FDSN Web Service Specifications

Version 1.2 (2018/09/10)

Defining:
fdsnws-dataselect v1.1
fdsnws-station v1.1
fdsnws-event v1.2

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 and alternate formats

event – returns parametric data for events in QuakeML and alternate formats

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 a minimum 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

Minimum functionality

Implementations of these service interfaces should support all methods specified as required. Additionally, interfaces should conform to the calling patterns and expected results identified in this document to be considered conformant with the specification. Authentication methods are an exception, and only need to be supported by data centers that wish to require authentication for access to specific data sets.

Each service definition includes required and optional parameters; an implementation must support the required parameters to be considered conformant. The optional parameters supported by any given implementation should be specified in the WADL returned by the service.

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) by default. Alternatively, the optional *nodata* parameter can be used to request that **HTTP status 404** (Not Found) be returned instead. A 404 status helps clarify situations where a 204 may be confusing, such as a web browser user, by including a message in the response. Clients should be aware that the 404 status code may be returned for error conditions such as problems with the web application or application container. Due to the multiple meanings for a 404 status, the 204 status is preferred for programmatic (non-browser) access.

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>
```

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.

WADL conventions

The WADL documents returned by FDSN services should follow these guidelines:

- All public parameters supported by an interface shall be documented in the WADL.
- Parameters supported by the interface shall be documented using the long version of the parameter name and not the abbreviated version.
- Parameters not included in the specification (data center-specific extensions) shall be documented with a type and a short description.

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
404	Alternate to 204 (set via the 'nodata' parameter), normally used for results returned to a web browser.
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. 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 2 (a& b). 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-station** 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 2a. Common parameter specifications for FDSN web services. Appropriate services for each parameter group are identified in italics.

Group	Subgroup	Parameter	Abbreviation	Default	Minimum	Maximum	Type	Units
time constraints								
	simple-time	***fdsnws-dataselect, fdnws-station, fdsnws-event***						
		starttime	start	[Any]	[Any valid time]		time	UTC
		endtime	end	[Any]	[Any valid time]		time	UTC
	window-time	***fdsnws-station***						
		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								
		fdsnws-dataselect, fdsnws-station						
		network	net	[Any]	[Valid ASCII or * or ?]		string	N/A
		station	sta	[Any]	[Valid ASCII or * or ?]		string	N/A
		location	loc	[Any]	[Valid ASCII or * or ?]		string	N/A
		channel	cha	[Any]	[Valid ASCII or * or ?]		string	N/A
geographic constraints								
	area-rectangle	***fdsnws-station, fdsnws-event***						
		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	***fdsnws-station, fdsnws-event***						
		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
output control								
	fdsnws-dataselect, fdsnws-station, fdsnws-event							
		format		*xml* or *miniseed*	*xml*, *miniseed* or *text*		string	N/A
		nodata		204	*204* or *404*		string	N/A

Table 2b. Service-specific parameter specifications for FDSN web services.

Group	Subgroup	Parameter	Abbreviation	Default	Minimum	Maximum	Type	Units
service-specific constraints								
	fdsnws-station							
