

GMQL Services

After installation, the base URI (BASE_URI) for using the GMQL web services is <http://hostname:port/GMQL-services/rest/>. Any request for the web services should start with such link, adding to the BASE_URI the service name and its parameters (in case sensitive mode). *Hostname* and *port* should be respectively replaced with the *ip* or *domain name* of the machine where the GMQL services have been installed and the appropriate port of the web server on such machine that runs the web services.

GMQL services can be used by directly calling them and obtaining back their answer in xml format, or by creating a programming language code that calls them; for example, a Java project can be created with the following dependency to Maven:

```
<dependency>
  <groupId>com.sun.jersey</groupId>
  <artifactId>jersey-client</artifactId>
  <version>1.18.1</version>
</dependency>
```

```
<dependency>
  <groupId>com.sun.jersey</groupId>
  <artifactId>jersey-core</artifactId>
  <version>1.18.1</version>
</dependency>
```

And then its Java application should look like as follows:

```
import java.net.URI;
import java.security.NoSuchAlgorithmException;

import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.UriBuilder;

import com.sun.jersey.api.client.Client;
import com.sun.jersey.api.client.WebResource;
import com.sun.jersey.api.client.config.ClientConfig;
import com.sun.jersey.api.client.config.DefaultClientConfig;

public class testservices {
  public static void main(String[] args) throws NoSuchAlgorithmException {
    ClientConfig config = new DefaultClientConfig();

    Client client = Client.create(config);

    WebResource service = client.resource(getBaseURI());

    System.out.println(service.path("rest").path("datasets").path("listAll").
      path("Alice").
      accept(MediaType.APPLICATION_XML).get(String.class));
  }

  private static URI getBaseURI() {
    return UriBuilder.fromUri("http://localhost:8080/GMQL-services/").build();
  }
}
```

Web Service description

I. Repository Manager Services

- Register User to GMQL repository

<u>URI:</u>	BASE_URI/datasets/registerUser/{username}	
<u>Interface:</u>	Response registerUser(@PathParam("username") String sc)	
<u>Method:</u>	GET	
<u>Input:</u>	username	Plain-text representation of the user name. {username} should be replaced with the username. Like BASE_URI/datasets/registerUser/Alice
<u>Output:</u>	Response	Response that contains plain-text as "Registered"

- Unregister User to GMQL repository

<u>URI:</u>	BASE_URI/datasets/unRegisterUser/{username}	
<u>Interface:</u>	Response unRegisterUser(@PathParam("username") String sc)	
<u>Method:</u>	GET	
<u>Input:</u>	username	Plain-text representation of the user name. {username} should be replaced with the username. Like BASE_URI/datasets/unRegisterUser/Alice
<u>Output:</u>	Response	Response that contains plain-text as "unRegistered"

- List user datasets

<u>URI:</u>	BASE_URI/datasets/listAll/{username}	
<u>Interface:</u>	Response listAllDataSets(@PathParam("username") String sc)	
<u>Method:</u>	GET	
<u>Input:</u>	username	Plain-text representation of the user name. {username} should be replaced with the username. Like BASE_URI/datasets/listAll/Alice
<u>Output:</u>	Xml/json	xml/json representation of list of datasets found for the specific user.

- List samples in a user dataset

<u>URI:</u>	BASE_URI/datasets/listDSSamples/{dataset}/{username}	
<u>Interface:</u>	Response listAllDataSets(@PathParam("dataset") String dataset, @PathParam("username") String sc)	
<u>Method:</u>	GET	
<u>Input:</u>	username	Plain-text representation of the user name. {username} should be replaced with the username. Like BASE_URI/datasets/listDSSamples/annotations/Alice

<u>Output:</u>	Xml/json	xml/json representation of list of samples found in the dataset for the specific user.
----------------	----------	--

- Delete a user dataset

<u>URI:</u>	BASE_URI/datasets/delete/{dataSetName}/{username}	
<u>Interface</u>	Response deleteDataSet(@PathParam("username") String sc, @PathParam("dataSetName") String dataSetName)	
<u>Method:</u>	GET	
<u>Input:</u>	dataSetName	The name of the dataset to be deleted from the repository for the specified user name.
	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/datasets/delete/HG19_ANN/Alice
<u>Output:</u>	plain-text	Plain-text representation the word "OK"

