

### Common

|     |                    |   |
|-----|--------------------|---|
| GET | /status            | Returns the Druid version, loaded extensions, memory used, total memory and other useful information about the process. |
| GET | /status/health     | Always returns a boolean "true" value with a 200 OK response, useful for automated health checks.                       |
| GET | /status/properties | Returns the current configuration properties of the process.  |

These endpoints are supported by all processes.

### Master Server

#### Coordinator - Leadership

|     |                                |   |
|-----|--------------------------------|---|
| GET | /druid/coordinator/v1/leader   | Returns the current leader Coordinator of the cluster.          |
| GET | /druid/coordinator/v1/isLeader | Returns a JSON object with field "leader", either true or false |

### Master Server

#### Segment Loading

|     |   |   |
|-----|---|---|
| GET | /druid/coordinator/v1/loadstatus        | Returns the percentage of segments actually loaded in the cluster                     |
| GET | /druid/coordinator/v1/loadstatus?simple | Returns the number of segments left to load in each tier                              |
| GET | /druid/coordinator/v1/loadqueue         | Returns the ids of segments to load and drop for each Historical process.             |
| GET | /druid/coordinator/v1/loadqueue?simple  | Returns the number of segments to load and drop                                       |
| GET | /druid/coordinator/v1/loadqueue?full    | Returns the serialized JSON of segments to load and drop for each Historical process. |

### Master Server

#### Metadata info

|     |   |  |
|-----|---|--|
| GET | /druid/coordinator/v1/metadata/datasources                                | Returns a list of the names of enabled datasources in the cluster.   |
| GET | /druid/coordinator/v1/metadata/datasources?includeDisabled                | Returns a list of the names of enabled and disabled datasources in the cluster.                                      |
| GET | /druid/coordinator/v1/metadata/datasources?full                           | Returns a list of all enabled datasources with all metadata about those datasources as stored in the metadata store. |
| GET | /druid/coordinator/v1/metadata/datasources/{dataSourceName}               | Returns full metadata for a datasource as stored in the metadata store.  |
| GET | /druid/coordinator/v1/metadata/datasources/{dataSourceName}/segments      | Returns a list of all segments for a datasource as stored in the metadata store.                                     |
| GET | /druid/coordinator/v1/metadata/datasources/{dataSourceName}/segments?full | Returns a list of all segments for a datasource with the full segment metadata as stored in the metadata store.      |



By [kinger1999](#)

Published 19th June, 2019.  
Last updated 19th June, 2019.  
Page 1 of 8.

Sponsored by [Readable.com](#)  
Measure your website readability!  
<https://readable.com>

### Master Server (cont)

|      |  |  |
|------|--|--|
| GET  | /druid/coordinator/v1/metadata/datasources/{dataSourceName}/segments/{segmentId} | Returns full segment metadata for a specific segment as stored in the metadata store.                                    |
| POST | /druid/coordinator/v1/metadata/datasources/{dataSourceName}/segments             | Returns a list of all segments, overlapping with any of given intervals  |
| POST | /druid/coordinator/v1/metadata/datasources/{dataSourceName}/segments?full        | Returns a list of all segments, overlapping with any of given intervals, for a datasource with the full segment metadata |