		level		station	*network, station, channel* or *response*		string	N/A
		includerestricted		TRUE	N/A		boolean	N/A
		includeavailability		FALSE	N/A		boolean	N/A
		updatedafter		[Any]	[Any valid time]		time	UTC
		matchtimeseries		FALSE	N/A		boolean	N/A
	fdsnws-event							
		mindepth		[Any]	[Any]	[Any]	float	kilometers
		maxdepth		[Any]	[Any]	[Any]	float	kilometers
		minmagnitude	minmag	[Any]	[Any]	[Any]	float	Defined by
		maxmagnitude	maxmag	[Any]	[Any]	[Any]	float	mag type
		magnitudetype	magtype	[Any]	N/A		string	N/A
		eventtype		[Any]	N/A		string	N/A
		includeallorigins		FALSE	N/A		boolean	N/A
		includeallmagintudes		FALSE	N/A		boolean	N/A
		includearrivals		FALSE	N/A		boolean	N/A
		eventid		[Any]	N/A		string	N/A
		limit		[Any]	1	[Positive #]	integer	N/A
		offset		1	1	[Positive #]	integer	N/A
		orderby		time	*time, time-asc, magnitude or magnitude-asc*		string	N/A
		contributor		[Any]	N/A		string	N/A
		catalog		[Any]	N/A		string	N/A
		updatedafter		[Any]	[Any valid time]		time	UTC
	fdsnws-dataselect							
		quality		B	Quality: D, R, Q, M or B		string	N/A
		minimumlength		0.0	0.0	[Positive #]	float	N/A
		longestonly		FALSE	N/A		boolean	N/A

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-DDT**HH: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-DDT**HH:MM:SS** (implies a time of **YYYY-MM-DD**T**HH**:**MM**:**SS**.000000)

where the sub-seconds are interpreted to be zeros and

YYYY-MM-DD (implies a time of **YYYY-MM-DD**T00:00:00.000000)

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 selection shall include any metadata epochs intersecting with the specified time range.

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 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 1.1

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 2: *simple-time*, *channel constraints* and parameters specific to **fdsnws-dataselect**. Table 3 contains parameter descriptions for this service.

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

Parameter	Support	Description
starttime	Required	Limit results to time series samples on or after the specified start time
endtime	Required	Limit results to time series samples on or before the specified end time
network	Required	Select one or more network codes. Can be SEED network codes or data center defined codes. Multiple codes are comma-separated.
station	Required	Select one or more SEED station codes. Multiple codes are comma-separated.
location	Required	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	Required	Select one or more SEED channel codes. Multiple codes are comma-separated.
quality	Optional	Select a specific SEED quality indicator, handling is data center dependent.
minimumlength	Optional	Limit results to continuous data segments of a minimum length specified in seconds,
longestonly	Optional	Limit results to the longest continuous segment per channel.
format	Optional	Specify format of result, the default value is 'miniseed'. If this parameter is not specified the service must return miniSEED.
nodata	Optional	Select status code for "no data", either '204' (default) or '404'.

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 shall be returned as a stream of miniSEED by default, 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. A data center may choose to return the results in alternate formats as specified by the format parameter.

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 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. The WADL shall describe all parameters supported by the interface and is primarily used to document which optional parameters are supported.

Behavior for the **error** conditions

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

Examples

Requesting all data channels from IU station COLA for the first 12 hours of year 2012 using a GET request:

http://service.iris.edu/fdsnws/dataselect/1/query?network=IU&station=COLA&starttime=2012-01-01T00:00:00&endtime=2012-01-01T12:00:00

Requesting all LHE, LHN, LHZ data channels from IU station COLA for the first 12 hours of year 2012 using a POST request:

http://service.iris.edu/fdsnws/dataselect/1/query

POST message body:

