

Admin commands of Sphinx Functional Object

Command name	Command options	Description
INDEX_CREATE Tested	Json with structure: { "index": "<index_name>", }	Create new index with specified name. Fields: index_name – index name, corresponds to index directory name Return: nothing
INDEX_CHECK Tested	Json with structure: { "index": "<index_name>", }	Checks are index exists with name specified. Fields: index_name – index name, corresponds to index directory name Return: 1 if index exists and 0 – if not
INDEX_STORE_DATA_FILE Tested	Json with structure: { "index": "<index_name>", "branch": "<branch_name>", "data": "<data_content>" }	Store content in the branch data file. Fields: index_name – index name, corresponds to index directory name branch_name – branch data set name, corresponds to branch data file name data_content – data file content, base64encoded Return: nothing
INDEX_STORE_SCHEMA_FILE Tested	Json with structure: { "index": "<index_name>", "data": "<schema_content>" }	Store content in the index schema file. Fields: index_name – index name, corresponds to index directory name schema_content – schema's file content, base64encoded Return: nothing
INDEX_REBUILD Tested	Json with structure: {	Rebuild index according with schema and branch's data files. Index name suppose index definition configuration file.

	<pre> "index": "<index_name>", "branches": ["<branch_name>", ...] } </pre>	<p>Creates index' files in the index' directory. Fields:</p> <p>index_name – index name, corresponds to index directory name and configuration file for indexation.</p> <p>branches – array of names of branches data files to be processed and index file produced. If the branch_name is "*" character – all data files need to be processed.</p> <p>Return:</p> <p>String, list of branches names that was not included in rebuild process due to error of file operation, structure, etc...</p>
INDEX_SET_DATA_DIR Not implemented	Reserved	Change default path to the data directory
INDEX_START Tested	<p>Json with structure:</p> <pre> { "index": "<index_name>" } </pre>	<p>Start or restart sphinx search daemon and use specified index for search. Fields:</p> <p>index_name – index name, corresponds to index directory name and configuration file for Sphinx searchd. If empty string value – default index name used.</p> <p>Return:</p> <p>nothing</p>
INDEX_STOP Tested	<p>Json with structure:</p> <pre> { "index": "<index_name>" } </pre>	<p>Stop sphinx searchd and index usage. Fields:</p> <p>index_name – index name, corresponds to index directory name and configuration file for Sphinx searchd. If empty string value – default index name used.</p> <p>Return:</p> <p>nothing</p>
INDEX_MERGE Tested	<p>Json with structure:</p> <pre> { "index": "<index_name>", "branches": ["<branch_name>", ...] } </pre>	<p>Merge specified index branches and overwrite current trunk. Produces new trunk index. Fields:</p> <p>index_name – index name, corresponds to index directory name and configuration file for indexation.</p> <p>branches – array of names of branches index files to be processed. If the branch_name is "*" character – all index</p>

		<p>files need to be processed.</p> <p>Return: List of branch names that were not merged, separated by comma or empty string in case of all requested branches indexes was merged.</p>
<p>INDEX_MERGE_TRUNK Implemented, need test.</p>	<p>Json with structure:</p> <pre>{ "index": "<index_name>", "branches": ["<branch_name>", ...] }</pre>	<p>Merge specified indexes branches with current trunk one by one. Produces updated trunk index. Fields:</p> <p>index_name – index name, corresponds to index directory name and configuration file for indexation.</p> <p>branches – array of names of branches index files to be processed.</p> <p>Return: List of branch names that were not merged, separated by comma or empty string in case of all requested branches indexes was merged.</p>
<p>INDEX_DELETE_DATA_FILE Tested</p>	<p>Json with structure:</p> <pre>{ "index": "<index_name>", "branches": ["<branch_name>", ...] }</pre>	<p>Delete data and index files of branch. Fields:</p> <p>index_name – index name, corresponds to index directory name.</p> <p>branches – array of names of branches data files to be deleted. If the branch_name is "*" character – all data files need to be deleted.</p> <p>Return: List of branch names that were not deleted, separated by comma or empty string in case of all requested branches indexes was deleted.</p>
<p>INDEX_DELETE_SCHEMA_FILE Tested</p>	<p>Json with structure:</p> <pre>{ "index": "<index_name>" }</pre>	<p>Delete schema file(s).</p> <p>Return: nothing</p>

