

FDSN Web Service Specifications

Version 1.0
DRAFT 2012/7/18

Purpose

To specify web service interfaces for the exchange of time series data, related metadata, event parameter and other data within the context of the International Federation of Digital Seismograph Networks (FDSN). The intention is to provide a specification that, when implemented at different FDSN data centers, can be used interchangeably by the same client software. The specification defines service names, query parameters and expected results.

List of service interfaces

dataselect – returns raw time series data in FDSN miniSEED format

station – returns metadata in FDSN StationXML format

event – returns parametric data for events in QuakeML

Common service characteristics

Versioning

The services are versioned according the following three-digit (x.y.z) pattern:

SpecMajor.SpecMinor.Implementation

Where the fields have the following meaning:

SpecMajor: The major specification version, all implementations sharing this *SpecMajor* value will be backwards compatible with all prior releases. Values start at 1.

SpecMinor: The minor specification version, incremented when optional parameters or behavior is added to the previous specification but backwards compatibility is maintained with the previous major versions, i.e. all 1.# service versions will be compatible with version 1.0. Values start at 0.

Implementation: The implementation version, an integer identifier specific to the data center implementation. Useful to track service updates for bug fixes, etc. but with no implication on conformance to the specification.

Together the *SpecMajor* and *SpecMinor* versions imply the expected behavior of a given service. This versioning scheme allows clients to expect specific behavior based on the *SpecMajor* version, while allowing the FDSN to extend the services with optional parameters while maintaining backwards compatibility. Each version number is service specific, there is no implication that *SpecMajor* version numbers across services are related.

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

The following base URI pattern is to be used at each data center implementing FDSN services:

`<site>/fdsnws/<service>/<majorversion>/`

where **service** is one of *dataselect*, *station* or *event* and *SpecMajor* **version** is an integer value specifying the major specification version supported by the service. A **site** is the domain name of the data center hosting FDSN web services. For example the base URI for version 1 of the FDSN station service at ORFEUS might be:

`www.orfeus-eu.org/fdsnws/station/1/`

All services should be available on TCP/IP port 80.

Common Service methods

All services should support these common methods:

query – to submit a data or information request

version – to request the full service version (*SpecMajor.SpecMinor.Implementation*)

application.wadl – to request a WADL for the interface

Details of each method and further methods specific to a service are documented in the relevant service descriptions.

For example, submitting queries to the station service at ORFEUS would be performed using this URI:

`http://www.orfeus-eu.org/fdsnws/station/1/query?<key=value parameters>`

Similarly requesting the implementation version of the same service would be done using this URI:

`http://www.orfeus-eu.org/fdsnws/station/1/version`

which returns as simple text string indicating the software version, e.g. "1.2.8".

Requesting the WADL for a the same interface:

`http://www.orfeus-eu.org/fdsnws/station/1/application.wadl`

Any data center implementing one or more of these service interfaces should implement all methods and query parameters specified and conform to the calling patterns and expected results identified in this document to be considered conforming with the specification. The exception is authentication methods that only need to be supported by data centers that wish to require authentication for access to specific data sets.

Request URI construction

Request URIs to FDSN web services must not exceed 2000 bytes in length including the URL encoding; most common requests will be much shorter. Requests should be URL Encoded as a standard practice before submission to the service, but services should handle non-encoded submissions as far as is possible.

Common Service responses

General responses that should be used by all services are outlined below. Responses specific to each service are detailed in later sections.

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 **HTTP status 204** (No Content).

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).

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

Table 1 includes a list of common status codes returned by FDSN web 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 FDSN services

Code	Description
200	Successful request, results follow
204	Request was properly formatted and submitted but no data matches the selection
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
413	Request would result in too much data being returned, returned error message should include the service limitations in the detailed description. Service limits should also be documented in the service WADL.
414	Request URI too large
500	Internal server error
503	Service temporarily unavailable, used in maintenance and error conditions

Request parameters

The common and specific request parameters for each service are summarized in Table 1. In this Table, the **abbreviation** names are acceptable synonyms for the given **parameter** name. For a specific service and parameter group, each sub-group parameter set is exclusive; for example, for a call to the **fdsnws-event** service the client should supply a simple-time or window time definition but not both. Service specific meanings for the parameters are described later in this document.