```
quality=M
IU COLA 00 LHE 2012-01-01T00:00:00 2012-01-01T12:00:00
IU COLA 00 LHN 2012-01-01T00:00:00 2012-01-01T12:00:00
IU COLA 00 LHZ 2012-01-01T00:00:00 2012-01-01T12:00:00
```

fdsnws-station version 1.1

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 2: *time constraints, channel constraints, geographic constraints* and the parameters listed that are specific for **fdsnws-station**. Table 4 contains parameter descriptions for this service.

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

Parameter	Support	Description
starttime	Required	Limit to metadata epochs starting on or after the specified start time.
endtime	Required	Limit to metadata epochs ending on or before the specified end time.
startbefore	Optional	Limit to metadata epochs starting before specified time.
startafter	Optional	Limit to metadata epochs starting after specified time.
endbefore	Optional	Limit to metadata epochs ending before specified time.
endafter	Optional	Limit to metadata epochs ending after specified time.
network	Required	Select one or more network codes. Can be SEED network codes or data center defined codes. Multiple codes are comma-separated.
station	Required	Select one or more SEED station codes. Multiple codes are comma-separated.
location	Required	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	Required	Select one or more SEED channel codes. Multiple codes are comma-separated.
minlatitude	Required	Limit to stations with a latitude larger than or equal to the specified minimum.
maxlatitude	Required	Limit to stations with a latitude smaller than or equal to the specified maximum.
minlongitude	Required	Limit to stations with a longitude larger than or equal to the specified minimum.
maxlongitude	Required	Limit to stations with a longitude smaller than or equal to the specified maximum.
latitude	Optional	Specify the latitude to be used for a radius search.
longitude	Optional	Specify the longitude to the used for a radius search.
minradius	Optional	Limit results to stations within the specified minimum number of degrees from the geographic point defined by the latitude and longitude parameters.
maxradius	Optional	Limit results to stations within the specified maximum number of degrees from the geographic point defined by the latitude and longitude parameters.
level	Required	Specify the level of detail for the results.
includerestricted	Optional	Specify if results should include information for restricted stations.
includeavailability	Optional	Specify if results should include information about time series data availability.
updatedafter	Optional *	Limit to metadata updated after specified time; updates are data center specific. * While this option is not required it is highly recommended due to usefulness.
matchtimeseries	Optional	Limit to metadata where selection criteria matches time series data availability.
format	Optional	Specify format of result, either ‘xml’ (default) or ‘text’ (defined below). If this parameter is not specified the service must return StationXML.
nodata	Optional	Select status code for “no data”, either ‘204’ (default) or ‘404’.

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 core 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 parameters for *network*, *station*, *location*, *channel*, *starttime* and *endtime* may be repeated as many times as necessary, all other parameters should be specified as key=value pairs on separate lines following this pattern:

```
parameter1=value
parameter2=value
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/station/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.

Controlling the level of detail

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 shall be returned as FDSN StationXML using MIME type **application/xml** by default or when the *format* parameter value is 'xml'. If the *format* parameter value is 'text' the results shall be returned in a simple ASCII format for *level* = network, station or channel using MIME type **text/plain** as defined below. Requests for text output at the response level should generate an error.

Behavior for the ***version*** method

The service shall return the 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. The WADL shall describe all parameters supported by the interface and is primarily used to document which optional parameters are supported.

Behavior for the **error** conditions

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

Behavior for the ***matchtimeseries*** parameter

When *matchtimeseries* is TRUE: in addition to matching metadata based on selection criteria, the results will be further limited to metadata for which the selection criteria (e.g. *starttime, endtime, network, station, location, channel*) also match the existence of available time series data.

Text output formats for network, station and channel levels

These simple text output formats contain one entry per line with common fields separated by vertical bar characters ("|": ASCII decimal 124). Field entries cannot contain vertical bar characters. Lines beginning with a hash character ("#": ASCII decimal 35) should be considered comment lines.

level=network

```
Network|Description|StartTime|EndTime|TotalStations
Network|Description|StartTime|EndTime|TotalStations
...
```

level=station

```
Network|Station|Latitude|Longitude|Elevation|SiteName|StartTime|EndTime
Network|Station|Latitude|Longitude|Elevation|SiteName|StartTime|EndTime
...
```

level=channel

```
Network|Station|Location|Channel|Latitude|Longitude|Elevation|Depth|Azimuth|Dip|Senso
rDescription|Scale|ScaleFrequency|ScaleUnits|SampleRate|StartTime|EndTime
Network|Station|Location|Channel|Latitude|Longitude|Elevation|Depth|Azimuth|Dip|Senso
rDescription|Scale|ScaleFrequency|ScaleUnits|SampleRate|StartTime|EndTime
...
```

The description for each column is, in most cases, directly equivalent to the like named element or attribute in StationXML and SEED, column values should follow the conventions used in those formats.