INDEX_APPEND_DATA_FILE Tested	Json with structure: <pre>{ "index": "<index_name>", "branch": "<branch_name>", "data": "<data_content>" }</pre>	Appends data file with document content formatted according with the schema specification. Fields: index_name – index name, corresponds to index directory name branch_name – branch data set name, corresponds to branch data file name data_content – data file content, base64encoded Return: nothing
INDEX_DELETE_DOC Tested	Json with structure: <pre>{ "index": "<index_name>", "documents": ["<document_Id>", ... NULL] }</pre>	Add document Id in to the deleted documents list for specified index. Fields: index_name – index name, corresponds to index directory name. documents – array of document Id. Return: nothing
INDEX_DELETE_DOC_NUMBER Tested	Json with structure: <pre>{ "index": "<index_name>" }</pre>	Get number of documents in the deleted documents list. Fields: index_name – index name, corresponds to index directory name. Return: Number of documents.
INDEX_PACK_DOC_DATA Tested	Json with structure: <pre>{ "index": "<index_name>", "branches": ["<branch_name>", ...] }</pre>	Processes branches' data files and delete documents listed in deleted documents list by document Id. Delete duplicated documents, leave last appended. Overwrites branch file(s). Cleans deleted documents list. Fields: index_name – index name, corresponds to index directory name. branches – array of names of branch data files to be processed. If the branch_name is "*" character – all branch data files need to be processed.

		Return: Number of document in the branch after packing.
INDEX_REMOVE Tested	Json with structure: { "index": "<index_name>" }	Remove index directory and all files. Fields: index_name – index name, corresponds to index directory name. Return: nothing
INDEX_COPY Tested	Json with structure: { "index_from": "<index_name>", "index_to": "<index_name>" }	Copy index directory and all files and subdirectories in to the new directory. Fields: index_from – source index name, corresponds to index directory name. index_to – destination index name, corresponds to index directory name. Return: nothing
INDEX_RENAME Tested	Json with structure: { "index_from": "<index_name>", "index_to": "<index_name>" }	Rename index directory. Fields: index_from – old index name, corresponds to index directory name. index_to – new index name, corresponds to index directory name. Return: nothing
INDEX_SET_CONFIG_VAR Partially implemented, but not marked red.	Json with structure: { "index": "<index_name>", "branch": "<branch_name>", "section": "<section_name>", "parameter": "<parameter_name>", "value": "<parameter_value>" }	Set configuration parameter value. Fields: index_name – index name, corresponds to index directory name. branch_name – name of branch, if empty – trunk index config. file requested. section_name – name of section in the configuration file. parameter_name – name of the parameter in the

	} }	configuration file. parameter_value – value of the parameter in the configuration file. Return: Nothing
INDEX_GET_CONFIG_VAR Partially implemented, but not marked red.	Json with structure: { "index": "<index_name>", "branch": "<branch_name>", "section": "<section_name>", "parameter": "<parameter_name>" }	Get configuration parameter value. Fields: index_name – index name, corresponds to index directory name. branch_name – name of branch, if empty – trunk index config. file requested. section_name – name of section in the configuration file. parameter_name – name of the parameter in the configuration file. Return: Value of the parameter in the configuration file.
INDEX_CHECK_SCHEMA Tested	Json with structure: { "index": "<index_name>", "data": "<schema_content>" }	Checks are schema valid and applicable for index specified. Fields: index_name – index name, corresponds to index directory name schema_content – schema's file content, base64encoded Return: nothing If schema is not valid or/and not applicable for index, the error code is set grater zero.
INDEX_STATUS_SEARCHD Tested	Json with structure: { "index": "<index_name>" }	Get status information from Sphinx searchd. Fields: index_name – index name, corresponds to index directory name, used to provide the searchd with proper configuration file in case of several instances of searchd are started on the same host. If empty string value – default last

