WOSTMANN & ASSOCIATES, INC.

Basis Public Service

API

Author: Michael Wener

Date: January 2, 2014

Version: 1.0

Submitted by

Wostmann & Associates, Inc.

TABLE OF CONTENTS

TABLE OF FIGURES

No table of figures entries found.

1 Introduction

1.1 Summary

It is very common for modern Web Applications to require serving content in both a form readable by humans as well as in a form consumable by another machine. The human eye reads rendered web pages but another machine needs a more efficient form to obtain the information. In the case of a web page Dynamic HTML provides the best format to serve. In the case of another machine a Web Service is the preferred method to serve content. The software industry has established a de facto set of approaches in solving these dual needs.

This document details an API providing access to the Alaska State Legislature BASIS web service. The API is REST based and is performed through HTTP requests. The responses are in structured XML.

2 Public Service Tier

2.1 XML Schema

An XML Schema called `basis.xsd` defines the XML content needed to produce the responses from the service tier. The primary components of the schema are Bills, Committees, Members, Meetings and Sessions. The schema contains comments describing the elements and types and their purpose. It describes the data types used in the REST exchanges. The XML described by this schema is contained in the HTTP response bodies.

The basis schema can be found at http://www.legis.state.ak.us/basis/basis.xsd

2.2 REST Requests

The REST implementation being proposed in this document uses HTTP. An HTTP request is driven by an HTTP method, query parameters and the headers that are part of the request. The sections that follow will define the valid methods, query parameters and the headers used as part of the Basis service interface.

2.2.1 Resources

The Basis service will expose a set of resources that are available via the service. These resources are the state that is transferred in a RESTful manner. These are identified using URLs. The XML Schema (see `basis.xsd`) contains a definition of the resources accessed via the RESTful interface.

2.2.2 URL Path

All HTTP methods require a URL Path for access. This section defines the path format used to access Basis state. The general format is as follows:

```
/basis/<section>[/<subsection>][?<parm>=<value>[&<parm>=<value>]*]

      <section> := bills | members | committees | sessions

The <section> value serves to specify the collection category the method
will target.

A limited set of standard query parameters are allowed as well. The
supported parameters are as follows:

            <parm> : = session | chamber | minifyresult

The <value> used with the session and chamber parm corresponds to the
session number or chamber code.

For minifyresult, if false is specified then the whitespace will not be
stripped from the result message body. Default is to minify all results.
```

2.2.3 Headers

Http Headers

The Content-Encoding gzip is the only supported encoding other than no encoding. If the client desires the content to be compressed using the gzip algorithm then a standard Content-Encoding header specifying support for gzip must be used.

```
Accept-Encoding: gzip;q=1.0
```

Service Specific Headers

The format for any headers specific to Basis is of the form:

```
X-Alaska-Legislature-Basis-<suffix>: <Header Value>
```

The format for any headers not specific to Basis is of the form:

```
X-Alaska-<suffix>: <Header Value>
```

All Basis service requests and responses shall contain a version header. If this header is missing the request shall be rejected with a 400 status code. The format is the following:

```
X-Alaska-Legislature-Basis-Version: 1.0
```

Query Headers

The following headers are Basis general query headers. They cover parameter based searching schemes.

A request can contain multiple query headers. The result of multiple query headers is to combine the entire list of comma separated values into one long query processing string.[1] Multiple headers are a cumulative constraint on the result set.

[1] This style of combination of header values is specified in the HTTP standards. Proxies are required to pass all headers with the same header name in the original order received.

```
X-Alaska-Legislature-Basis-Query: <Query>[,<Query>]*

      <Query> := <Include>[;<Constraint>[=<Value>]?]*

      <Include> := <Bill> | <Session> | <Committee> | <Member> | <Meeting>

      <Bill> := Sponsors | Actions | Subjects | Versions | Fiscalnotes

                | Votes | Bills

      <Session> := Journals

      <Committee> := Members | Meetings | Bills | Committees

      <Member> := Committees | Bills | Votes | Members

      <Meeting> := Media | Minutes | Meetings

      An Include identifies which component of a section instance is being
      requested to be returned in the query. It also provides scope for the
      related constraints. So each include can have a series of constraints
      that apply to XML elements related to the Include.

      Sponsors

      <Constraint> := <See section on Constraints>

      A Constraint identifies how to constrain the immediately preceding
      Include. There may be multiple Constraint values following an Include.

      <Value> := <Text>

      Text values may include an '*' as a wildcard character.
```

Constraints

The set of legal constraints can be found by sending an OPTIONS HTTP request for a particular resource.

Include	Legal Constraints
Actions	code;text;date;startdate;enddate;page;startpage;endpage
Bills	name;chamber;bill;onfloor;shorttitle;statustext;statuscode;committeecode; fulltext;subject
Committees	category;chamber;location;name;title;code

Fiscalnotes	chamber;date;startdate;enddate;startpage;endpage
Journals	chamber;date;startdate;enddate;page;startpage;endpage;fulltext
Media	
Meetings	date;startdate;enddate;details
Members	building;chamber;comment;district;party;phone;firstname;lastname;ismajority;code
Minutes	fulltext
Sponsors	building;chamber;comment;district;party;phone;firstname;lastname;ismajority;code
Subjects	text
Versions	fulltext
Votes	vote;title

Most Constraints follow the *key=value* pattern. Where *value* is a pattern match allowing any number of wild card characters. An asterisk is the wild card character.