### Master Server

#### Datasources

|     |  |  |
|-----|--|--|
| GET | /druid/coordinator/v1/datasources  | Returns a list of datasource names found in the cluster.   |
| GET | /druid/coordinator/v1/datasources?simple                                       | Returns a list of JSON objects containing the name and properties of datasources found in the cluster.   |
| GET | /druid/coordinator/v1/datasources?full   | Returns a list of datasource names found in the cluster with all metadata about those datasources.   |
| GET | /druid/coordinator/v1/datasources/{dataSourceName}                             | Returns a JSON object containing the name and properties of a datasource   |
| GET | /druid/coordinator/v1/datasources/{dataSourceName}?full                        | Returns full metadata for a datasource .   |
| GET | /druid/coordinator/v1/datasources/{dataSourceName}/intervals                   | Returns full metadata for a datasource .   |
| GET | /druid/coordinator/v1/datasources/{dataSourceName}/intervals                   | Returns a set of segment intervals.  |
| GET | /druid/coordinator/v1/datasources/{dataSourceName}/intervals?simple            | Returns a map of an interval to a JSON object containing the total byte size of segments and number of segments for that interval.   |
| GET | /druid/coordinator/v1/datasources/{dataSourceName}/intervals?full              | Returns a map of an interval to a map of segment metadata to a set of server names that contain the segment for that interval.   |
| GET | /druid/coordinator/v1/datasources/{dataSourceName}/intervals/{interval}        | Returns a set of segment ids for an interval.  |
| GET | /druid/coordinator/v1/datasources/{dataSourceName}/intervals/{interval}?simple | Returns a map of segment intervals contained within the specified interval to a JSON object  |
| GET | /druid/coordinator/v1/datasources/{dataSourceName}/intervals/{interval}?full   | Returns a map of segment intervals contained within the specified interval to a map of segment metadata to a set of server names that contain the segment for an interval. |



By [kinger1999](#)

Published 19th June, 2019.

Last updated 19th June, 2019.

Page 2 of 8.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

### Master Server (cont)

|        |  |   |
|--------|--|---|
| GET    | /druid/coordinator/v1/datasources/{dataSourceName}/intervals/{interval}/serverview | Returns a map of segment intervals contained within the specified interval to information about the servers that contain the segment for an interval. |
| GET    | /druid/coordinator/v1/datasources/{dataSourceName}/segments                        | Returns a list of all segments for a datasource in the cluster.   |
| GET    | /druid/coordinator/v1/datasources/{dataSourceName}/segments?full                   | Returns a list of all segments for a datasource in the cluster with the full segment metadata.  |
| GET    | /druid/coordinator/v1/datasources/{dataSourceName}/segments/{segmentId}            | Returns full segment metadata for a specific segment in the cluster.  |
| GET    | /druid/coordinator/v1/datasources/{dataSourceName}/tiers                           | Return the tiers that a datasource exists in.   |
| POST   | /druid/coordinator/v1/datasources/{dataSourceName}                                 | Enables all segments of datasource which are not overshadowed by others.  |
| POST   | /druid/coordinator/v1/datasources/{dataSourceName}/segments/{segmentId}            | Enables a segment of a datasource.  |
| DELETE | /druid/coordinator/v1/datasources/{dataSourceName}                                 | Disables a datasource.  |
| DELETE | /druid/coordinator/v1/datasources/{dataSourceName}/intervals/{interval}            | Runs a Kill task for a given interval and datasource.   |
| DELETE | /druid/coordinator/v1/datasources/{dataSourceName}/segments/{segmentId}            | Disables a segment.   |

Note that all interval URL parameters are ISO 8601 strings delimited by a \_ instead of a / (e.g., 2016-06-27\_2016-06-28).

### Master Server

#### Retention Rules

|     |   |   |
|-----|---|---|
| GET | Retention Rules   | Returns all rules as JSON objects for all datasources in the cluster including the default datasource.  |
| GET | /druid/coordinator/v1/rules/{dataSourceName}            | Returns all rules for a specified datasource.   |
| GET | /druid/coordinator/v1/rules/{dataSourceName}?full       | Returns all rules for a specified datasource and includes default datasource.   |
| GET | /druid/coordinator/v1/rules/history?interval=<interval> | Returns audit history of rules for all datasources  |
|     |   | default value of interval can be specified by setting druid.audit.manager.auditHistoryMillis (1 week if not configured) in Coordinator runtime.properties |
| GET | /druid/coordinator/v1/rules/history?count=<n>           | Returns last entries of audit history of rules for all datasources.   |



By [kinger1999](#)

Published 19th June, 2019.

