SafetyNet Web Services

Application Program Interface (API) JSON Supplement

June 28, 2017

Predictive SOLUTIONS

.

Contents

This document is intended to supplement the SafetyNet Web Services Application Program Interface (API) Reference Document. It documents the differences in request formats, endpoint URLs and data formats. It does not document entity types, request/response payloads or other information that is common to both SOAP and JSON web services, as that can be found in the main API document and/or WSDL.

All operations that are available via the SOAP web service and documented in the web services API are available via JSON HTTP requests.

Request HTTP Headers

All HTTP requests must contain the following HTTP headers:

Name	Description	Examples
snt-authorization	This is how authentication information is sent to the web service. See the Authentication section for more details.	BASIC ANNCD76CBNQ22d…
Accept	Specifies the content-type to use with the web service. This must be included or responses will not be formatted correctly.	application/json
snt-version	The version of the web service to invoke.	v1_1

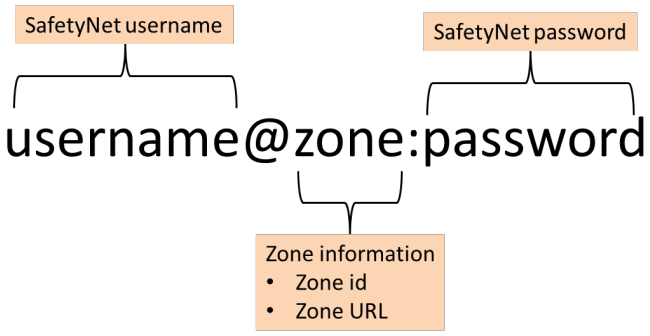
Authentication

All requests to the web service must include the "snt-authorization" HTTP header in the request for authentication. The JSON web service supports HTTP Basic Authentication. See http://tools.ietf.org/html/rfc2617 for more information.

Basic authentication allows users to invoke the JSON web service using a username, password, and zone information (id or url). The HTTP "snt-authorization" uses the following format:

```
BASIC <encoded_credentials>
```

where <encoded_credentials> is the following string encoded in base64:



When using a zone url, should not contain http or https, only the domain and port, such as "company.predictivesolutions.com:8080[i1]"

Some examples of BASIC authentication and the equivalent base64 encoded string:

```
admin@10018:password =
      YWRtaW5AMTAxMjk6cGFzc3dvcmQg

admin@company.predictivesolutions.com:password =
   YWRtaW5AY29tcGFueS5wcmVkaWN0aXZlc29sdXRpb25zLmNvbTpwYXNzd29yZA==
```

JSON Operations

All operations that are exposed via the WSDL(s) are available via the JSON web service as well. The data formats are the same as the SOAP counterparts, with the exception that payloads use JSON instead of XML and that JSON payloads are not wrapped in a parent element. For example, the SOAP request:

```
<v1:IncidentQueryRequest>
         <v1:IncidentSearchCriteria>
            <maxResults>25</maxResults>
         </v1:IncidentSearchCriteria>
</v1:IncidentQueryRequest>
```

would have the JSON equivalent of:

```
{
      incidentSearchCriteria": {
            "maxResults": 25
      }
}
```

The table below shows the relationship between SOAP endpoints and the corresponding JSON endpoints:

SOAP Operation	Operation URL	Method	Payload
QueryRequest	/query	POST	<entity>SearchCriteria
InsertRequest	/insert	PUT	<entity> (without ID)
UpdateRequest	/update	PUT	<entity> (with ID)
DeleteRequest [5]	/delete/{ID}?modifyDate=123456789	DELETE	none (ID and modifyDate in URL)
FindByIDRequest	/findByID/{ID}	GET	none (ID in URL)
GetIDsRequest	/getIDs	POST	<entity>SearchCriteria
CodesRequest [3]	/codes	POST	filter
CountRequest	/count	POST	<entity>SearchCriteria
AssociationQueryRequest	/association/query	POST	<entity>AssociationSearchCriteria
AssociationInsertRequest	/association/insert	PUT	varies

[1] company refers to the company named created for a zone url mapping in the SafetyNet system. If you are unsure of what the value should be, contact support. This value is also the subdomain used when accessing the SafetyNet web application.

AssociationUpdateRequest	/association/update	PUT	varies
AssociationDeleteRequest [4]	/association/delete	POST	varies

1. To determine the JSON service endpoint URL, append the operation URL to the service endpoint (ex. *http://ws.predictivesolutions.com/service/json/ContactService*
2. Observation operations are part of InspectionService. Use the same url part as above, with observation in front of the operation. For example, observation query would be *../InspectionService/observation/query*, observation delete would be *../InspectionService/observation/delete/12345*
3. There is not an operation for observation codes, use InspectionCodesRequest.
4. Association delete request uses a POST method because multiple IDs can be sent in one request.
5. modifyDate is passed as a query parameter for entity delete requests.