The constraint *fulltext* specifies that the full text of the corresponding Document Type should be included. The constraint *fulltext= urlonly* specifies that only the url is to be returned.

The constraint *ismajority* accepts the strings *true* or *false*. If no string is given it defaults to *true*. If any other string besides *true* is given it defaults to *false*.

Range Header

This header works as a qualifying constraint on other headers. For example, a query header would be constrained by limiting the result set within some specified range.

```
X-Alaska-Query-ResultRange: [<N>][".."<M>]

      N := PositiveInteger

If a single integer value is given the first N values will be returned. If a
range value is given then the results from N to M inclusive are returned. If
N is missing then the last M values are returned.
```

Response Headers

When a query is run against the server these headers are used to return some meta-information about the quality of the query.

```
X-Alaska-Query-Count: <N>

      N := PositiveInteger

The response shall contain this header. The value containing the number of
entities returned in the query.
```

```
X-Alaska-Legislature-Basis-Version: <N>.<N>

The response shall contain this header. The value containing the version of
service the Server is running.
```

```
X-Alaska-Query-Result-Information: <key>=<value>,*

The response may contain this header. The header will contain a
miscellaneous collection of key value pairs separated by commas. These pairs
are information the server may wish to communicate to the client.
```

Query Header Examples

The header combination below will return bills from session 26 that have documents with title text containing "Oil", and include their list of Actions.

```
GET /basis/bills?session=26 HTTP/1.1

Host: basisservicehost.legis.state.ak.us:80

...

X-Alaska-Legislature-Basis-Version: 1.0

X-Alaska-Legislature-Basis-Query: Bills; title=*Oil*

X-Alaska-Legislature-Basis-Query: Actions
```

The header combination below will return bills from the current session along with their associated sponsors and subject information if the subject is FORESTRY.

```
GET /basis/bills HTTP/1.1

Host: basisservicehost.legis.state.ak.us:80

...

X-Alaska-Legislature-Basis-Version: 1.0

X-Alaska-Legislature-Basis-Query: Sponsors

X-Alaska-Legislature-Basis-Query: Subjects; Text=FORESTRY
```

If you want to return bills with FORESTY as their subject then the following query would be made.

```
GET /basis/bills HTTP/1.1

Host: basisservicehost.legis.state.ak.us:80

...

X-Alaska-Legislature-Basis-Version: 1.0

X-Alaska-Legislature-Basis-Query: Bills; Subject=FORESTRY
```

The header combination below will return all house bills from session 26, but only with their summary information (see `basis.xsd` for a description of what is considered summary information for a bill).

```
GET /basis/bills?session=26&chamber=H HTTP/1.1

Host: basisservicehost.legis.state.ak.us:80

...

X-Alaska-Legislature-Basis-Version: 1.0
```

The header combination below will return the last 10 house and senate bills from session 26, but only with their summary information

```
GET /basis/bills?session=26 HTTP/1.1

Host: basisservicehost.legis.state.ak.us:80

...

X-Alaska-Legislature-Basis-Version: 1.0

X-Alaska-Query-ResultRange: ..10
```

2.2.4 Verb Definitions

GET and HEAD are considered safe methods by the HTTP standards. This means that these two methods will **never** cause modifying side effects on the server. For the Basis service they both provide state retrieval. In the case of GET actual XML content is returned as part of the retrieval and in the case of HEAD no body is returned. Only headers are returned indicating information relevant to the request.

GET	Query resource for specified content.
HEAD	Query resource for specified content and return no content. This can be used to obtain count values.

The OPTIONS method is considered a safe method from the standpoint of Basis. This method can be used by a client to discover all legal header values from the server for a specified resource.

OPTIONS	Discover legal operations from the Basis server.

2.3 REST Responses

2.3.1 Warnings and Errors

A Basis response may contain Warning and/or Error elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<Basis xmlns="http://www.legis.state.ak.us/Basis">
   <Warning xmlns="">
      <Code>DataMayBeInvalid</Code>
      <Description>This is a test system. Data may be inaccurate.</Description>
   </Warning>
   <Error xmlns="">
      <Code>BaoSystemFault</Code>
      <Description>Invalid Session Number.</Description>
   </Error>
</Basis>
```

If the response contains a Warning element then the response may still be usable. If the response contains an Error element then the response should be considered a failure and no data besides what is contained in the Warning and Error elements should be used.

3 Client Optimizations

Request performance can differ significantly depending on the parameters passed via the REST request. In order to optimize a users experience it is suggested that broad queries restrict the level of detail being requested. When detail regarding a specific bill is needed, for example, it is better to structure the requests in the form of fragment requests using AJAX style requests.