Ranking list REST API

A rudimentary REST API has been implemented on the ranking list server.

The purpose is to let other sites, like a new frontend for SGP.AERO, access the ranking list data without contacting the database itself or knowledge of the DB Schema.

The advance is that the database can be changed without any impact onto other systems, as long as the API functions remain with the same calls and data return format.

The REST calls and the jSon format is widely popular at the time of writing.

The data is returned in descending order with ranked pilot no 1 as the first pilot entry.

A call with a page size of 50 will now only use 188 milliseconds to process, so this will even be suitable for a client call as well as server calls.
Returning 500 pilots will use approx. 260 milliseconds.

Example:

https://rankingdata.fai.org/rest/api/rlstandings?page_size=5&page_num=0
: returning five pilots, page 0

https://rankingdata.fai.org/rest/api/rlstandings?page_size=5&page_num=10
: returning five pilots, page 10

I.e.:
Page num: 0 to n-1
Page size: number of pilots to return

In the data you will also find the following fields:
"Pagination-Page" i.e. page number
"Pagination-Limit" i.e. page size
New in 0.12:
“Pagination-TotalPages” i.e. Total number of pages
“last_update”: Date we last updated the ranking list
"Gender": all/female/male/junior

Example

An url like https://rankingdata.fai.org/rest/api/rlstandings?page_size=1&page_num=0
Returns the following jSon formatted data:

```
{
   "status": 200,
   "status_message": "Data Found",
   "Pagination-Page:": "0",
   "Pagination-Limit:": "2",
   "data": {
      "object_name": [
         {
            "pilotid": "491",
            "surname": "Kawa",
            "firstname": "Sebastian",
            "email": "sebkaw@gmail.com",
            "nationality": "POL",
            "rankingpts": "1000.0",
            "lastcomp": "2922",
            "rankingpos": "1",
            "poschange": "0",
            "ptschange": "0.0",
            "photo": "491.JPG",
            "compname": "World Gliding Championships",
            "startdate": "2017-01-08",
            "compclass": "15m",
            "compid": "2922",
            "ovposition": "1",
            "location": "Benalla, Australia"
         },
         {
            "pilotid": "304",
            "surname": "Sommer",
            "firstname": "Michael",
            "email": "",
            "nationality": "GER",
            "rankingpts": "999.6",
            "lastcomp": "3443",
            "rankingpos": "2",
            "poschange": "0",
            "ptschange": "0.0",
            "photo": "304.jpg",
            "compname": "World Gliding Championships",
            "startdate": "2018-07-28",
            "compclass": "Open",
            "compid": "3443",
            "ovposition": "1",
            "location": "Hosin, Czech republic"
         }
      ]
   }
}
```

Changes in 0.11

Get data for one pilot knowing the Ranking list ID:
https://rankingdata.fai.org/rest/api/rlpilot?id=<RL ID>

Example:
https://rankingdata.fai.org/rest/api/rlpilot?id=2834

Response:

```
{
    "object_name": [
        {
            "pilotid": "2834",
            "surname": "Bjørnevik",
            "firstname": "Lars Rune",
            "dob": "1960-08-11",
            "email": "lars.rune.bjornevik@gmail.com",
            "homeclub": "Gardermoen SFK",
            "nationality": "NOR",
            "photo": "2834.jpg",
            "description": "",
            "quest1": "x",
            "quest2": "y",
            "quest3": "z",
            "quest4": "x",
            "sponsor": "",
            "sponsorlogo": "",
            "rankingpts": "590.1",
            "occupation": "IT Constultant",
            "lastcomp": "2798",
            "rankingpos": "2428",
            "poschange": "-180",
            "ptschange": "-34.7",
            "notify": "yes",
            "IDcompetitor": null,
            "gender": "1",
            "failicence": "56137",
            "videolink": "vid",
            "gallerylink": "photo",
            "_15m_glider": "",
            "_15m_reg": "",
            "_15m_compno": "",
            "_15m_loggersn1": "",
            "_15m_loggersn2": "",
            "_18m_glider": "",
            "_18m_reg": "",
            "_18m_compno": "",
            "_18m_loggersn1": "",
            "_18m_loggersn2": "",
            "_open_glider": "",
            "_open_reg": "",
            "_open_compno": "",
            "_open_loggersn1": "",
            "_open_loggersn2": "",
            "_std_glider": "",
            "_std_reg": "",
            "_std_compno": "",
            "_std_loggersn1": "",
            "_std_loggersn2": "",
            "_club_glider": "",
            "_club_reg": "",
            "_club_compno": "",
            "_club_loggersn1": "",
            "_club_loggersn2": "",
            "_20m_glider": "",
            "_20m_reg": "",
            "_20m_compno": "",
            "_20m_loggersn1": "",
            "_20m_loggersn2": "",
            "sponsor1_logo": "",
            "sponsor1_url": "",
            "sponsor2_logo": "",
            "sponsor2_url": "",
            "sponsor3_logo": "",
            "sponsor3_url": "",
            "deceased": "0"
        }
    ]
}
```