In these text formats the meaning of the StartTime and EndTime values change depending on the level of the output. For channel-level output the time values refer to channel operating times, for station-level output the time values refer to station operating times and for network-level output the time values refer to network operating times. Similarly, the meaning of the Latitude and Longitude values change depending on the level, they are either station coordinates or channel coordinates.

In the network level output, the TotalStations value is the number of stations associated with the given network entry.

In the channel level output, the Scale value is the total sensitivity, the ScaleFrequency is the frequency at which the Scale is valid and the ScaleUnits are the units of data after the Scale value has been applied. These values are equivalent to the values in the StationXML <InstrumentSensitivity> structure or stage 0 of SEED headers.

In the channel level output, the SensorDescription field is equivalent to the StationXML element Network::Station::Channel::Sensor::Type, which is commonly mapped to description identified in field 6 of SEED Blockette 52.

Examples

Requesting metadata for all channels from IU station COLA with no time limitations:

http://service.iris.edu/fdsnws/station/1/query?network=IU&station=COLA&level=channel

Requesting metadata for all BH* channels between latitude -10 and -45 and longitude between 112 and 155 (e.g. Australia):

http://service.iris.edu/fdsnws/station/1/query?channel=BH&maxlatitude=-10&minlatitude=-45&minlongitude=122&maxlongitude=155&level=channel*

Requesting metadata in simple text format at the network level:

http://service.iris.edu/fdsnws/station/1/query?network=II,IU&format=text&level=network

```
#Network | Description | StartTime | EndTime | TotalStations
II|Global Seismograph Network (GSN - IRIS/IDA)|1986-01-01T00:00:00|2500-12-12T23:59:59|50
IU|Global Seismograph Network (GSN - IRIS/USGS)|1988-01-01T00:00:00|2500-12-12T23:59:59|255
```

Requesting metadata in simple text format at the station level:

http://service.iris.edu/fdsnws/station/1/query?network=IU&format=text&level=station

```
#Network | Station | Latitude | Longitude | Elevation | SiteName | StartTime | EndTime
IU|ANMO|34.9459|-106.4572|1850.0|Albuquerque, New Mexico, USA|1989-08-29T00:00:00|1995-07-14T00:00:00
IU|ANMO|34.9459|-106.4572|1850.0|Albuquerque, New Mexico, USA|1995-07-14T00:00:00|2000-10-19T16:00:00
IU|ANMO|34.9502|-106.4602|1839.0|Albuquerque, New Mexico, USA|2000-10-19T16:00:00|2002-11-19T21:07:00
…
```

Requesting metadata in simple text format at the channel level:

http://service.iris.edu/fdsnws/station/1/query?station=COLA&format=text&level=channel

```
#Network | Station | Location | Channel | Latitude | Longitude | Elevation | Depth | Azimuth |
Dip | SensorDescription | Scale | ScaleFreq | ScaleUnits | SampleRate | StartTime | EndTime
IU|COLA|20|HNE|64.873599|-147.8616|200.0|0.0|90.0|0.0|Kinemetrics FBA-23 Low-Gain
Sensor|53687.1|1.0|M/S**2|80.0|2005-09-28T22:00:00|2009-07-08T22:00:00
IU|COLA|20|HNN|64.873599|-147.8616|200.0|0.0|0.0|0.0|Kinemetrics FBA-23 Low-Gain
Sensor|53687.1|1.0|M/S**2|80.0|2005-09-28T22:00:00|2009-07-08T22:00:00
IU|COLA|20|HNZ|64.873599|-147.8616|200.0|0.0|0.0|-90.0|Kinemetrics FBA-23 Low-Gain
Sensor|53687.1|1.0|M/S**2|80.0|2005-09-28T22:00:00|2009-07-08T22:00:00
IU|COLA|00|BH1|64.873599|-147.8616|80.0|120.0|111.0|0.0|Geotech KS-54000 Borehole
Seismometer|8.04479E8|0.02|M/S|20.0|2005-09-27T00:00:00|2009-07-08T22:00:00
IU|COLA|00|BH2|64.873599|-147.8616|80.0|120.0|201.0|0.0|Geotech KS-54000 Borehole
Seismometer|8.1814E8|0.02|M/S|20.0|2005-09-27T00:00:00|2009-07-08T22:00:00
IU|COLA|00|BHZ|64.873599|-147.8616|80.0|120.0|0.0|-90.0|Geotech KS-54000 Borehole
Seismometer|8.44329E8|0.02|M/S|20.0|2005-09-27T00:00:00|2009-07-08T22:00:00
…
```

