis domain>/G/j/csvc/<api>/<partner id>/<org id>/<application id>/<msisdn>?ts=<timestamp>&tk=<signature>
```

Bold: Required for all calls

Plain: Required for certain calls

<Italic>: Variable

Table 2: Elements of REST Calls to the Aegis Web API

Request Element	Example	Description
<METHOD>	GET	HTTP method.
https://<Aegis domain>	https://www.aegismobilitylabs.com	HTTPS to an Aegis domain.
/G/j/csvc	/G/j/csvc	Constant.
/<api>	/context	API name.
/<partner id>	/795d21164a73413b9dfd7c8ee94ad5e6	Partner ID.
/<organization id>	/ups	Organization ID.
/<application id>	/guardian	Application ID. Not present in all requests.
/<msisdn>	/16045556666	MSISDN (phone number). Not present in all requests.
?ts=<timestamp>	?ts=1334868028	Unix timestamp when the message was sent. Not present in all requests.
&tk=<signature>	&tk=03bccfe988c6965147b8e798ecb27b5c	Cryptographic signature. Not present in all requests.

4.1.2 Responses

The response to all requests is a JSON-formatted array with this structure:

```
{
  "status": "<status code>",
  "msg": "<status message>",
  "data": "<data object>"
}
```

Table 3: Elements Present In All Responses to Your Requests

Response Element	Description
status	An integer value representing a response code. A value of 200 means the request was OK.
msg	Textual details about whether the request succeeded or failed.
data	The response object. Its structure varies depending on the request type.

4.1.3 Example Responses

```
{  
  "status":200,  
  "msg":"OK",  
  "data":{  
    "domain":"www.aegismobilitylabs.com"  
  }  
}
```

```
{  
  "status":603,  
  "msg":"Request must come through https",  
  "data": false  
}
```

4.1.4 Error Codes

Table 4: Error Code Meanings

Error Code	Meaning
400	General error in the request parameters, such as an invalid value. The msg field provides additional detail.
500	Internal error during processing, such as a database problem.
401	Not authorized. The request was made against a URL that required valid credentials.
602	Invalid or unsupported organization domain name.
603	The request was expected to come over HTTPS.
606	Invalid HMAC signature.
610	Invalid Application ID.
611	Invalid or missing partner ID.

4.2 Looking Up the Aegis Domain

Aegis servers are partitioned to serve separate organizations. You need to know which Aegis domain handles the organization you are working with. Once you know the Aegis domain, you don't need to look it up every time.

To look up the domain, send an HTTP GET request to Aegis's `domain_lookup` service. The request URI (see 4.2.1) must include your partner ID and the organization ID (see Table 1).

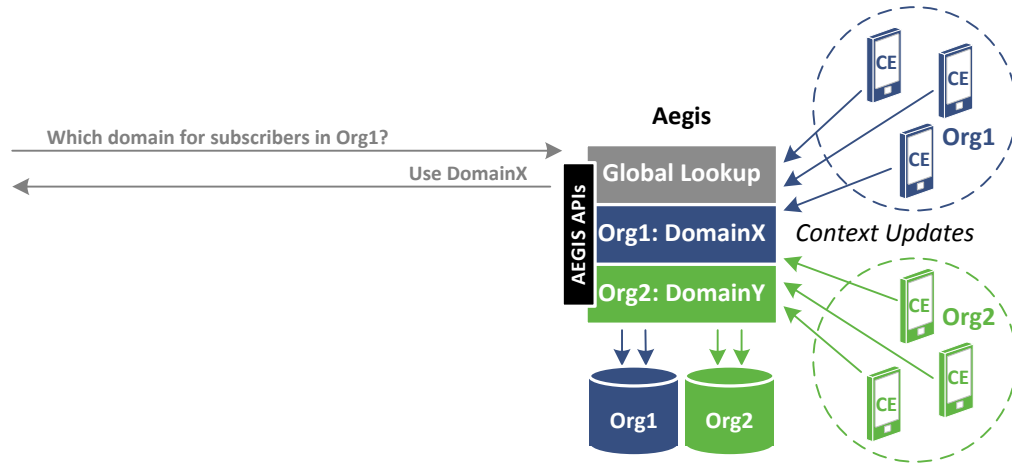


Figure 2. Retrieving a Subscriber's Latest Context

The service returns a JSON-formatted string (see 4.2.2) that includes the Aegis domain to use for the organization.

4.2.1 domain_lookup – Request

```
GET https://www.aegismobilitylabs.com/G/j/csvc/ref/<partner id>/<org id>
```

To request a domain lookup, you need the information in Table 5.

Table 5: Required Elements of a domain_lookup Request

Request Element	Description
partner id	Your partner identifier, assigned to you by Aegis (see Table 1).
organization id	The identifier of the organization you are looking up. Organization identifiers are provided to you by Aegis (see Table 1).

4.2.2 domain_lookup - Response

```
{  
  "status":200,  
  "msg":"OK",  
  "data":{  
    "domain": "<Aegis domain>"  
  }  
}
```

The `domain_lookup` service returns a JSON string that includes the information in Table 6.

Table 6: Elements of a domain_lookup Response

Response Element	Description
status	Status code.
msg	Status message.
data: domain	The Aegis domain to use for further API requests regarding the subscriber or organization.

4.2.3 domain_lookup - Example

4.2.3.1 REQUEST

```
GET https://www.aegismobilitylabs.com/G/j/csvc/ref/  
795d21164a73413b9dfd7c8ee94ad5e6/myorg
```

4.2.3.2 RESPONSE FROM AEGIS