- Prepare a dataset in a local folder

<u>URI:</u>	BASE_URI/datasets/prepare/{dataSetName}/{clean}/{username}	
<u>Interface</u>	Response prepareFileZip(@PathParam("username") String sc, @PathParam("dataSetName") String dataSetName, @PathParam("clean") String clean)	
<u>Method:</u>	POST	
<u>Input:</u>	dataSetName	The name of the dataset to be prepared in the repository for the specified user name.
	clean	Plain-text which is either true or false. To clean the previous folder if it exists and rebuild, or just send ready when it exists.
	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/datasets/prepare/HG19_ANN/true/Alice
<u>Output:</u>	plain-text	Plain-text representation the word "OK". Folder contains the datasets samples, copied from HDFS, and the query used to run it plus the vocabulary used in the metadata.

- Zip dataset folder and send it back to the user.

<u>URI:</u>	BASE_URI/datasets/zip/{dataSetName}/{username}	
<u>Interface</u>	Response downloadFileZip(@PathParam("username") String username, @PathParam("dataSetName") String dataSetName)	
<u>Method:</u>	GET	
<u>Input:</u>	dataSetName	The name of the dataset to be zipped from the repository local folder. This web service is usually used after preparing the local folder.

	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/datasets/zip/HG19_ANN/Alice
<u>Output:</u>	plain-text	Response containing the zip file as an attachment.

- Upload samples: only uploading samples to the server local file system, without adding to the repository, in a folder named as the dataset name.

<u>URI:</u>	BASE_URI/datasets/uploadSamples/{dataSetName}/{status}/{username}	
<u>Interface:</u>	@Consumes (MediaType. <i>MULTIPART_FORM_DATA</i>) public Response uploadSample(@PathParam("username") String sc, @FormDataParam("file") InputStream uploadedInputStream, @PathParam("dataSetName") String dataSetName, @PathParam("status") String status, @FormDataParam("file") FormDataContentDisposition fileDetails)	
<u>Method:</u>	POST	
<u>Input:</u>	dataSetName	The name of the dataset for the samples to be added to.
	status	Plain-text which has the value "first", "next", or "schema". When the file is "first", we clear a temp folder to receive all the samples, one by one, when the schema is received we store it next to the samples folder.
	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/datasets/uploadSamples/HG19_ANN/first/Alice. Note: Sample file should be attached to the HTTP request.
<u>Output:</u>	plain-text	Plain-text representation the word "OK".

- Add samples to a dataset. Samples are indexed and added to the chosen dataset in the HDFS.

<u>URI:</u>	BASE_URI/datasets/addSamples/{dataSetName}/{username}	
<u>Interface:</u>	@Consumes (MediaType. <i>MULTIPART_FORM_DATA</i>) public Response addSamples(@PathParam("username") String sc, @PathParam("dataSetName") String dataSetName)	
<u>Method:</u>	POST	
<u>Input:</u>	dataSetName	The name of the dataset for the samples to be added to from the local folder where are the samples uploaded using the previous web service. This web service is usually used after the uploadSamples web service.
	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/datasets/addSamples/HG19_ANN/Alice
<u>Output:</u>	plain-text	Plain-text representation the word "OK".

- Create dataset: using as dataset schema the one either from an uploaded file or built as set of columns names and types. The dataset will be built empty.

<u>URI:</u>	BASE_URI/datasets/createDataSet/{dataSetName}/{username}	
<u>Interface</u>	Response createDataSet(@PathParam("username") String sc, @PathParam("dataSetName") String dataSetName, @FormParam("schema") String schema, @FormParam("names") List<String> names, @FormParam("types") List<String> types)	
<u>Method:</u>	POST	
<u>Input:</u>	dataSetName	The name of the dataset to be added to the repository.
	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/datasets/createDataSet/HG19_ANN/Alice
<u>Output:</u>	plain-text	Plain-text as "ok".

II. Query Manager Services

- Read a GMQL script file from the repository.

<u>URI:</u>	BASE_URI/query/read/{filekey}/{username}	
<u>Interface</u>	@Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON}) public Response readQuery(@PathParam("username") String sc, @PathParam("filekey") String filekey)	
<u>Method:</u>	GET	
<u>Input:</u>	filekey	It is a number that points to the file in the repository. This number is got when browsing the repository using repository web services.
	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/query/read/1/Alice
<u>Output:</u>	plain-text	Response containing the content of the file as plain text.

- Delete a GMQL query file.