Get data for one pilot using parts of full name

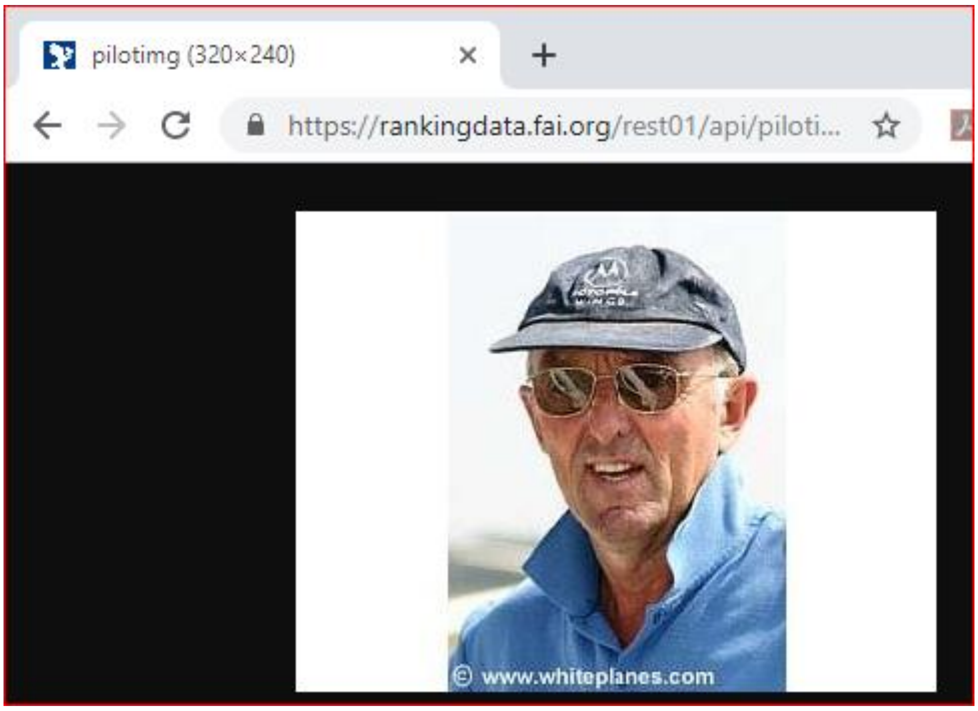
https://rankingdata.fai.org/rest/api/rlpilot?partialFullname=lars rune

Response:

```
{
    "object_name": [
        {
            "pilotid": "2834",
            "surname": "Bjørnevik",
            "firstname": "Lars Rune",
            "dob": "1960-08-11",
            "email": "lars.rune.bjornevik@gmail.com",
            "homeclub": "Gardermoen SFK",
            "nationality": "NOR",
            "photo": "2834.jpg",
            "description": "",
            "quest1": "x",
            "quest2": "y",
            "quest3": "z",
            "quest4": "x",
            "sponsor": "",
            "sponsorlogo": "",
            "rankingpts": "577.1",
            "occupation": "IT Constultant",
            "lastcomp": "2798",
            "rankingpos": "2383",
------- Cut away -----
            "sponsor3_url": ""
        }
    ]
}
```

Get the pilot photo

https://rankingdata.fai.org/rest/api/pilotimg?id=5



Changes in 0.12

- New function
  A list of news can be retived by calling:
  https://rankingdata.fai.org/rest/api/news

- The API now honors the deceased flag and does not return any pilot profiles with this flag set.
- The date of birth (dob) has been removed from the response from all calls to the API, due to privacy reasons.
- jSon response has some new fields: last_update, Genderselection, numRecords and Pagination-TotalPages

This release is only released on the new RankingData7 server, having Apache Version: Apache/2.4.38 (Debian) and PHP version: 7.3.12.