Last updated 19th June, 2019.

Page 3 of 8.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

### Master Server (cont)

|      |   |  |
|------|---|--|
| GET  | /druid/coordinator/v1/rules/{dataSource-Name}/history?interval=<interval> | Returns audit history of rules for a specified datasource<br><br>default value of interval can be specified by setting druid.audit.manager.auditHistoryMillis (1 week if not configured) in Coordinator runtime.properties |
| GET  | /druid/coordinator/v1/rules/{dataSource-Name}/history?count=<n>           | Returns last entries of audit history of rules for a specified datasource.   |
| POST | /druid/coordinator/v1/rules/{dataSource-Name}                             | POST with a list of rules in JSON form to update rules.  |

Note that all interval URL parameters are ISO 8601 strings delimited by a \_ instead of a / (e.g., 2016-06-27\_2016-06-28).

### Master Server

#### Intervals

|     |   |  |
|-----|---|--|
| GET | /druid/coordinator/v1/intervals                   | Returns all intervals for all datasources with total size and count.                         |
| GET | /druid/coordinator/v1/intervals/{interval}        | Returns aggregated total size and count for all intervals that intersect given isointerval.  |
| GET | /druid/coordinator/v1/intervals/{interval}?simple | Returns total size and count for each interval within given isointerval.                     |
| GET | /druid/coordinator/v1/intervals/{interval}?full   | Returns total size and count for each datasource for each interval within given isointerval. |

Note that all interval URL parameters are ISO 8601 strings delimited by a \_ instead of a / (e.g., 2016-06-27\_2016-06-28).

### Master Server

#### Compaction Configuration

|        |  |   |
|--------|--|---|
| GET    | /druid/coordinator/v1/config/compaction  | Returns all compaction configs.   |
| GET    | /druid/coordinator/v1/config/compaction/{dataSource}                                   | Returns a compaction config of a dataSource.  |
| POST   | /druid/coordinator/v1/config/compaction/taskslots?ratio={someRatio}&max={someMaxSlots} | Update the capacity for compaction tasks. ratio and max are used to limit the max number of compaction tasks. |
| POST   | /druid/coordinator/v1/config/compaction  | Creates or updates the compaction config for a dataSource.  |
| DELETE | /druid/coordinator/v1/config/compaction/{dataSource}                                   | Removes the compaction config for a dataSource.   |

See Compaction Configuration for configuration details.



By **kinger1999**

Published 19th June, 2019.

Last updated 19th June, 2019.

Page 4 of 8.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

### Master Server

#### Server Information

|     |                                      |   |
|-----|--------------------------------------|---|
| GET | /druid/coordinator/v1/servers        | Returns a list of servers URLs using the format {hostname}:{port}.  |
| GET | /druid/coordinator/v1/servers?simple | Returns a list of server data objects in which each object has the following keys: <i>host</i> : host URL include ( <i>{hostname}:{port}</i> ) <i>type</i> : process type (indexer-executor, historical) <i>currSize</i> : storage size currently used <i>maxSize</i> : maximum storage size <i>priority</i> tier |

### Overlord

#### Leadership

|     |                            |  |
|-----|----------------------------|--|
| GET | /druid/indexer/v1/leader   | Returns the current leader Overlord of the cluster.                  |
| GET | /druid/indexer/v1/isLeader | This returns a JSON object with field "leader", either true or false |