Requesting two time windows of data for two IU stations with different channel selections using a POST request:

http://service.iris.edu/fdsnws/station/1/query

POST message body:

```
level=channel
includerestricted=FALSE
IU COLA 00 LH? 2012-01-01T00:00:00 2012-01-01T12:00:00
IU ANMO 10 BH? 2013-07-01T00:00:00 2013-02-07T12:00:00
```

fdsnws-event version 1.2

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 2: *simple-time* and *geographic constraints* and the parameters listed that are specific for **fdsnws-event**. Table 5 contains parameter descriptions for this service.

Table 5. Parameter descriptions for the **fdsnws-event** service.

Parameter	Support	Description
starttime	Required	Limit to events on or after the specified start time.
endtime	Required	Limit to events on or before the specified end time.
minlatitude	Required	Limit to events with a latitude larger than or equal to the specified minimum.
maxlatitude	Required	Limit to events with a latitude smaller than or equal to the specified maximum.
minlongitude	Required	Limit to events with a longitude larger than or equal to the specified minimum.
maxlongitude	Required	Limit to events with a longitude smaller than or equal to the specified maximum.
latitude	Optional	Specify the latitude to be used for a radius search.
longitude	Optional	Specify the longitude to the used for a radius search.
minradius	Optional	Limit to events within the specified minimum number of degrees from the geographic point defined by the latitude and longitude parameters.
maxradius	Optional	Limit to events within the specified maximum number of degrees from the geographic point defined by the latitude and longitude parameters.
mindepth	Required	Limit to events with depth more than the specified minimum.
maxdepth	Required	Limit to events with depth less than the specified maximum.
minmagnitude	Required	Limit to events with a magnitude larger than the specified minimum.
maxmagnitude	Required	Limit to events with a magnitude smaller than the specified maximum.
magnitudetype	Optional	Specify a magnitude type to use for testing the minimum and maximum limits.
eventtype	Optional	Limit to events with a specified eventType. The parameter value can be a single item, a comma-separated list of items, or the asterisk wildcard. Allowed values are from the QuakeML 1.2 EventType enumeration.
includeallorigins	Optional	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	Optional	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	Optional	Specify if phase arrivals should be included.
eventid	Optional	Select a specific event by ID; event identifiers are data center specific.
limit	Optional	Limit the results to the specified number of events.
offset	Optional	Return results starting at the event count specified, starting at 1.
orderby	Required	Order the result by time or magnitude with the following possibilities: *time*: order by origin descending time *time-asc*: order by origin ascending time *magnitude*: order by descending magnitude *magnitude-asc*: order by ascending magnitude
catalog	Optional	Limit to events from a specified catalog
contributor	Optional	Limit to events contributed by a specified contributor.
updatedafter	Optional *	Limit to events updated after the specified time. * While this option is not required it is highly recommended due to usefulness.
format	Optional	Specify format of result, either 'xml' (default) or 'text' (defined below). If this parameter is not specified the service must return QuakeML.
nodata	Optional	Select status code for "no data", either '204' (default) or '404'.

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 shall be returned as QuakeML 1.2 using MIME type **application/xml** by default or when the *format* parameter value is 'xml'. If the *format* parameter value is 'text' the results shall be returned in a simple ASCII format using MIME type **text/plain** as defined below.

Behavior for the ***catalogs*** method

The service shall return the 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 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 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. The WADL shall describe all parameters supported by the interface and is primarily used to document which optional parameters are supported.