Table 1. Parameter specifications for FDSN web services, including applicable services for each group in italics.

Group	Subgroup	Parameter	Abbreviation	Default	Minimum	Maximum	Type	Units
time constraints								
	simple-time	<i>fdsnws-dataselect, fdsnws-station, fdsnws-event</i>						
		starttime	start	[Any]	[Any Valid Time]		time	UTC
		endtime	end	[Any]	[Any Valid Time]		time	UTC
	window-time	<i>fdsnws-station</i>						
		startbefore		[Any]	[Any Valid Time]		time	UTC
		startafter		[Any]	[Any Valid Time]		time	UTC
		endbefore		[Any]	[Any Valid Time]		time	UTC
		endafter		[Any]	[Any Valid Time]		time	UTC
channel constraints								
		<i>fdsnws-dataselect, fdsnws-station</i>						
		network	net	[Any]	[Valid ASCII or * or ?]		string	[Not Applicable]
		station	sta	[Any]	[Valid ASCII or * or ?]		string	[Not Applicable]
		location	loc	[Any]	[Valid ASCII or * or ?]		string	[Not Applicable]
		channel	cha	[Any]	[Valid ASCII or * or ?]		string	[Not Applicable]
geographic constraints								
	area-rectangle	<i>fdsnws-station, fdsnws-event</i>						
		minlatitude	minlat	-90.0	-90.0	90.0	float	degrees
		maxlatitude	maxlat	90.0	-90.0	90.0	float	degrees
		minlongitude	minlon	-180.0	-180.0	180.0	float	degrees
		maxlongitude	maxlon	180.0	-180.0	180.0	float	degrees
	area-circle	<i>fdsnws-station, fdsnws-event</i>						
		latitude	lat	0.0	-90.0	90.0	float	degrees
		longitude	lon	0.0	-180.0	180.0	float	degrees
		minradius		0.0	0.0	180.0	float	degrees
		maxradius		180.0	0.0	180.0	float	degrees
service-specific constraints								
	<i>fdsnws-station</i>							
		level		station	One of: <i>network, station, channel</i> or <i>response</i>		string	[Not Applicable]
		includerestricted		TRUE	[Not Applicable]		boolean	[Not Applicable]
		includeavailability		FALSE	[Not Applicable]		boolean	[Not Applicable]
		updatedafter		[Any]	[Any Valid Time]		time	UTC
	<i>fdsnws-event</i>							
		mindepth		[Any]	[Any]	[Any]	float	kilometers
		maxdepth		[Any]	[Any]	[Any]	float	kilometers
		minmagnitude	minmag	[Any]	[Any]	[Any]	float	Defined by the mag type
		maxmagnitude	maxmag	[Any]	[Any]	[Any]	float	mag type
		magnitudetype	magtype	[Any]	[Not Applicable]		string	[Not Applicable]
		includeallmagnitudes		FALSE	[Not Applicable]		boolean	[Not Applicable]
		includearrivals		FALSE	[Not Applicable]		boolean	[Not Applicable]
		preferredonly		TRUE	[Not Applicable]		boolean	[Not Applicable]
		eventid		[Any]	[Not Applicable]		string	[Not Applicable]
		limit		[Any]	0	[Any Positive]	integer	[Not Applicable]
		offset		1	1	[Any Positive]	integer	[Not Applicable]
		orderby		time	One of: <i>time, time-asc, magnitude</i> or <i>magnitude-asc</i>		integer	[Not Applicable]
		contributor		[Any]	[Not Applicable]		string	[Not Applicable]
		catalog		[Any]	[Not Applicable]		string	[Not Applicable]
		updatedafter		[Any]	[Any Valid Time]		time	UTC
	<i>fdsnws-dataselect</i>							
		quality		B	SEED Quality (D, R, Q, M) or B		string	[Not Applicable]
		minimumlength		0.0	0.0	[Any positive]	float	[Not Applicable]
		longestonly		FALSE	[Not Applicable]		boolean	[Not Applicable]

Wildcards and lists in channel constraints parameters

The channel constraint parameters (network, station, location and channel) may include two specific wildcard characters with the following meaning:

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

The channel constraint parameters may also be submitted as comma-separated lists in order to select two or more values with a single request. For example, the channel parameter may be used to specify multiple channels:

channel=LHE,LHN,LHZ,BHZ (the individual values may also include wildcards)

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.ssssss

Where the fields are as specified in ISO 8601 and a “T” separates the date and time. The “**ssssss**” are the sub-seconds expressed between 1/10 of a second to microsecond resolution; they do not all need to be present.

