GLYCAN LIST PAGE WEB SERVICE DOCUMENTATION

**Version1.0.**

**Updated (3/27/2018- Rupali Mahadik)**

**Pradeep Kumar Ragu Chanthar**

**René Ranzinger**

**CCRC, University of Georgia**

Contents

[Introduction 2](#_Toc505180765)

[Web service call 2](#_Toc505180766)

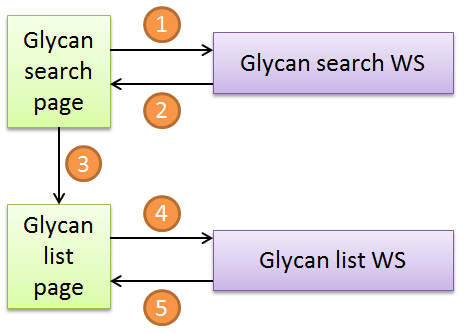
[Function 2](#_Toc505180767)

[Response 2](#_Toc505180768)

# Introduction

This web services provides the glycan list generated as result of a search. In a search webpage a web services for performing the search is called and an ID of the search result is returned. This ID is used as part of this web service to retrieve the individual elements of the search result.

# Glycan Search Workflow

User enters search criteria on the glycan search page.

1. JavaScript generates a JSON search query calls Glycan search WS.
2. Search WS returns the result list ID
3. Result list ID is passed to the Glycan list page
4. Result list ID is passed to glycan list WS together with information about sorting and number of requested results
5. Glycan list WS returns requested number of glycan results.

|  |  |
| --- | --- |
| Bildergebnis für please note icon | **Note!**  When performing a search the server has to assign a unique ID to each search result and store the search result and ID. This will also allow us to persist search results even across database changes. |

# Web service call

**HTTP Request method** POST

**Variables** *id* ID of the search result

*offset* Position in the search result the first entry comes from

*limit* Number of entries to retrieve starting with the entry at the offset position

*sort* Criteria to sort the list by (*id*, mass, number of monosaccharides, number of proteins, number of enzymes)

*order* Sorting direction (*asc*, *desc*)

Example:

<http://url.domain/get_glycan_list?id=AB345TB7&offset=50&limit=20&sort=id&order=asc>

This query will retrieve the element 50-69 from the search result AB345TB7 which was ordered by the glycan ID.

# Function

Based on the glycan list ID the webservice will sort the list by the provided criteria (sort and order) and return the requested chunk (offset, limit) of the list. The server side pagination is required since large result sets (more than 1000 results) will perform very poorly with standard tables. The result set will be encoded in JSON (see separate JSON schema) and return. The JSON contains the result set, information about the search query and pagination information.

# Response

**Content** JSON

**HTTP Content Type** application/json

**Access-Control-Allow-Origin** \*

|  |  |
| --- | --- |
| Bildergebnis für please note icon | **Warning**  To all Cross-Origin Resource Sharing (CORS) the web service has to allow all domains access. Otherwise browsers will not permit to call this web service by a webpage. Therefore the HTTP response header must contain the appropriate *Access-Control-Allow-Origin* statement. |

|  |  |
| --- | --- |
| Bildergebnis für please note icon | **Note**  An example JSON is available separately and the JSON data structure is documented in a JSON schema. |