Behavior for the **error** conditions

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

Behavior for the **eventtype** parameter

The *eventtype* parameter can hold an item of the EventType enumeration as defined in the QuakeML 1.2 standard, or a comma-separated list of items. Alternatively, it can hold the asterisk (*) wildcard, which indicates that no selection on eventType should be performed. This means that events with unset eventType are included in the result. The asterisk wildcard is the default value. If *eventtype* is not the asterisk wildcard, the result list is filtered such that it contains only events with event types as specified in the *eventtype* query parameter. Allowed values are (separated by commas, specified in lowercase, but mixed case is allowed):

> not existing, not reported, earthquake, anthropogenic event, collapse, cavity collapse, mine collapse, building collapse, explosion, accidental explosion, chemical explosion, controlled explosion, experimental explosion, industrial explosion, mining explosion, quarry blast, road cut, blasting levee, nuclear explosion, induced or triggered event, rock burst, reservoir loading, fluid injection, fluid extraction, crash, plane crash, train crash, boat crash, other event, atmospheric event, sonic boom, sonic blast, acoustic noise, thunder, avalanche, snow avalanche, debris avalanche, hydroacoustic event, ice quake, slide, landslide, rockslide, meteorite, volcanic eruption

EventType values as defined in QuakeML 1.2 are all-lowercase with words separated by single blank spaces. However, the query parameter is case insensitive, i.e., its value has to be converted to lowercase before filtering.

Text output format

This simple text output format contains one event per line with common fields separated by vertical bar characters (ASCII decimal 124). Field entries cannot contain vertical bar characters. Lines beginning with a hash character ("#": ASCII decimal 35) should be considered comment lines.

```
EventID|Time|Latitude|Longitude|Depth/km|Author|Catalog|Contributor|ContributorID|MagType|Magnitude|MagAuthor|EventLocationName|EventType
EventID|Time|Latitude|Longitude|Depth/km|Author|Catalog|Contributor|ContributorID|MagType|Magnitude|MagAuthor|EventLocationName|EventType
...
```

As multiple origins, multiple magnitudes and arrivals cannot be represented in the text format, service implementations should ignore the 'includeallorigins', 'includeallmagnitudes' and 'includearrivals' parameters when the text output has been selected.

The description for each column of the text format is as follows:

EventID	Event ID assigned by the data center, as to be used with *eventid* parameter
Time	Origin time of the event in UTC
Latitude	Event latitude in degrees
Longitude	Event longitude in degrees
Depth/km	Event depth in kilometers
Author	Identifier of event origin author
Catalog	Identifier of source catalog
Contributor	Identifier of event information contributor
ContributorID	Event ID as reported by the contributor
MagType	Magnitude scale identifier (e.g. mb, ML, Mw)

Magnitude	Magnitude value
MagAuthor	Identifier of magnitude author
EventLocationName	Geographic description of event location (e.g. Flinn-Engdahl region name)
EventType	Event type classification string as of QuakeML 1.2 EventType enumeration

Example

Requesting origins for all events above magnitude 6 in 2012:

https://DataCenter/fdsnws/event/1/query?minmag=6&starttime=2012-01-01T00:00:00&endtime=2013-01-01T00:00:00

Requesting origins above magnitude 7 in text format:

https://DataCenter/fdsnws/event/1/query?minmag=7&starttime=2012-09-01T00:00:00&endtime=2012-12-01T00:00:00&format=text

```
#EventID|Time|Latitude|Longitude|Depth/km|Author|Catalog|Contributor|ContributorID|MagType|Magnitude|MagAu
thor|EventLocationName|EventType
usp000jv5f|2012-11-07T16:35:46.930|13.988|-91.895|24|us|us|us|usp000jv5f|mww|7.4|us|offshore
Guatemala|earthquake
usp000juhz|2012-10-28T03:04:08.820|52.788|-132.101|14|us|us|us|usp000juhz|mww|7.8|us|Haida Gwaii,
Canada|earthquake
usp000jta1|2012-09-30T16:31:35.970|1.929|-76.362|170|us|us|us|usp000jta1|mww|7.3|us|Colombia|earthquake
usp000jrsw|2012-09-05T14:42:07.800|10.085|-85.315|35|us|us|us|usp000jrsw|mww|7.6|us|Costa Rica|earthquake
```