### Overlord

#### Tasks

|      |  |  |
|------|--|--|
| GET  | /druid/indexer/v1/tasks                  | Retrieve list of tasks. Accepts query string parameters state, datasource, createdTimeInterval, max, and type. |
| GET  | /druid/indexer/v1/completeTasks          | Retrieve list of complete tasks. Equivalent to /druid/indexer/v1/tasks?state=complete.                         |
| GET  | /druid/indexer/v1/runningTasks           | Retrieve list of running tasks. Equivalent to /druid/indexer/v1/tasks?state=running.                           |
| GET  | /druid/indexer/v1/waitingTasks           | Retrieve list of waiting tasks. Equivalent to /druid/indexer/v1/tasks?state=waiting.                           |
| GET  | /druid/indexer/v1/pendingTasks           | Retrieve list of pending tasks. Equivalent to /druid/indexer/v1/tasks?state=pending.                           |
| GET  | /druid/indexer/v1/task/{taskId}          | Retrieve the 'payload' of a task.  |
| GET  | /druid/indexer/v1/task/{taskId}/status   | Retrieve the status of a task.   |
| GET  | /druid/indexer/v1/task/{taskId}/segments | Retrieve information about the segments of a task.   |
| GET  |  | This API is deprecated and will be removed in future releases.   |
| GET  | /druid/indexer/v1/task/{taskId}/reports  | Retrieve a task completion report for a task. Only works for completed tasks.                                  |
| POST | /druid/indexer/v1/task                   | Endpoint for submitting tasks and supervisor specs to the Overlord. Returns the taskId of the submitted task.  |
| POST | /druid/indexer/v1/task/{taskId}/shutdown | Shuts down a task.   |



By [kinger1999](#)

Published 19th June, 2019.

Last updated 19th June, 2019.

Page 5 of 8.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

### Overlord (cont)

|        |   |  |
|--------|---|--|
| POST   | /druid/indexer/v1/datasources/{dataSource}/shutdownAllTasks | Shuts down all tasks for a dataSource.   |
| POST   | /druid/indexer/v1/taskStatus                                | Retrieve list of task status objects for list of task id strings in request body.                      |
| DELETE | /druid/indexer/v1/pendingSegments/{dataSource}              | Manually clean up pending segments table in metadata storage for dataSource.<br>Returns a JSON object. |

### Overlord

#### Supervisors

|      |   |  |
|------|---|--|
| GET  | /druid/indexer/v1/supervisor                          | Returns a list of strings of the currently active supervisor ids.                                |
| GET  | /druid/indexer/v1/supervisor?full                     | Returns a list of objects of the currently active supervisors.                                   |
| GET  | /druid/indexer/v1/supervisor/<supervisorId>           | Returns the current spec for the supervisor with the provided ID.                                |
| GET  | /druid/indexer/v1/supervisor/<supervisorId>/status    | Returns the current status of the supervisor with the provided ID.                               |
| GET  | /druid/indexer/v1/supervisor/history                  | Returns an audit history of specs for all supervisors (current and past).                        |
| GET  | /druid/indexer/v1/supervisor/<supervisorId>/history   | Returns an audit history of specs for the supervisor with the provided ID.                       |
| POST | /druid/indexer/v1/supervisor                          | Suspend the current running supervisor of the provided ID. Responds with updated SupervisorSpec. |
| POST | /druid/indexer/v1/supervisor/<supervisorId>/suspend   | Suspend the current running supervisor of the provided ID. Responds with updated SupervisorSpec. |
| POST | /druid/indexer/v1/supervisor/suspendAll               | Suspend all supervisors at once.   |
| POST | /druid/indexer/v1/supervisor/<supervisorId>/resume    | Resume indexing tasks for a supervisor. Responds with updated SupervisorSpec.                    |
| POST | /druid/indexer/v1/supervisor/resumeAll                | Resume all supervisors at once.  |
| POST | /druid/indexer/v1/supervisor/<supervisorId>/reset     | Reset the specified supervisor.  |
| POST | /druid/indexer/v1/supervisor/<supervisorId>/terminate | Terminate a supervisor of the provided ID.   |
| POST | /druid/indexer/v1/supervisor/terminateAll             | Terminate all supervisors at once.   |
| POST | /druid/indexer/v1/supervisor/<supervisorId>/shutdown  | Shutdown a supervisor.   |
| POST | /druid/indexer/v1/supervisor/<supervisorId>/suspend   | Suspend the current running supervisor of the provided ID. Responds with updated SupervisorSpec. |



By [kinger1999](#)

Published 19th June, 2019.