```
{  
  "status":200,  
  "msg":"OK",  
  "data":{  
    "domain":"api0.aegismobilitylabs.com"  
  }  
}
```

The domain_lookup service reports that the lookup succeeded and to send all context requests regarding the subscriber or organization to api0.aegismobilitylabs.com.

4.3 Retrieving a Subscriber's Latest Context

Use the `get_application_state` service to get the latest context state of one subscriber's mobile device. To do so, send an HTTP GET request to the Aegis domain returned in the lookup.

The request URI (see 4.3.1) must include your partner ID, the subscriber's organization ID, and the CE application ID of interest, plus the subscriber's MSISDN (phone number). The URI must also include a timestamp and an HMAC signature (see 4.3.1.1).

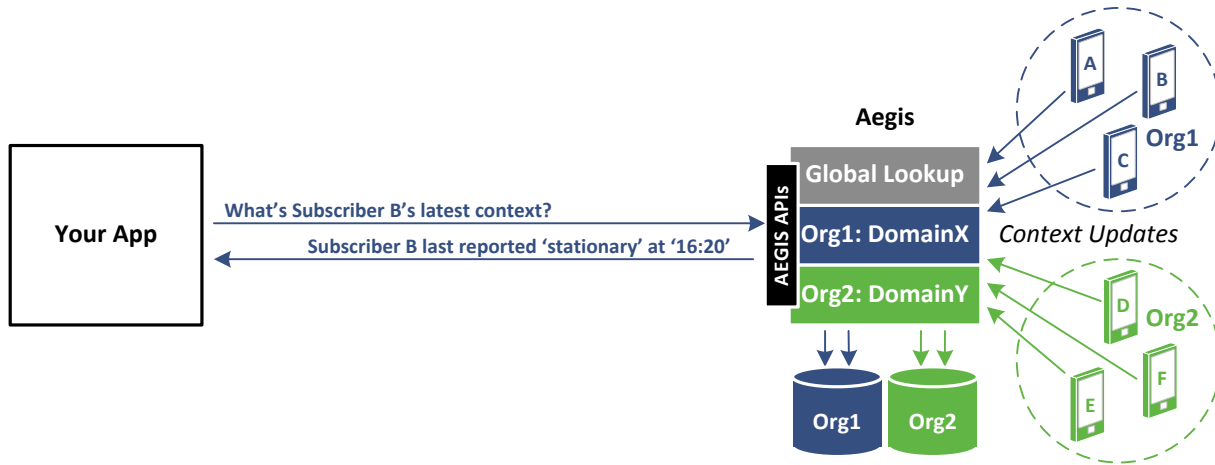


Figure 3. Registering for Context-Change Callbacks

The service returns a JSON-formatted string (see 4.3.2) that contains the status of the request, an array of context states (Moving, Geozone, and Point of Interest), the time the context last changed, and a context engine configuration ID.

4.3.1 `get_application_state` - Request

```
GET https://<domain>/G/j/csvc/context/<partner id>/<org id>/<application id>/<msisdn>?ts=<timestamp>&tk=<signature>
```

To request the application state of a context-aware application running on a mobile device, you need the information in [Table 7](#).

6.3 Register URI for Callback

6.3.1 Prerequisites

- Your Callback URI: `http://mydomain/api/context`
- Domain: `api0.aegismobility.com`
- Partner ID: `795d21164a73413b9dfd7c8ee94ad5e6`
- Organization ID: `myorg`
- Application ID: `myapp`
- Secret: `jq9epBhsMdJ1sRtW8lcehbJczp45bsqCNXArNbIfYWZZpRt1Z0ye6JYij`

6.3.2 Send request

Build a request URI and assign it to a variable:

```
$aegis_service_url = 'https://api0.aegismobility.com/G/j/csvc/cb/795d21164a73413b9dfd7c8ee94ad5e6/myorg/myapp?ts=1334868021';
```

Your callback URI:

```
http://mydomain.com/api/context
```

Encode your callback URI:

```
$body = 'url=' . urlencode('http://mydomain.com/api/context');
```

Encoded POST body:

```
url=http%3A%2F%2Fmydomain.com%2Fapi%2Fcontext
```

Concatenate the POST body with the request URI:

```
$msg = $aegis_service_url . $body;
```

Generate signature:

```
getSignature($aegis_service_url, $secret);
```

Result:

```
07799870902c86ce70270b4ff6ff0e91
```

Complete message:

```
POST https://api0.aegismobility.com/G/j/csvc/context/795d21164a73413b9dfd7c8ee94ad5e6/myorg/myapp?ts=1334868021&tk=07799870902c86ce70270b4ff6ff0e91
```

```
url=http%3A%2F%2Fmydomain.com%2Fapi%2Fcontext
```

6.3.3 Receive response

```
{  
  "status":200,  
  "msg":"OK",  
  "data":""  
}
```

6.4 Receive a Callback

Message incoming from Aegis:

```
POST http://mydomain.com/api/context?ts=1334868021
&tk=54beb6b81e334068f3889f37e86384eb

dataset=context&msisdn=16045556666&MOVING=STATIONARY&MOVING_T=1334878313&GZONE
=OUTZONE&GZONE_T=1334878313&POI=OUTPOI&POI_T=1334878313&seq=1
```

Generate a signature and check it against the tk value in the request:

```
$url = 'http://mydomain.com/api/context?ts=1334868021';
# Use the following to get the raw (not urldecoded) request
$body = file_get_contents('php://input');
$signature_to_check = $_GET['tk'];

$my_signature = getSignature($url . $body, $secret);

if ($my_signature == $signature_to_check) {
    # Do something with the data
    $msisdn = $_POST['msisdn'];
    $moving = $_POST['MOVING'];
    $moving_time = $_POST['MOVING_T'];
} else {
    # Invalid request since the signatures don't match
    print "Request is bad, ignoring\n";
}
```