Release date March 4, 2020

Changes in 0.13

- DB routines moved to PDO from *cli
- Adding Basic Auth to REST API to allow access to admin functions
  - Get* functions are public available.
  - Set* functions need authorization like session[rl_id]
    or $_SERVER["HTTP_AUTHORIZATION"] with Basic user and password set
    Unauth calls to set functions will get "401 Unauthorized" response
    SetNewUser, setpassword, setchamp/setcomp
  - Also the functions called "GetFull*" and "Lookup*" need authorization, as they returns details as date of birth (dob) and email addresses.
- New functions
  - rlpilotfull
    - Authentication needed
    - if parameter pilot “id” set (id=xx) returns all data registered on this pilot profile, also including all competition results.
    - If parameter “partialFullname” is set, a search is done to find given input sting in a combination of firstname, surname and county name
      The data returned can be limited by using “limit”, for example “limit=10” will limit the returned set to 10 profiles matching the string combination.
  - rlpilotfull
    - Authentication needed
    - If id set:
      Returns all pilot details and comps for a given pilot id (parameter id=xx)
  - Lookupuserid
    - Authentication needed
    - A lookup function to check if a profile has a user ID defined
  - setnewuser
    - Authentication needed
    - Mandatory parameters:
      - id: The profile pilot id
      - email: Login user ID
      - pw: Password (at least 7 characters. 8a-z, A-Z, 0-9 and -_])

                - fn: Full name of the user

Release date March 21, 2020 for testing

Changes in 0.14

- New functions
    - getsystemstatus
      Returns a jSon status message on when the last DB Event schedule was executed:
      ```
      {
        "status": 200,
        "status_message": "Days since last schedule run: 0",
        "numRecords": 2,
        "data": {
          "EVENT_DEFINITION": "Call SnapshotRLstandings()",
          "LAST_EXECUTED": "2020-04-10 00:00:00"
        }
      }
      ```
    - rlCountries
        - parameter plainjson
            - 0: Returns plain jSon data
            - 1: Returns json data formatted for a select box (dropdown) list

- The rlstandings api call can how also give Historical standings. A parameter called histdate is added, and this must be a valid date in the historical standings.
    - Plain jSon format:
      /rest/api/getsnapshotdates?plainjson=1
    - Formatted for a select box (dropdown):
      /rest/api/getsnapshotdates

- Minor changes
    - Using concat to have points always displayed with one number after decimal sign

Changes in 0.15

- New functions

  setuserprofile
  Saves new or amended data in a pilot profile
  the "POST" must be a jSon array with all of these fields set:
  [{firstname:""},{surname:""},{email:""},{occupation:""},{failicence:""},{dob:""},{gender:},{homeclub:""},{nationality:""},{photo:""},{description:""},{quest1:""},{quest2:""},{quest3:""},{quest4:""},{pilotid:}]

  lookupuseremail
  Check if an email is registerd in the USERS table

  stats
  Retuns some statistics: (jSon)
  #pilots, #pilotswithpoints, #pilotswithdetalils, #Championships, #Competitions

getchampandcompstats (jSon)
Returns a list of Championships and Competitions pr year

- Parameter year

getcountryscores (jSon)
Retuns a list of countyscores for a given year

- Parameter year

Changes in 0.16

- New functions
  LookupUser
  - Parameter “id”
    ID: Ranking list ID for the user to lookup in the “USER” table (Table for login handling)

  Request must be done by a web user with privelegs groupid = 1, i.e. “Admin”

  GetRlHistoricalStandings

  Used by the table on frontpage to show historical standings in stead of current.

  - Parameter (in addition to standard) “fromdate”
    Returns the standings at “fromdate”
    Format: <YYYY-MM-DD>

  Setnewuserprofile

  - Current use must be authenticated
  - Adds a new pilot profile. Parameter is a POST of jSon encoded sting:
    ['surname' => $p.surname, 'firstname' => $p.firstname,'dob' => $p.dob,'email' => $p.email,'nationality' => $p.nationality,'gender' => $p.gender]
  - The call returns
    - 0 if unsuccessful
    - RL ID of the new pilot profile

  setNewPassword Not implemeted

  setuserprofile

  - Parameter “profiledata”
    Parameter is a POST of jSon encoded sting.
    Request must be done by a web user with privelegs groupid = 1, i.e. “Admin”

  setrlimage

  - Stores a new pilot image for current user

  Getcompinfo

  - Parameter “compid”
    Returns information about a competition

  getcompresult

    - Parameter "compid"
      Returns the results from a competition:
        - Main standings
        - Dayresults

  Setnewpassword

    - Parameter "profiledata"
      Parameter is a POST of jSon encoded sting (username, password, fullname, rl_pw, pilotidnum).
      Request must be done by a web user with privelegs groupid = 1, i.e. "Admin"

Changes in 0.17

- New functions
  Champlistforcountry
  Retives a list of chapionships for a country
    - Parameter "country"
      Tree letter abbreviation for country.
      The olympic abbreviation list is used

Changes in 0.18

- New functions
    - paypaltransactions
      Retrives a list (max 100) of transactions between dates dstart and dend
      Example:
      https://rankingdata.fai.org/rest/api/paypaltransactions?dstart=2020-01-01&dend=2020-01
    - paypaltransactiondetails
      Retrives the details of one transaction given by: trid
      Example:
      https://rankingdata.fai.org/rest/api/paypaltransactiondetails?trid=81J59780LW524235A

Changes in 0.19

- New parameter in getchampandcomps:
    - If hasresults=true then chapionships are filterd, where it's only shown the championships that has no results in one or more comps.

