WERA Data Exchange

Damir Deželjin

2016-03-23

1 General

NIB is operating a HF radar system with two set of antennas. One set of antennas monitors the Gulf of Trieste, whiles the other is directed to Gulf of Piran. Similarly, OGS from Trieste is operating its own HF radar system, which antenna range also covers the Gulf of Trieste.

Each of the mentioned antenna systems is collecting radial sea current and waves data. By combining two or more overlapping radial data measurements collected from adequately distributed antennas set, actual directional data can be computed. The later is being computed for the joined NIB and OGS HF radar network for the Gulf of Trieste.

As already mentioned, NIB and OGS HF radar systems are setup to operate together. The mentioned system is hereafter referred to as ***WERA network***, whiles each individual system is referred to as ***WERA system***.

By simplifying a bit, one can say a *WERA system* comprises of:

- Set of antennas and receiver electronics.
- *WERA station* computer connected to the mentioned electronics.
- Server computer that collects, post-process, does the computation for combining data from multiple *WERA systems*, and stores the data in a database.

Last two points from the above list can run each on its own computer or they can run on a single computer.

NIB set up a system comprising of two computers, one for each role as follows:

WERA Station role is handled by a computer named `wera-station`, which is deployed in a container in Piran, next to the Piran's church. This computer is connected to antennas electronics and is actually collecting the measurements.

WERA Server is deployed in a NIB computer room and connected to high speed network. This system periodically retrieves data from both NIB's and OGS's *WERA stations*. It stores the raw data in its database, as well it post process the data, combining both raw data sets. It also permanently stores the collected and post-processed data.

The stored data is exposed on Internet using a standard and well adopted `RESTful` interface [1]. The details are described in the next chapter.

2 RESTful WERA Data API

WERA data exchange API is published at `https://www.nib.si/mbp/apps/wera-data-exchange.rest` , and it is password protected using HTTP Basic authentication. A unique username and password was allocated to each partner to whom the WERA data access was granted.

Data exchange API documentation and a test RESTful interface is accessible at the above mentioned URL.

Following APIs are available:

Autnehtication service
Exposes common authentication endpoints. These services are usually used just once per session, i.e. logging in and acquiring a valid session key at the beginning of each set of data transfers, and logging out at the end.

Metadata service
Various static data are exposed here, that are data that doesn't change over time such as *list of all gulfs*, *list of all data types*, *list of all stations* and *list of all coordinates.*

Data service
This is used to retrieve the actual data.

2.1 Data retrieval work-flow

Important notices on API usage

Authentication

Basic HTTP authentication is used to protect the web API documentation and test site, as well the API invocations itself. Additionally, a session based authentication is used to impose per-role based access on individual API calls.

This means, the user should authenticate using Basic HTTP authentication to get access to the site in first place. Still, before executing any API

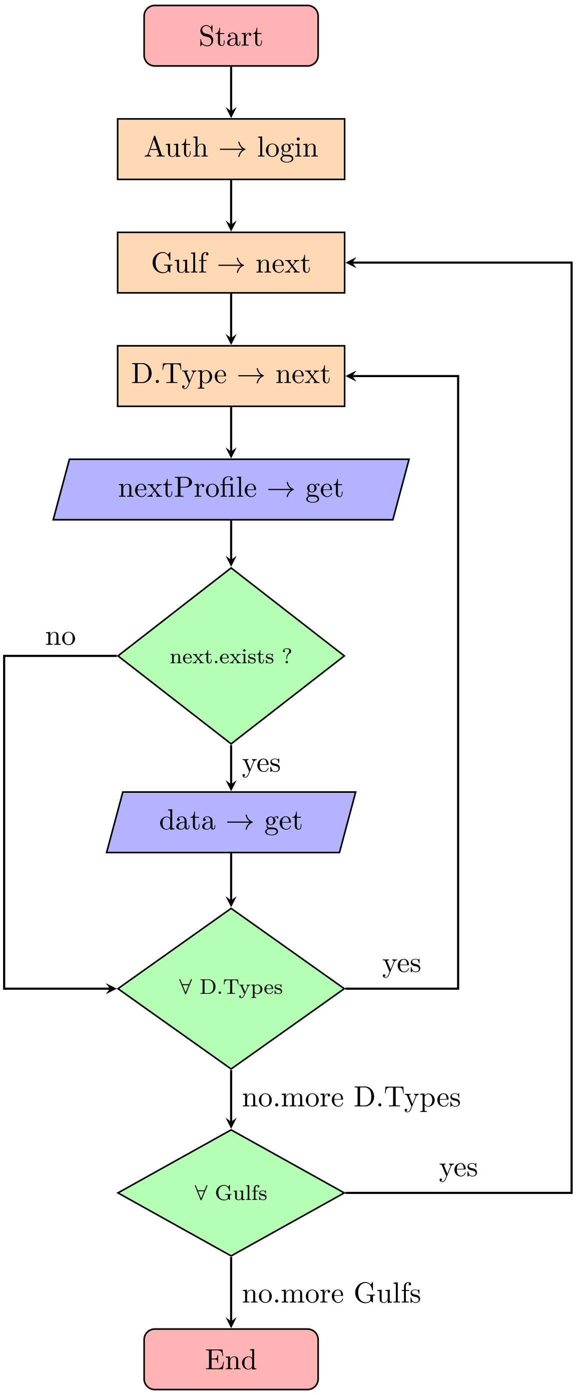

Figure 1: Data retrieval work-flow.

command, the `Authentication` → `login` method should be invoked in order to set the session cookie. The mentioned session cookie is valid for two hours within a single session. It relies on two HTTP headers which should be passed with all subsequent API invocations.

API call return status

The API comply with RESTful specification [1]. This applies to HTTP return codes as well. In general, an HTTP return code like $2xx$ means success or data exists, a return code like $4xx$ means no data exists (**note**: is the correct data being requested) or no the user has no enough permissions (**note**: did the user authenticate and is the authentication cookie carried along with the request), or the requested method doesn't exist, and HTTP codes $5xx$ in general means a server error occured.

Gulfs

WERA system is collecting data for both Gulf of Trieste and Gulf of Piran. Use the `Metadata` → `getAllGulfs` to retrieve all possible gulfs.

Data types

The API exposes all WERA data. This include *raw* (radial) and combined data. Use the `Metadata` → `getAllDataTypes` to retrieve all possible data types.

Stations

WERA network, operated by NIB, comprises of two WERA systems, i.e. NIB and OGS ones. The station parameter is relevant only to raw (radial) data and should be set accordingly to the values returned by Metadata → getAll-Stations.
IMPORTANT notice: The stations parameter should be set to 0 when retrieving combined data sets.

Data coordinate system

All WERA data are geo-location coded. This means, each single data record refers to a matrix of sampling points defined by the IxIy coordinate system - use the `Metadata` → `getAllIxIy` to retrieve the list of all matrix point coordinates.

Polling for new data

`Data` → `getNextProfile` should be used to check if new data exist. The call is lightweight and can be called frequently as e.g. once every per 10 minutes for every single data type. However, it is very important, other services aren't called if there is no need for this - so, if the data was already successfully transferred, the same data should not be retrieved once again if there is no need for it.

Metadata service

The data exposed by `Metadata` service is in general immutable. Therefore this data should be retrieved once and cached locally for a longer period. It still makes sense to occasionally validate if the locally cached data for any change.

Bibliography

[1] Wikipedia. Representational state transfer — Wikipedia, the free encyclopedia. [Online; accessed 23-March-2016; url: `https://en.wikipedia.org/wiki/Representational_state_transfer`].