Sphinx data provider functional object message protocol

Request protocol

Json format of message:

```
{
  "type": <message_type>,
  "data": [{<message_body>}],
  “ttl”: <message_ttl>
}
```

message_type – numeric integer {0, 1, 2}; 0 – search, 1 – index, 2 - manage
message_body – Json data different for each query type:
message_ttl – message TTL for network transport level.

type 0 – search:

```
{
  "q": “<query_string>”,
  "filters": “<filters_json>”,
  "parameters": [{<query_parameters>}]
  “order”: [{<order_parameters>}]
}
```

query_string – searched string, base64encoded*
filters_json – json string with array of filter items.

query_parameters – array of one dimensional key-value arrays – parameter_name/parameter_value:

```
“queryId”:”<query_ID>”,
“jsonType”:”<return_json_type_mask>”,
“sort_by”:”<sort_field_name>”,
“order_by”:”<sort_order_by>”,
“offset”:”<offset_results>”,
“limit”:”<limit_results>”,
“cutoff”:”<cutoff_results>”,
“return_jason_ext_fields”: [{<external_fields_names>}],
“max_results”: “<max_results_number>”,
“timeout”: “<timeout_value>”
```

Items description:

query_ID – numeric unsigned integer, ID of query
return_json_type_mask – numeric unsigned integer bit mask bit 0 – fill documents “Match Info” (MI) data, 1 – fill “Request Info” (RI) data, 2 – fill “Attributes” (At) of “Match Info” (MI), 3 – fill “Word Info” (WI) data of RI.
sort_field_name – name of field that will be used by Sphinx searchd to sort results,

sort_order_by – integer order direction {0 - SPH_SORT_RELEVANCE, 1 - SPH_SORT_ATTR_DESC, 2 – SPH_SORT_ATTR_ASC, 3 - SPH_SORT_TIME_SEGMENTS, 4 - SPH_SORT_EXTENDED, 5 – SPH_SORT_EXPR} detailed description see in addition 2 below.

offset_results – numeric unsigned integer, offset result data
limit_results – numeric unsigned integer, the amount necessary for obtaining results from Sphinx (at one data node).
cutoff_results – numeric unsigned integer, used to improve the monitoring of performance. The parameter specifies the searchd forcibly stop the search as soon as it was found and processed cutoff matches.
external_fields_names – json string array of extended fields names (Sphinx attributes) that need to be included in to the match info returned. If array is empty – all extended fields according with the schema must be returned. If array is filled – only specified fields must be returned. If field name not defined in the schema or not supported – it just ignored.
max_results_number – integer value - maximum number of results can be returned as response result from router after reducing.
timeout_value – TCP socket read response timeout, ms for Sphinx searchd. If exceeded – empty results returned.

order_parameters – array key-value – parameter name / parameter value:
“algorithm”:”<weight_algorithm>”,
“fields”:[”<fields_names>”],
“order_by”:”<order_type>”,

Items description:

weight_algorithm – integer value defines what algorithm of weight calculation will be used {0 – the value is calculated as accumulation of hexadecimal values of fields listed in the **fields_names** array. If array is empty – only one Sphinx weight value used.}
fields_names – array of names of extended fields that will be accumulated from right to left. (Lower array indexed value take lower position in weight string). Extended fields defined by xml schema of index or one of set of predefined names: node_name, node_number, sphinx_weight and doc_id.
order_type – integer value specifies ordering direction {0 – not sort, 1 – sort ascending by weight, 2 – sort descending}.

Filter item:

```
{
  "type": <filter_type>,
  "attribute": "<attribute_name>",
  "values": ["<attribute_value>",   ],
  "exclude": <exclude_value>
}
```

filter_type – type of the filter, 0 – simple, 1 – range integer, 2 – range float, 3 – simple multi, creates copies of filter for each value in values array.
attribute_name – name of the Sphinx attribute field.
values – array of attribute values that depends on "type".
attribute_value – in case of type is 0 – it is value of the Sphinx attribute, in case of type is 1 – it is min or max value range. Min and max values items set sequentially as item 0 and item 1 in array.
exclude – value of exclude parameter.