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.

Time parameter interpretation

The *starttime* parameter should be interpreted as selecting any data or information (time series samples, earthquake origins, metadata epochs) 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.

The *startbefore*, *startafter*, *endbefore* and *endafter* parameters, used only for the **fdsnws-station** service, specify the selection of information strictly starting or ending before or after a specified time value; they do not match any information occurring exactly at the time specified. These selection parameters are specifically for metadata and are useful for matching information that would otherwise be difficult with the other time parameters.

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.

Blank SEED location identifier selection

Blank or empty location identifiers are encoded in SEED as two space characters (ASCII decimal 32). These may be specified by setting the *location* parameter to a string of two spaces, but care must be taken to properly encode the URI. For clarity and convenience, blank SEED location identifiers may also be specified by setting the *location* parameter to "--" (two dashes) which the service should translate to two space characters as needed.

fdsnws-dataselect version 1.0

The following methods shall be supported by the service:

- query** – to submit a data request
- queryauth** – to authenticate and submit a data request
- version** – to request the full service version number
- application.wadl** – to request a WADL for the interface

Requests using the **query** method:

The service shall accept requests formulated using the selection parameters identified in Table 1: *simple-time*, *channel constraints* and parameters specific to **fdsnws-dataselect**. Table 2 contains parameter descriptions for this service.

Table 2. Parameter descriptions for the **fdsnws-dataselect** service

Parameter	Description
starttime	Limit results to time series samples on or after the specified start time
endtime	Limit results to time series samples on or before the specified end time
network	Select one or more network codes. Can be SEED network codes or data center defined codes. Multiple codes are comma-separated.
station	Select one or more SEED station codes. Multiple codes are comma-separated.
location	Select one or more SEED location identifiers. Multiple identifiers are comma-separated. As a special case "--" (two dashes) will be translated to a string of two space characters to match blank location IDs.
channel	Select one or more SEED channel codes. Multiple codes are comma-separated.
quality	Select a specific SEED quality indicator, handling is data center dependent.
minimumlength	Limit results to continuous data segments of a minimum length specified in seconds,
longestonly	Limit results to the longest continuous segment per channel.

These parameters may be submitted using either of the HTTP GET or POST methods. The POST method is useful to allow a large number of request parameters to be submitted.

For the GET method, the parameters should be submitted as key=value pairs and may not be specified more than once; if a parameter is submitted multiple times the result is undefined.

For the POST method, all parameters must be submitted as part of the POST body. The *quality*, *minimumlength* and *longestonly* parameters should be specified as key=value pairs on separate lines and the *simple-time* and *channel constraints* parameters repeated as many times as necessary following this pattern:

```
quality=M
minimumlength=0.0
longestonly=FALSE
NET STA LOC CHA STARTTIME ENDTIME
NET STA LOC CHA STARTTIME ENDTIME
NET STA LOC CHA STARTTIME ENDTIME
```

This would be POSTed to the URI "<site>/fdsnws/dataselect/1/query".

All rules for parameters apply equally whether specified using the GET or POST methods with the exception of blank location IDs, which must be specified as "--" in the POST body due to spaces being used as the field separator.

Responses for the *query* method:

The results of a successful request should be returned as a stream of miniSEED using the MIME type **application/vnd.fdsn.mseed**. The exact nature of the miniSEED returned by this service is data center dependent, a center may choose to remove duplicated data or prune to the exact window specified by the caller. The time series is generally expected to represent unprocessed, “raw” recorded data as supplied by the network operator or data owner.

