

GeoWS Web Service Template

Version 1.1
2016-09-15

Purpose

The primary purpose of this document is to specify the common characteristics of service interfaces developed as part of the EarthCube GeoWS - Geoscience Web Services Building Block project. A core principle of the project is to create interfaces with as little difference in design, and therefore usage, as possible.

Goal

As part of the GeoWS project, the initial goal is to create interfaces for discovery and access to identified data sets using the same (or very similar) query parameters for a space-time query and return the information in simple text or image formats. Later goals include the delivery of data in a small set of commonly used formats and enhanced selection criteria. In addition to adhering to the template defined in this document, GeoWS service interfaces are expected to implement additional parameters and output formats to support domain-specific needs.

Common Service Characteristics

Versioning

A common *major.minor.implementation* (X.Y.Z) scheme, where only the major version is meaningful across all data centers and follows the Template version.

Calling pattern

The services will be invoked using a subset of REST and HTTP methods. In particular, all services will be invoked using the HTTP GET method whenever possible and the HTTP POST method for the few cases where the selection parameters are potentially large in size.

Service path and port

Each service/data center is allowed to create arbitrary endpoints and are encouraged to include capabilities for versioning and future evolution. Each end point should be fully documented as described later in this document.

All services are strongly recommended to be offered on TCP/IP port 80, all production level services should be on port 80.

Common Service Responses

General responses that should be used by all services are outlined below.

No data selected

If a properly formatted request is submitted but would result in no data being returned to the user the service should return either **HTTP status 204** (No Content), **HTTP status 404** (Not Found) or an empty container (with **HTTP status 200**). If a service will support both 204 and 404 status codes, the *nodata* parameter should be supported to allow a client to select between them.

Result set limitations

Limitations on the amount of information returned for any given request may be imposed independently for each service by each data center. If a client submits a request that would result in a data set beyond the service limit the service should return an **HTTP status 413** (Request Entity Too Large). A **413** should also be returned when the request itself is too large, the standard meaning of the status. In most cases, differentiating between these two cases will be obvious.

Errors messages

All errors reported to the client, either HTTP 4xx or 5xx status codes, should include an error message transmitted as MIME type **text/plain** using the following pattern:

Error <CODE>: <SIMPLE ERROR DESCRIPTION>

<MORE DETAILED ERROR DESCRIPTION>

Usage details are available from <SERVICE DOCUMENTATION URI>

Request:
<SUBMITTED URL>

Request Submitted:
<UTC DATE TIME>

Service version:
<3-LEVEL VERSION>

Common HTTP status codes

All services will use appropriate HTTP 1.1 status codes in interactions with clients. Any modifications or extensions to the standard status code meanings will be documented in this template.

Table 1 includes a list of common status codes returned conforming services. In most cases these

will be the only codes returned by the services. Under error, maintenance or otherwise unusual conditions a client may receive other HTTP codes generated by web service containers, layer 7 routers (e.g. load balancers), firewalls, etc.

Table 1. common HTTP status codes returned by GeoWS services

Code	Description
200	Successful request, results follow
204	Request was properly formatted and submitted but no data matches the selection
301	Moved permanently - return new location (highly recommended for web services to assist user with URL changes)
400	Bad request due to improper specification, unrecognized parameter, parameter value out of range, etc.
401	Unauthorized, authentication required
403	Authentication failed or access blocked to restricted data
404	A synonym for 404, but can also be produced under error conditions (e.g. proxy response when backend has failed).
413	Request would result in too much data being returned or the request itself is too large returned error message should include the service limitations in the detailed description.
414	Request URI too large
500	Internal server error
503	Service temporarily unavailable, used in maintenance and error conditions

Common request parameters

The common request parameters for each service are summarized in Table 2. In this Table, the **alternative** names are acceptable synonyms for the given **parameter** name. Each service interface will very likely support additional parameters that are specific to the data set or traditional user base.

Table 2. Common parameter definitions:

	Parameter	Alternative	Default	Min/Max	Type	Units
Time	starttime		[Any]	[Any]	Time	UTC[+TZ]
	endtime		[Any]	[Any]	Time	UTC[+TZ]
	time		[Any]	[Any]	Time	UTC[+TZ]
	geologicstartage		[Any]	[Any]	Time	Ma
	geologicendage		[Any]	[Any]	Time	Ma
	geologicage		[Any]	[Any]	Time	Ma
Geographic	minlatitude	south	-90	-90/90	Float	Decimal Degrees N
	maxlatitude	north	90	-90/90	Float	Decimal Degrees N
	minlongitude	west	-180		Float	Decimal Degrees E
	maxlongitude	east	180		Float	Decimal Degrees E
	latitude					Decimal

Identification	longitude				Degrees N Decimal Degrees E
	station	site	[Any]	String	
Output control	format	output		String	

Table 3. Parameter descriptions:

Parameter	Description
starttime	Specify results beginning at or after given time
endtime	Specify results ending at or before given time
time	Specify results occurring at given time
geologicstartage	Specify results beginning at or after given geologic age
geologicendage	Specify results beginning at or before given geologic age
geologicage	Specify results occurring at given geologic age
minlatitude	Specify the Southern boundary
maxlatitude	Specify the Northern boundary
minlongitude	Specify the Western boundary
maxlongitude	Specify the Eastern boundary
station	Limit to results from specified station, site, etc.
format	Return results is specified format: <i>text, XML, ...</i>

Result set format:

Preferably, a client may request the results to be formatted using a *format* (or *output*) parameter. This allows for the greatest flexibility and easiest use by end users. Additionally, a service implementation may allow result set selection using requested media-type in the request HTTP header.