References

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

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

FDSN StationXML - http://www.fdsn.org/xml/station/

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

Changes

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

2012/07/18

1) Remove <Total> elements from **catalog** and **contributor** XML responses for **fdsnws-event**.
2) Designate the **fdsnws-station** specification as provisional until the FDSN StationXML schema is defined.

2012/11/20

1) Include examples request URI's for each service.
2) Remove provisional status on fdsnws-station.

2013/04/24 –Version 1.0 final

1) Replace parameter name *preferredonly* with *includeallorigins* in the fdsnws-event section of Table 1 and make the default value FALSE. This now matches other references to the parameter names and was the original intention.
2) Change the example of exclusive parameter sub-groups to refer to **fdsnws-station** as a more appropriate illustration (page 4).
3) Add link to FDSN StationXML specification in references and remove provisional language.
4) Remove the DRAFT designation.

2013/09/18 – Integrate proposed version 1.1 changes

1) Add designation of required versus optional parameters for each service interface.
2) Allow **fdsnws-station** to accept arbitrary selections via the POST method. Support for this method is required for all 1.1 implementations of fdsnws-station.
3) Add optional *format* parameter to each service definition. The default formats of miniSEED, StationXML and QuakeML remain the same for each interface. For station and event, document an alternate, simple text format to return when *format=text* including examples.
4) Add optional *nodata* parameter to each service definition that accepts values of '204' (default) and '404'. This parameter controls the HTTP status code to return when the request was successful but matched no data.
5) Add optional *matchtimeseries* parameter to the **fdsnws-station** definition. This parameter is used to limit results to selections that match time series data availability.
6) Clarify definition of *starttime* and *endtime* parameters to explicitly include metadata epochs intersection with the specified time range.

7) Clarify definition of minimum and maximum latitude and longitude to be inclusive by adding "or equal to" to their description.

2013/10/04 – 1.1 release candidate

1) Add description and clarification for the columns of the text formats for **fdsnws-station** and **fdsnws-event**.
2) Avoid ambiguity of required parameters by re-naming the appropriate column "Support" and changing the values to "Required" and "Optional".
3) Add another example request to the **fdsnws-event** interface illustrating a query for text data and a portion of the response that might be expected.

2013/10/15 – 1.1 release candidate 2

1) Add section describing WADL conventions.
2) Clarify that when the *format* parameter is not specified the default format must be returned.
3) Clarify that an HTTP 413 status shall be returned when either the request entity itself is too large or the resulting data set would be too large, the latter being an extension of the standard meaning.

2013/10/25 – Version 1.1 final

1) Clarify limitations of text output format: vertical bars cannot be included in fields, multiple origins & magnitudes cannot be included, phase arrivals cannot be included.
2) Document intended behavior for the matchtimeseries parameter of fdsnws-station.
3) Change minimum value for the *limit* parameter to **fdsnws-event** to 1, previously 0.
4) Reformat Table 1 so complete table is visible.
5) Re-number tables, Table 1 was previously used for two different tables.

2015/06/19 – Version 1.1b (clarifying edit, internal version change only)

1) Add *SiteName* column to station-level text template (page 12) for **fdsnws-station** as was the original intention and to match the example and predominant implementations.

2018/09/10 - Version 1.2

1) Update **fdsnws-event** specification to version 1.2, with the following changes:
    a) addition of the *eventtype* parameter
    b) addition of the EventType column in the text output format (and example)
2) Split Table 2 into Table 2a and 2b to allow additions and increase legibility.