Behavior for the *queryauth* method

HTTP Digest Authentication (RFC 2617) should be requested from the client. After successful authentication, a request should be accepted and handled as per the *query* method. Authentication credentials (user database and associated licensing schemes) are data center specific.

Behavior for the *version* method

The service shall return the implementation version as a simple text string using the MIME type **text/plain**. Any parameters submitted with the method will be ignored.

Behavior for the *application.wadl* method

The service shall return a WADL conformant description of the interface using the MIME type **application/xml**. Any parameters submitted with the method will be ignored.

Behavior for the **error** conditions

All errors returned to the client should use the type and pattern described in the Common Service Responses section.

fdsnws-station version 1.0 (provisional until FDSN StationXML is defined)

The following methods shall be supported by the service:

query – to submit a data request

version – to request the full service version number

application.wadl – to request a WADL for the interface

Requests using the *query* method:

The service shall accept requests formulated using the selection parameters identified in Table 1: *time constraints, channel constraints, geographic constraints* and the parameters listed that are specific for **fdsnws-station**. Table 3 contains parameter descriptions for this service.

Table 3. Parameter descriptions for the **fdsnws-station** service

Parameter	Description
starttime	Limit to metadata epochs starting on or after the specified start time.
endtime	Limit to metadata epochs ending on or before the specified end time.
startbefore	Limit to metadata epochs starting before specified time.
startafter	Limit to metadata epochs starting after specified time.
endbefore	Limit to metadata epochs ending before specified time.
endafter	Limit to metadata epochs ending after specified time.
network	Select one or more network codes. Can be SEED network codes or data center defined codes. Multiple codes are comma-separated.
station	Select one or more SEED station codes. Multiple codes are comma-separated.
location	Select one or more SEED location identifiers. Multiple identifiers are comma-separated. As a special case "--" (two dashes) will be translated to a string of two space characters to match blank location IDs.
channel	Select one or more SEED channel codes. Multiple codes are comma-separated.
minlatitude	Limit to stations with a latitude larger than the specified minimum.
maxlatitude	Limit to stations with a latitude smaller than the specified maximum.
minlongitude	Limit to stations with a longitude larger than the specified minimum.
maxlongitude	Limit to stations with a longitude smaller than the specified maximum.
latitude	Specify the latitude to be used for a radius search.
longitude	Specify the longitude to be used for a radius search.
minradius	Limit results to stations within the specified minimum number of degrees from the geographic point defined by the latitude and longitude parameters.
maxradius	Limit results to stations within the specified maximum number of degrees from the geographic point defined by the latitude and longitude parameters.
level	Specify the level of detail for the results.
includerestricted	Specify if results should include information for restricted stations.
includeavailability	Specify if results should include information about time series data availability.
updatedafter	Limit to metadata updated after specified date; updates are data center specific.

These parameters should be submitted as key=value pairs using the HTTP GET method and may not be specified more than once; if a parameter is submitted multiple times the result is undefined.

The *level* parameter controls the amount of detail included in the returned FDSN StationXML results with the following hierarchy: network, station, channel and response. For example, if the station level is requested the Station elements will be included in the results but not Channel or Response related information.

Responses for the *query* method:

The results of a successful request should be returned as FDSN StationXML using MIME type **application/xml**.

Behavior for the **version** method

The service shall return the implementation version as a simple text string using the MIME type **text/plain**. Any parameters submitted with the method will be ignored.

Behavior for the **application.wadl** method

The service shall return a WADL conformant description of the interface using the MIME type **application/xml**. Any parameters submitted with the method will be ignored.

Behavior for the **error** conditions

All errors returned to the client should use the type and pattern described in the Common Service Responses section.

fdsnws-event version 1.0

The following methods shall be supported by the service:

- query** – to submit a data request
- catalogs** – to submit a request for available catalogs
- contributors** – to submit a request for available contributors
- version** – to request the full service version number
- application.wadl** – to request a WADL for the interface

Requests using the *query* method:

The service shall accept requests formulated using the selection parameters identified in Table 1: *simple-time* and *geographic constraints* and the parameters listed that are specific for **fdsnws-event**. Table 4 contains parameter descriptions for this service.