Addition 1: Example of filters json array contains two filters of type "0" and "1":

```
[
    {
      "type": 0,
      "attribute": "mmedia",
      "values": [
        "123",
        "345",
        "234"
      ],
      "exclude": 0
    },
    {
      "type": 1,
      "attribute": "bseo",
      "values": [
        "1",
        "100"
      ],
      "exclude": 0
    }
]
```

Addition 2 : sort_order_by field value description related with Sphinx sorting modes (see query_parameters, parameter sort_order_by):

SPH_SORT_RELEVANCE	Sort by relevance in descending order (best matches first).
SPH_SORT_ATTR_DESC	Sort by an attribute in descending order (bigger attribute values first).
SPH_SORT_ATTR_ASC	Sort by an attribute in ascending order (smaller attribute values first).
SPH_SORT_TIME_SEGMENTS	Sort by time segments (last hour/day/week/month) in descending order, and then by relevance in descending order.
SPH_SORT_EXTENDED	Sort by SQL-like combination of columns in ASC/DESC order.
SPH_SORT_EXPR	Sort by an arithmetic expression.

type 1 – indexation:

```
{
  "name": “<index_name>”,
  "body": “<document_file>”,
  "parameters": [{<query_parameters>}]
}

index_name – name of the index, if empty – current index used.
document_file – document body contains document items for each
              indexation unit according with schema, base64encoded*.
              Document data will be checked and appended to the default
              branch. If default branch is not exists – it will be
              created. After that index rebuild and merge actions will
              be done. If all actions are okay and this index was
              started it will be restarted.
query_parameters – Json with custom items structure (reserved)
```

type 2 – admin command:

```
{
  "command": “<command_string>”,
  "options": “<command_options_string>”,
}
command – admin command text, string
options – admin command options Json with custom items structure
```

Response protocol

```
{
  "error_code": <error_code>,
  "error_message": “<error_message>”
  "data": “<response_body>”
  "time": “<elapsed_time>”
}
```

error_code – error state code, 0 – no errors, >0 – some error
error_message – error description text

data – Sphinx search result Json
time – internal common execution time (including request parsing and response creation), msec

Appendix 1.

Error codes and explanation

Code	Description
0	No errors - return successful without errors
1	Parse error – wrong Json format or another parse error.
2	Command error (not supported, not recognized, etc…)
3	General internal error of functional object
4	Error load configuration file
5	Error make default json
1000	General internal error of search request
1001	Error search because not activity searchd
1002	Error search because seachd stoped but pid file exists
1011	Error of set match mode
1012	Error of set sort mode
1013	Error of set ranking mode
1014	Error of set server (host and port)
1015	Error of set limits (max results number of document's and etc.)
1016	Error of add filters
1017	Error of set allowed query time
1021	Error open connection
1022	Error close connection
1031	Error start of searchd daemon
1032	Error stop of searchd daemon
2000	General internal error of indexation request
3000	General internal error of admin request
3001	Internal error command initialization
3011	Internal error json serialize

3012 Internal error json unserialize

3021 Internal error copy file

3022 Internal error move file

3023 Internal error remove file

3024 Received bad branch name

3025 Internal error make directories

3026 Internal error clean directories

3027 Internal error get file list

3028 Internal error get index list

3029 Internal error get free allowed port

3031 Internal error make source

3032 Internal error make index

3033 Internal error make merge

3034 Internal error make trunk index

3035 Internal error delete trunk index

3036 Internal error move index

3101 Internal error run command create index

3102 Internal error run command check index

3103 Internal error run command store data file

3104 Internal error run command store schema file

3105 Internal error run command index rebuild

3106 Internal error run command index start

3107 Internal error run command index stop

3108 Internal error run command index merge

3109 Internal error run command index merge trunk

3110 Internal error run command index delete data file

3111 Internal error run command index delete schema file

3112 Internal error run command index append data file

3113 Internal error run command index delete doc (set list ID's)

3114 Internal error run command index delete doc number

3115 Internal error run command index pack doc data

3116 Internal error run command index remove

3117 Internal error run command index copy

3118 Internal error run command index rename

3119 Internal error run command set configuration parameter value

3120 Internal error run command get configuration parameter value

3121 Internal error run command index check schema

3122 Internal error run command index status about Sphinx searchd

3123 Internal error run command index status from Sphinx indextool

3124 Internal error run command index get max doc ID

3125 Internal error run command get list of data file names

3126 Internal error run command get list of branches file names

3127 Internal error run command get branches files parameters section fields

3128 Internal error run command get branches indexes status from Sphinx indextool.

3129 Internal error run command index connection

3130 Internal error run command index disconnect

Appendix 2.
Sphinx search result Json structure

```
{
  “max_results”: <max_results_number>,
  “sort_mode”: <sort_mode>,
  “ttl”: “<ttl_value>”,
  “doc_info”: “<doc_info_json>”,
  “match_info”: “<match_info_json>”,
  “word_info”: “<word_info_json>”
}
```