		<p>started index information returned.</p> <p>Return: String printout from Sphinx searchd application command: searchd -status</p>
<p>INDEX_STATUS Tested</p>	<p>Json with structure: { "index": "<index_name>" }</p>	<p>Get status information from Sphinx indextool. Fields:</p> <p>index_name – index name, corresponds to index directory name, used to provide the searchd with proper configuration file in case of several instances of searchd are started on the same host. If empty string value – default last started index information returned.</p> <p>Return: String printout from Sphinx indextool application command: indextool -dumpheader <index_name></p>
<p>INDEX_MAX_DOC_ID Tested</p>	<p>Json with structure: { "index": "<index_name>" }</p>	<p>Get maximal document Id that was processed during last rebuild action:</p> <p>index_name – index name, corresponds to index directory name, used to provide the searchd with proper configuration file in case of several instances of searchd are started on the same host. If empty string value – default last started index information returned.</p> <p>Return: Integer number.</p>
<p>INDEX_DATA_LIST Tested</p>	<p>Json with structure: { "index": "<index_name>" }</p>	<p>Get list of data file names stored in the index directory:</p> <p>index_name – index name, corresponds to index directory</p>

	<pre>} </pre>	<p>name, used to provide the searchd with proper configuration file in case of several instances of searchd are started on the same host. If empty string value – default last started index information returned.</p> <p>Return: String with names of data files separated by comma.</p>
INDEX_BRANCHES_LIST	<p>Json with structure:</p> <pre>{ "index": "<index_name>" }</pre>	<p>Get list of branches file names stored in the index directory:</p> <p>index_name – index name, corresponds to index directory name, used to provide the searchd with proper configuration file in case of several instances of searchd are started on the same host. If empty string value – default last started index information returned.</p> <p>Return: String with names of branches files separated by comma.</p>
INDEX_BRANCHES_INFO Need test.	<p>Json with structure:</p> <pre>{ "index": "<index_name>", "branches": ["<branch_name>",...] }</pre>	<p>Get branches files parameters section fields. Fields:</p> <p>index_name – index name, corresponds to index directory name and configuration file for indexation.</p> <p>branches – array of names of branches index files to be processed. If the branch_name is "*" character – all index files need to be processed.</p> <p>Return: List of items for each branch separated by ampersand in format: <branch_name>=<fields_list>[&...] The fields_list – url-encoded list of parameters fields items separated by comma in format: <field_name=field_value>[,...]</p>

INDEX_BRANCHES_STATUS Need test.	Json with structure: <pre>{ "index": "<index_name>", "branches": ["<branch_name>", ...] }</pre>	Get branches indexes status information from Sphinx indextool. Fields: index_name – index name, corresponds to index directory name and configuration file for indexation. branches – array of names of branches index files to be processed. If the branch_name is "*" character – all index files need to be processed. Return: List of items for each branch separated by ampersand in format: <branch_name>=<information_string>[&...] The information_string – url-encoded string printout from Sphinx indextool application command: indextool -dumpheader <index_name>
INDEX_CONNECT Need test.	Json with structure: <pre>{ "index": "<index_name>" }</pre>	Connect to the sphinx search daemon and use specified index for search. Fields: index_name – index name, corresponds to index directory name and configuration file for Sphinx searchd. If empty string value – default index name used. Return: Nothing
INDEX_DISCONNECT Need test.	Json with structure: <pre>{ }</pre>	Disconnect from the sphinx search daemon. If not connected no error set. Return: Nothing

* In case of command success execution and returns nothing – the **error_code** field is zero value and **data** field is empty.

* In case of command returns value it is set in the **data** field.

* In case of error happened – error code set in the **error_code** field and error message set in the **error_message** field

Trivial commands sequence during index life time cycle:

Initial phase

1. INDEX_CREATE (creates directory for index files and another kind settings files and data files directories)
2. INDEX_STORE_SCHEMA_FILE
3. INDEX_STORE_DATA_FILE
4. INDEX_APPEND_DATA_FILE (can be skipped)
5. INDEX_PACK_DOC_DATA (can be skipped)
6. INDEX_REBUILD
7. INDEX_MERGE
8. INDEX_START

Accumulate phase

9. INDEX_DELETE_DOC
10. INDEX_STORE_DATA_FILE
11. INDEX_APPEND_DATA_FILE

Rebuild phase

12. INDEX_PACK_DOC_DATA
13. INDEX_REBUILD
14. INDEX_MERGE
15. INDEX_START

Remove phase

16. INDEX_STOP
17. INDEX_REMOVE