Table 4. Parameter descriptions for the **fdsnws-event** service

Parameter	Description
starttime	Limit to events on or after the specified start time.
endtime	Limit to events on or before the specified end time.
minlatitude	Limit to events with a latitude larger than the specified minimum.
maxlatitude	Limit to events with a latitude smaller than the specified maximum.
minlongitude	Limit to events with a longitude larger than the specified minimum.
maxlongitude	Limit to events with a longitude smaller than the specified maximum.
latitude	Specify the latitude to be used for a radius search.
longitude	Specify the longitude to be used for a radius search.
minradius	Limit to events within the specified minimum number of degrees from the geographic point defined by the latitude and longitude parameters.
maxradius	Limit to events within the specified maximum number of degrees from the geographic point defined by the latitude and longitude parameters.
mindepth	Limit to events with depth more than the specified minimum.
maxdepth	Limit to events with depth less than the specified maximum.
minmagnitude	Limit to events with a magnitude larger than the specified minimum.
maxmagnitude	Limit to events with a magnitude smaller than the specified maximum.
magnitudetype	Specify a magnitude type to use for testing the minimum and maximum limits.
includeallorigins	Specify if all origins for the event should be included, default is data center dependent but is suggested to be the preferred origin only.
includeallmagnitudes	Specify if all magnitudes for the event should be included, default is data center dependent but is suggested to be the preferred magnitude only.
includearrivals	Specify if phase arrivals should be included.
eventid	Select a specific event by ID; event identifiers are data center specific.
limit	Limit the results to the specified number of events.
offset	Return results starting at the event count specified, starting at 1.
orderby	Order the result by time or magnitude with the following possibilities: <i>time</i> : order by origin descending time <i>time-asc</i> : order by origin ascending time <i>magnitude</i> : order by descending magnitude <i>magnitude-asc</i> : order by ascending magnitude
catalog	Limit to events from a specified catalog
contributor	Limit to events contributed by a specified contributor.
updatedafter	Limit to events updated after the specified time.

These parameters should be submitted as key=value pairs using the HTTP GET method and may not be specified more than once; if a parameter is submitted multiple times the result is undefined.

Responses for the *query* method:

The results of a successful request should be returned as QuakeML 1.2 using MIME type **application/xml**.

Behavior for the *catalogs* method

The service shall return the list of available catalogs in the simple XML schema illustrated below using the MIME type **application/xml**. Any parameters submitted with the method will be ignored.

```
<Catalogs>
  <Catalog>EMSC</Catalog>
  <Catalog>GCMT</Catalog>
  <Catalog>NEIC PDE</Catalog>
  <Catalog>ISC</Catalog>
</Catalogs>
```

Behavior for the *contributors* method

The service shall return the list of available contributors in the simple XML schema illustrated below using the MIME type **application/xml**. Any parameters submitted with the method will be ignored.

```
<Contributors>
  <Contributor>NEIC</Contributor>
  <Contributor>EMSC</Contributor>
  <Contributor>ISC</Contributor>
</Contributors>
```

Behavior for the *version* method

The service shall return the implementation version as a simple text string using the MIME type **text/plain**. Any parameters submitted with the method will be ignored.

Behavior for the *application.wadl* method

The service shall return a WADL conformant description of the interface using the MIME type **application/xml**. Any parameters submitted with the method will be ignored.

Behavior for the *error* conditions

All errors returned to the client should use the type and pattern described in the Common Service Responses section.

References

SEED data format - http://www.iris.edu/manuals/SEEDManual_V2.4.pdf

QuakeML - <https://quake.ethz.ch/quakeml/>

FDSNStationXML

To be based on StationXML - <http://www.data.scec.org/station/xml.html>

This specification document is written with the assumption that data availability, similar to that produced by the IRIS DMC's ws-availability service will be incorporated into the FDSN StationXML schema.

Web Application Description Language (WADL) - <http://www.w3.org/Submission/wadl/>

Changes

2012/07/12 – Initial specification, release candidate 1

2012/07/18 – Remove <Total> elements from **catalog** and **contributor** XML responses for fdsn-event. Designate the fdsn-station specification as provisional until the FDSN StationXML schema is defined.