max_results_number - maximum number of results can be returned.
sort_mode - mode of results sorting {0, 1, 2}. 0 – not sort, 1 – sort ascending by weight, 2 – sort descending by weight.
TODO:”rename to order_by to have the same parameter name as in the request message”
ttl_value – integer value – Time To Leave in milliseconds for reducer task.
Reducer task accumulates aggregates and sorts the results items from messages that received from all nodes connected to reducer node. If some connected node not responds in time
the node response will be rejected by reducer task manager and lost. Default value is 1 sec.
doc_info_json - json string content list of doc ID's and weight's
match_info_json – json string content list of document's information
word_info_json – json string content list of statistic word information

docs json:

```
{
   [{
     “doc_id”: <doc_id>,
     “weight”: <weight>
  },… NULL]
}
```

doc_id – document ID.
weight – weight of document's.

match info json:

```
{
   [{
     “doc_id”: <doc_id>,
     “weight”: <weight>,
     “attr”: [{
     “name”: ”<attribute_name>”,
     “value”: “<attribute_value>”
     },… NULL],
  },… NULL]
}
```

doc_id – document ID.
weight – weight of document's.
attribute_name – name of attribute parameters.
attribute_value – value of attribute parameters.

word info json:

```
{
   [{
      “word”: ”<word>”,
      “hits”: <hits_count>,
      “docs”: <docs_count>,
      “node_name”: ”<node_name>”,
      “query_id”: <query_id>,
      “query”: “<query>”,
      “total”: <total>,
      “total_found”: <total_found>,
      “time_msec”: <time_msec>
  },… NULL]
}
```

word – word used in query, base64encoded.
hits_count – count of hits word's in query.
docs_count – count of doc's used in query.
node_name – name of used node.
query_id – ID of query.
query – string of query.
total – count of retrieved document's.
total_found - count all found document's.
msec – time execute of request (in msec).

Sphinx search result data Json structure

```
{
   “MI”:[{
            “At”:[{ “<attribute_name>”: “<attribute_value>”},...NULL],
            “Id”:“<doc_id>”,
            “W”:“<weight>”},...NULL
         ],
   “RI”:[{
            “WI”:[{
                  “d”:<docs>,
                  “h”:<hits>,
                  “w”:”<word>”},...NULL
                  ],
            “node”:”<node>”,
            “q”:“<query>”,
            “qid”:<query_id>,
            “max”:<max_results_number>,
            “order”:<order_by>,
            “r”:<retrived_count>,
            “f”:<found_count>,
            “time”:<time_msec>},...NULL
         ]
}
```

MI – array of Sphinx search “Match Information”
AT – array of Attributes defined in xml schema document

attribute_name – name of attribute parameter, defined by xml document schema as Sphinx “sphinx:attr” tags. One of explicit attribute is Sphinx weight named “sphinx_weight” that always present and not related with xml schema definitions.
attribute_value – value of attribute parameter
doc_id – document ID defined in xml schema document.
weight – string weight of document calculated using some pre-defined or custom algorithm.

RI – array of search “Request Information”
WI – array of Sphinx search “Word Information”

docs – count of doc's used in query.
hits – count of hits word's in query.
word – word used in query, base64encoded.

node – name of used node.
query – string of query.
query_id – ID of query.
max_results_number - maximum number of results can be returned.
order_by - mode of results sorting {0, 1, 2}. 0 – not sort (order not changed), 1 – sort ascending, 2 – sort descending.
retrived_count – count of retrieved document's.
found_count – total count all found document's (candidates).
time – time execute of request, msec.

Base64encoded* - for possibility to use base64 code/encode need make declaration of JSON_USE_BASE64=1 for project code.