  This function is useful at the end of a seson when we chase organizers that has not submitted results.

  Example:
  https://rankingdata.fai.org/rest/api/getchampandcomps?year=2022&nestedjson=1&hasresults=true

Changes in 0.20

- New function
    - pilotsInCountry
        - Authentication needed
        - Parameter country=NNN where NNN=tree letter abberation based on IOC country codes (https://en.wikipedia.org/wiki/List_of_IOC_country_codes)

Changes in 0.21

- New function returning championship details and comptetitions
    - Getchampionshipcomps
        - Parameter champID
    - Cmd
        - "champ" returns info on the championship
        - "comp" returns info of the competitions in the championship
- lookupprofileemail
    - Checks if a profile has the an email address in use.
- Replaced https://rankingdata.fai.org/ with https://rankingdata.fai.org/ in documentation.

Changes in 0.22

Implementing AMS Rest API call for new championships

Authentication needed.

- New function to enable FAI AMS to add a new championship with competitons (classes)
    - setamsnewchampionship
      Creates a new championship with classes.
        - Parameter "championship": jSon array
    - On success returns array championship and comp ID's + status OK
    - Authenticated REST API Call
        - Use header "Authorization: 'Bearer xxxxxxxxxxxxxxxxxxxxxxxxx' (xxxx… is the token)
        - Token generator: http://bj-deb01/rest/CreateBearerToken.php
        - Table for tokens is igcrankings.user_rest_api_token

Changes in 0.23

Moving away from rankingdata7.fai.org to rankingdata.fai.org

"

Code listing:

File ".htaccess"

This file will rewrite the request from the REST notation to a normal PHP call

```
# Rest API settings
RewriteEngine On    # Turn on the rewriting engine
RewriteRule ^api/([0-9a-zA-Z_-]*)$ api.php?name=$1&page_size=$2&page_num=$3
[NC,L,QSA]
```

File "dbsettings.php"

Setup of the parameters for the Database

```
<?php
  // Useful DB settings
  $hostname = 'mysql';
  $port = '3306';

  //   Prod
  $DBName = 'igcrankings';
  $DBusername = 'igcrankings';
  $DBpassword = '********';

?>
```