Last updated 19th June, 2019.

Page 6 of 8.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

### Overlord

#### Dynamic Configuration

|      |   |  |
|------|---|--|
| GET  | /druid/indexer/v1/worker  | Retrieves current overlord dynamic configuration.  |
| GET  | /druid/indexer/v1/worker/history?-interval={interval}&counter={count} | Retrieves history of changes to overlord dynamic configuration. Accepts interval and count query string parameters to filter by interval and limit the number of results respectively. |
| GET  | /druid/indexer/v1/scaling   | Retrieves overlord scaling events if auto-scaling runners are in use.  |
| POST | /druid/indexer/v1/worker  | Update overlord dynamic worker configuration.  |

### Data Server

#### MiddleManager

|      |   |  |
|------|---|--|
| GET  | /druid/worker/v1/enabled                | Check whether a MiddleManager is in an enabled or disabled state                                   |
| GET  | /druid/worker/v1/tasks                  | Retrieve a list of active tasks being run on MiddleManager.  |
| GET  | /druid/worker/v1/task/{taskId}/log      | Retrieve task log output stream by task id   |
| POST | /druid/worker/v1/disable                | 'Disable' a MiddleManager, causing it to stop accepting new tasks but complete all existing tasks. |
| POST | /druid/worker/v1/enable                 | 'Enable' a MiddleManager, allowing it to accept new tasks again if it was previously disabled.     |
| POST | /druid/worker/v1/task/{taskId}/shutdown | Shutdown a running task by taskId.   |

### Data Server

#### Peon

|     |   |   |
|-----|---|---|
| GET | /druid/worker/v1/chat/{taskId}/rowStats           | Retrieve a live row stats report from a Peon. See task reports for more details.      |
| GET | /druid/worker/v1/chat/{taskId}/unparseable-Events | Retrieve an unparseable events report from a Peon. See task reports for more details. |

### Data Server

#### Historical Segment Loading

|     |                                  |   |
|-----|----------------------------------|---|
| GET | /druid/historical/v1/-loadstatus | Returns JSON of the form {"cacheInitalized":<value>}, where value is either true or false indicating if all segments in the local cache have been loaded. |
|-----|----------------------------------|---|



By [kinger1999](#)

Published 19th June, 2019.

Last updated 19th June, 2019.

Page 7 of 8.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

### Data Server (cont)

GET /druid/historical/v1/readiness Similar to /druid/historical/v1/loadstatus, but instead of returning JSON with a flag, responses 200 OK if segments in the local cache have been loaded, and 503 SERVICE UNAVAILABLE, if they haven't.

### Query Server

Broker

Datasource Information

GET /druid/v2/datasources Returns a list of queryable datasources.

GET /druid/v2/datasources/{dataSourceName} Returns the dimensions and metrics of the datasource.

Optionally, you can provide request parameter "full" to get list of served intervals with dimensions and metrics being served for those intervals. You can also provide request param "interval" explicitly to refer to a particular interval.

GET /druid/v2/datasources/{dataSourceName}/candidates?intervals={comma-separated-intervals}&numCandidates={numCandidates} Returns segment information lists including server locations for the given datasource and intervals. If "numCandidates" is not specified, it will return all servers for each interval.

### Query Server

Load Status

GET /druid/broker/v1/loadstatus Returns a flag indicating if the Broker knows about all segments in Zookeeper.

### Query Server

Queries

POST /druid/v2/ The endpoint for submitting queries. Accepts an option ?pretty that pretty prints the results.

POST /druid/v2/candidates/ Returns segment information lists including server locations for the given query..



By [kinger1999](#)

Published 19th June, 2019.

Last updated 19th June, 2019.

Page 8 of 8.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>