Result that is a list of resources

In some cases, a request may be fulfilled by returning a list of resources for the requested data, and the client must fetch the listed resources individually. The returned list of resources should use a common format.

The list of resources will be structured following the IETF [BagIt File Packaging Format](#) specification for a "Payload Manifest". Resources will be listed in a text tag file with one resource per line, of the form:

URL [CHECKSUM [LENGTH [MIME-Type]]]

Each line is terminated by carriage return (CR), and fields are separated by at least one whitespace character. The URL must not contain whitespace. The CHECKSUM is optional, and is calculated using the MD5 algorithm. The LENGTH is optional, express in integer bytes (file size), and must be preceded by a checksum. The MIME-Type is optional, with values defined by registries such as

[IANA](#) and vendor-specific extensions, and must be preceded by a length.

Wildcards and lists in query parameters

When wildcards are supported for parameter values, these wildcard characters should be supported with the following meaning:

- * – Matches zero to many characters
- ? – Matches any single character

Time parameter values

All **time** values are specified in UTC and should use one particular variation of the ISO 8601 standard with the addition of sub-seconds:

YYYY-MM-DDTHH:MM:SS.sssss[±HH[:MM] or Z]

Where the fields are as specified in ISO 8601 and a “T” separates the date and time. The “**sssss**” are the sub-seconds expressed between 1/10 of a second to microsecond resolution; they do not all need to be present. The time zone specification is optional can be expressed as either **±HH[:MM]**, where the minutes are optional, or simply **Z** for UTC.

Two shorter variations for specifying time should be supported:

YYYY-MM-DDTHH:MM:SS (implies a time of **YYYY-MM-DDTHH:MM:SS.000000**)

where the sub-seconds are interpreted to be zeros and

YYYY-MM-DD (implies a time of **YYYY-MM-DDT00:00:00.000000**)

where the hour, minute, second and sub-second are interpreted to be zeros.

** Consider adoption of the 8601 subset defined in RFC 3339*

String, float, boolean and integer parameter types

String type parameters may include ASCII characters, including the asterisk (*) and question mark (?) that have special meaning as wildcards for specific parameters.

Float type parameters should be specified in decimal notation, for example 98.1023. Scientific notation using an exponent or other representations will not be accepted and should generate errors.

Boolean type parameters should be specified as either **TRUE** or **FALSE**, case insensitive.

Integer type parameters should be specified in decimal notation without a decimal point, exponential or other notation.

Time parameter interpretation

The *starttime* and *endtime* parameters are inclusive. The *starttime* parameter should be

interpreted as selecting any data or information at times on or later than the value specified. Similarly, the *endtime* selects any data or information at times on or prior to the value specified.

Requests submitted via HTTP POST

For the POST method, all parameters must be submitted as part of the POST body. The non-repeated key-value parameters should be specified as key=value pairs on separate lines, followed by the repeated values (defined in documentation) that are listed as many times as necessary following this pattern:

```
key1=value1
key2=value2
key3=value3
value1 value2 value3 STARTTIME ENDTIME
value1 value2 value3 STARTTIME ENDTIME
value1 value2 value3 STARTTIME ENDTIME
```

This would be POSTed to the service's query method.

All rules for parameters apply equally whether specified using the GET or POST.

Documentation

End user documentation

Each service should have full documentation. This documentation is designed for human readability and describes the purpose of the service and includes a description of each parameter.

Ideally, documentation for service endpoints will be served, in HTML, from the "root" location of the service interface or otherwise easily discoverable.

The root documentation either includes directly or a link to a dated service revision list. The root documentation also includes a link to a "URL Builder".

URL Builder web tool

A web-based "URL Builder" accompanies each service end-point. Linked from the root documentation, the purpose of a Builder is to allow easy exploration and query prototyping of a service. This online tool presents a form with input boxes for each parameter; as users complete each input field a URL representing the query is created. The created URL is clickable and can be cut and pasted as needed.

Machine-readable documentation

The recommended machine-readable documentation language is [Swagger's Open API Specification](#),

formally Swagger Specification, but now [Linux Foundation - Open API Initiative](#).

The API specification should be available at `/v2/swagger` relative to the service interface, e.g. `http://<host>/<service-context-path>/v2/swagger`.

Common annotation for simple text formats

When appropriate service endpoints should offer highest value data, or a subset of data, in a simple, easy to read text format whenever possible. The GeoCSV text format specification should be used to describe the data set: <http://geows.ds.iris.edu/documents/GeoCSV.pdf>