In addition to the endpoint operations above, there is a generic endpoint that provides common operations for all the entities.

SOAP Operation	Operation URL	Method	Payload
ConfigFieldsRequest	/ConfigService/fields	GET	none

1. To determine the JSON service endpoint URL, append the operation URL to the service endpoint (ex. *http://ws.predictivesolutions.com/service/json/ConfigService*)

Important Data Type Notes

Case Sensitivity

JSON requests use the same data element names as SOAP. However, due to the way that schemas are translated into JSON data structures, there are a few exceptions with regards to case sensitivity of property names. As a general rule, the first letter of a property name will be lowercase.

SOAP Example

```
<v1:IncidentSearchCriteria>
      <maxResults>25</maxResults>
</v1:IncidentSearchCriteria>
```

JSON Example

```
{
      incidentSearchCriteria": {
            "maxResults": 25
      }
}
```

Enumerations

Enumerations use the same values as the SOAP web service. The only exception is the custom data enumeration. Due to a known issue with how schemas are generated into JSON payloads, use the JSON value in the table below for customDataType:

SOAP Value	JSON Value
customDataText1	CUSTOM_DATA_TEXT_1
customDataText2	CUSTOM_DATA_TEXT_2
customDataText3	CUSTOM_DATA_TEXT_3
customDataText4	CUSTOM_DATA_TEXT_4
customDataText5	CUSTOM_DATA_TEXT_5
customDataText6	CUSTOM_DATA_TEXT_6
customDataText7	CUSTOM_DATA_TEXT_7
customDataText8	CUSTOM_DATA_TEXT_8
customDataText9	CUSTOM_DATA_TEXT_9
customDataText10	CUSTOM_DATA_TEXT_10
customDataText11	CUSTOM_DATA_TEXT_11
customDataText12	CUSTOM_DATA_TEXT_12
customDataDate1	CUSTOM_DATA_DATE_1
customDataDate2	CUSTOM_DATA_DATE_2
customDataDate3	CUSTOM_DATA_DATE_3
customDataDate4	CUSTOM_DATA_DATE_4
customDataNumeric1	CUSTOM_DATA_NUMERIC_1
customDataNumeric2	CUSTOM_DATA_NUMERIC_2
customDataNumeric3	CUSTOM_DATA_NUMERIC_3
customDataNumeric4	CUSTOM_DATA_NUMERIC_4

Attributes

Attributes are used in the SOAP web service. However, JSON does not support attributes. Attributes are treated the same as normal properties on the object:

SOAP Example

```
<customData customDataType="customDataText1">
      <Text>Test custom data value</Text>
</customData>
```

JSON Equivalent

```
{
      customDataType: "CUSTOM_DATA_TEXT_1",
      text: "Test custom data value"
}
```

Dates

Dates are sent in JSON requests as numeric values, representing the number of milliseconds since the epoch date. It is the responsibility of the client to convert this value to a readable format.

Binary Data

Binary data for attachments and images use base64 to encode data. It is the responsibility of the client to perform this conversion. For the purposes of request payloads, binary data that is base64 encoded is sent as a string, but the web service will attempt to decode this value when processing the request. Sending raw binary data will result in an error.

Inspection Insert Example

As previously mentioned, this document is only used as a supplement to the SafetyNet Web Services API document. The data structures are discussed in detail, and the same structure is used for both SOAP and JSON. The difference is that the payload for SOAP requests are represented as XML, and the JSON web service uses strict JSON.

The following example shows an insertion of an inspection with one observation.

Request

PUT https://sandbox.predictivesolutions.com/service/InspectionService/insert

Headers

```
snt-authorization: BASIC
YWRtaW5Ac2FuZGJveC5wcmVkaWN0aXZlc29sdXRpb25zLmNvbTpwYXNzd29yZA==
snt-version: v1_1
Accept: application/json
```

Message Body

```
{
      "inspectionDate": "1417073362653",
      "projectID": 204386,
      "reviewedWith": "reviewer",
      "comments": "Test Comments",
      "inspectorID": 308112,
      "observation": [
            {
             "comments": "safe",
             "safe": true,
             "categoryID": 146361
            }
      ]
}
```

Troubleshooting

Response is XML instead of JSON (unable to marshal type ... as an element because it is missing an @XmlRootElement annotation

Include the “Accept” header in the request and set to “application/json” to ensure that the response is formatted in JSON.

ResponseCode 2, "Message transformation failed."

The request body is not valid JSON, use a JSON formatter to ensure that it validates.

HTTP/1.1 405 Method Not Allowed

The URL is incorrect. Please see a list of valid URLs in the JSON Operations section above.