File: api.php

The main file to handle the request

```php
<?php
// Set the jSon Header
header("Content-Type:application/json");
// Include the data processing function PHP script
require "data.php";

// paging the jSon result
// Defaults to first page with 20 entries
$pagesize = 20; // Page size in paging calls
$pagenum  = 0;  // Page number in paging calls
$pages    = 0;  // The page in paging calls

$pilotId  = 0;  // Ranking list ID
$partialFullname = ""; // Parts of pilots full name



// Get the API parameter
// I.E. /rest/api/parameter
if(!empty($_GET['name']))
{
    // Interpretate the "API" call
    $name=$_GET['name'];

    // make sure the call is lower case
    $name=strtolower($name);

    switch ($name)
    {
        case "rlstandings":
        // Get the Query parameters
        if(isset($_GET['page_size']))
        {
            $pagesize = $_GET['page_size'];
        }
        if(isset($_GET['page_num']))
        {
            $pagenum = $_GET['page_num'];
        }
        // Get data table
        $rldata=GetRlStandings($pagesize, $pagenum);
        // Get number of pages and insert to header
        $pages =  GetRlPages($pagesize);
        break;
    case "rlpilot":
        // Get the Query parameters
        // ID
        if(isset($_GET['id']))
        {
            $pilotId = $_GET['id'];
            // If the pilot ID is set and higher than 0, we'll go with that..
            $rldata = GetPilotbyID($pilotId);
        }
        // partialFullname
        if(isset($_GET['partialFullname']))
        {
            $partialFullname = $_GET['partialFullname'];
            $rldata = GetPilotByParialName($partialFullname);
        }

        // $rldata = "name is rlpilot with id $pilotId, partialFullname is $partialFullname";
        break;
    case "pilotimg":

        if(isset($_GET['id']))
        {
            $pilotId = $_GET['id'];
            if (is_numeric($pilotId))
            {
                // Some debug info:
                // echo "Pilot ID: $pilotId \n";
                // Get photo file name form the database
                $aPhoto=GetPilotPhotobyID($pilotId);
                $imgPhoto = $aPhoto[0]["photo"];

                // Get the full server path
                $imgPhotoFullPath = $_SERVER['DOCUMENT_ROOT'] . '/PilotImages/' . $imgPhoto;

                if (file_exists($imgPhotoFullPath))
                {
                    $size = getimagesize($imgPhotoFullPath);
                    $fp = fopen($imgPhotoFullPath, 'rb');
                    if ($size and $fp)
                    {
                        // Reset
                        ob_end_clean();
                        // Set Headers
                        header('Cache-Control: no-cache, no-store, max-age=0, must-revalidate');
                        header('Pragma: no-cache');
                        header("Content-Type: " .$size['mime']);
                        header('Content-Length: '.filesize($imgPhotoFullPath));
                        // Send the photo binary
                        fpassthru($fp);

                        // End transmission..
                        ob_end_clean();
                        // leave...
                        exit;
                    }
                }else
                {
                    echo "No Photo Found";
                    exit;
                }
            }
        }
        break;
    case "bar":
        $rldata = "name is bar";
        break;
    }

    if(empty($rldata))
    {
        response(200,"Data Not Found",NULL,0,0,0);
    }
    else
    {
        // gZip the response:
        // https://stackoverflow.com/questions/19043284/how-to-get-send-gzip-content-as-php-response
        ob_start("ob_gzhandler");
        response(200,"Data Found",$rldata,$pagenum,$pagesize,$pages);
        ob_end_flush();
    }

}
else
{
    // return invalid status
    response(400,"Invalid Request",NULL,0,0,0);
}

// Generic jSon Return to requestor
function response($status,$status_message,$data,$pagenum,$pagesize, $totalpages)
{
    header("HTTP/1.1 ".$status);

    // Add status
    $response['status']=$status;
    $response['status_message']=$status_message;
    // Add pagination
    if($totalpages > 0)
    {
        $response['Pagination-Page:']=$pagenum;
        $response['Pagination-Limit:']=$pagesize;
        $response['Pagination-TotalPages:']=$totalpages;
    }
    //Add the data
    $response['data']=$data;
    // Format the responce as jSon
    //$json_response = json_encode($response);
    $json_response = json_encode($data);
    echo $json_response;
}
```

File: data.php

Function(s) to communicate with the database

```php
<?php

// GetRlStandings
function GetRlStandings($pgSize, $pgNum)
{
  // Add DB functions
  include("dbcontroller.php");
  // The SQL query
  $sql = "SELECT pilot.pilotid, pilot.surname, pilot.firstname, pilot.email,
pilot.nationality, pilot.rankingpts, pilot.lastcomp, " .
  "pilot.rankingpos, pilot.poschange, pilot.ptschange, pilot.photo, " .
  "championship.compname, championship.startdate, competition.compclass, " .
  "competition.compid, compresults.ovposition, championship.location " .
  "FROM     competition INNER JOIN " .
  "  championship ON competition.champid = championship.champid INNER JOIN " .
  "   pilot ON competition.compid = pilot.lastcomp INNER JOIN " .
  "   compresults ON pilot.pilotid = compresults.pilotnum AND pilot.lastcomp =
compresults.compnum " .
  "order by pilot.rankingpos asc";

  // set limit and offset
  $off_set = $pgSize * $pgNum;
  $sql = $sql . " limit $pgSize offset $off_set";

  // Get the data
  $result = mysqli_query($con, $sql);
  // check if the query retuned any data
  if ($result === FALSE) {
    return null;
  }

  $rows = array();
  while($r = mysqli_fetch_assoc($result)) {
    $rows['object_name'][] = $r;
    }
  return $rows;
}
```

File: dbcontroller.php

Function(s) to communicate with the database

```php
<?php
  // Get db settings:
  include("dbsettings.php");

  // Connect to DB
  $con=mysqli_connect($hostname, $DBusername, $DBpassword, $DBName, $port);
  // Set charset:
  mysqli_set_charset($con, 'utf8');

  // Check connection
  if (mysqli_connect_errno())
  {
    error_log( "RL Rest view failed to connect to MySQL: " . mysqli_connect_error(),  1, "lrb@lrb.no");
    exit("RL Rest view failed to connect to MySQL");
  }
?>
```

Notes

The implementation is loosely based on:

https://shareurcodes.com/blog/creating%20a%20simple%20rest%20api%20in%20php