<u>URI:</u>	BASE_URI/query/delete/{filekey}/{username}	
<u>Interface</u>	Response deleteQuery(@PathParam("username") String sc, @PathParam("filekey") String filekey)	
<u>Method:</u>	GET	
<u>Input:</u>	filekey	It is a number that points to the file in the repository. This number is got when browsing the repository using repository web services.
	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/query/delete/1/Alice
<u>Output:</u>	plain-text	Response containing 200 on success.

- Save a GMQL query file

<u>URI:</u>	BASE_URI/query/save/{query}/{filename}/{filekey}/{username}	
<u>Interface</u>	Response deleteQuery(@PathParam("username") String sc, @PathParam("filekey") String filekey)	
<u>Method:</u>	POST	
<u>Input:</u>	query	Plain-text containing the the script to be stored in the file.
	filename	The script file name.
	filekey	It is a number that points to the file in the repository. This number is got when browsing the repository using repository web services. When this is a new file then the filekey is empty.
	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/query/save/script+to+be+written/newFile//Alice
<u>Output:</u>	plain-text	Response containing the filekey on success.

- Run a GMQL script

<u>URI:</u>	BASE_URI/query/run/{filekey}/{execType}/{username}	
<u>Interface</u>	Response deleteQuery(@PathParam("username") String sc, @PathParam("filekey") String filekey)	
<u>Method:</u>	GET	
<u>Input:</u>	filekey	It is a number that points to the file in the repository. This number is got when browsing the repository using repository web services.
	exectype	Plain-text which can be the value of either "MapReduce" or "local". To set the execution to either map reduce or local execution.
	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/query/run/1/MAPREDUCE/Alice
<u>Output:</u>	plain-text	Response containing the jobid on success.

- Trace the execution of a GMQL job and get the job status

<u>URI:</u>	BASE_URI/query/trace/{jobid}/{username}	
<u>Interface</u>	Response traceJob(@PathParam("username") String sc, @PathParam("jobid") String jobId)	
<u>Method:</u>	GET	
<u>Input:</u>	jobid	It is a string that represents the jobid and this is returned when running the job using the previous web service.
	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/query/trace/job_1_testGMQL/Alice
<u>Output:</u>	XML/JSON	Response containing status xml on success.

- List all the job ids for a username

<u>URI:</u>	BASE_URI/query/jobs/{username}	
<u>Interface</u>	Response getJobs(@PathParam("username") String sc)	
<u>Method:</u>	GET	
<u>Input:</u>	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/query/jobs/Alice
<u>Output:</u>	XML/JSON	Response containing the xml list of jobids on success.

III. Repository File Browser Services

- Browse metadata files

<u>URI:</u>	BASE_URI/browse/meta/{username}	
<u>Interface</u>	@Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON}) public Response browseRepositoryMetadata(@PathParam("username") String sc)	
<u>Method:</u>	GET	
<u>Input:</u>	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/browse/meta/Alice
<u>Output:</u>	XML/JSON	Response containing the xml list of metafiles on success. All the files directories that have “.meta” extension will be returned, containing the files keys.

- Browse schema files

<u>URI:</u>	BASE_URI/browse/schema/{username}	
<u>Interface</u>	@Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON}) public Response browseRepositorySchemas(@PathParam("username") String sc)	
<u>Method:</u>	GET	
<u>Input:</u>	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/browse/schema/Alice
<u>Output:</u>	XML/JSON	Response containing the xml list of schema files on success. All the file directories that have “.schema” extension will be returned, containing the files keys.

- Browse query files

<u>URI:</u>	BASE_URI/browse/query/{username}	
<u>Interface</u>	@Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON}) public Response browseRepositoryQueries(@PathParam("username") String sc)	
<u>Method:</u>	GET	
<u>Input:</u>	username	Plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/browse/query/Alice
<u>Output:</u>	XML/JSON	Response containing the xml list of query files on success. All the file directories that have “.query” extension will be returned, containing the files keys.

- Browse all files

<u>URI:</u>	BASE_URI/browse/All/{username}	
<u>Interface</u>	@Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON}) public Response browseRepositoryAll(@PathParam("username") String sc)	
<u>Method:</u>	GET	
<u>Input:</u>	username	plain-text representation of the user name. {username} should be replaced with the username. Example: BASE_URI/browse/All/Alice.
<u>Output:</u>	XML/JSON	Response containing of Xml list of query files when success. All the files directories will be returned. Containing the Files keys.