

1) Input JSON string is passed as a parameter to call method Process of class SphinxSearcher..

Next work take place in “Black Box”.

2) Call unserialize() - exctract data out input JSON string in object class type SphinxInputJsonMessageSearch.

3) For check result state after last operation call method isError() for class SphinxInputJsonMessageSearch.

4) Return of result. Must be false (no errors), otherwise next step of executing doesn't have sense.

5) Call getExternalFields() of class SphinxInputJsonMessageSearch for get list fields used as external.

6) Return list of fields.

7) Call setAllowedFields() of class SphinxSearcher for right initialization.

8) Call getOrderFields() of class SphinxInputJsonMessageSearch for get sequence fields need for calculate weight.

9) Return list of fields.

10) Call getQueryParameters() of class SphinxInputJsonMessageSearch for get list query parameters need to initialization of SphinxSearcher .

11) Return list of parameters.

12) Call setQueryId() of class SphinxResult for save parameter and possibility pack in JSON message after search processing.

13) Call setJsonType of class SphinxResult for initialization types creating JSON message after search processing.

14) Call setSortByName() of class SphinxResult for save parameter and possibility pack in JSON message after search processing.

15) Call setSortMode() of class SphinxSearcher for initialization types sort after make search request.

16) Call getOrderParameters() of class SphinxInputJsonMessageSearch for get list order parameters need to calculate weight after search processing.

17) Return list of parameters.

18) Call setLimits() of class SphinxSearcher for initialization received parameters before search.

19) Call resetFilters() of class SphinxSearcher for clear all old filters.

20) Call getFilters() of class SphinxInputJsonMessageSearch for get list filters possibility can apply to query request.

21) Return list of filters.

22) Call applyFilter() of class SphinxSearcher for initialization functional object before search and use filters for search processing.

23) Call searchDocument() of class SphinxSearcher for execute of search request.

24) Result of search request save in class SphinxResult.

25) Call setNodeName() of class SphinxResult for save parameter and possibility pack in JSON message after search processing.

26) Call setHandleType() of class SphinxResult for save parameter and possibility pack in JSON message after search processing.

27) Call setMaxResultNumber() of class SphinxResult for save parameter and possibility pack in JSON message after search processing.

28) Call setSortMode() of class SphinxResult for save parameter and possibility pack in JSON message after search processing.

29) Send type of strategy calculate of weight as parameters to class CalculateStrategyFactory to method create.

30) Call method create() of class CalculateStrategyFactory for create new object of class CalculateStrategy.

31) Return handle created object types CalculateStrategy.

32) Call getMatches() of class SphinxResult for get list of documents received after execute search request.

33) Send list of documents to class SphinxWeightCalculator.

34) Call setStrategy() of class SphinxWeightCalculator for set handle CalculateStrategy.

35) Call method calculate() of class SphinxWeightCalculator for execute calculate weights.

36) Update result weights of matches content in class SphinxResult.

37) Call getJSON() of class SphinxResult for create JSON string with result of search request.

38) Return result data as result execute search command of SphinxSearcher in “Black Box”.