

RESTful service

Objectives

The aim of this project is to develop system of RESTful service.

External APIs

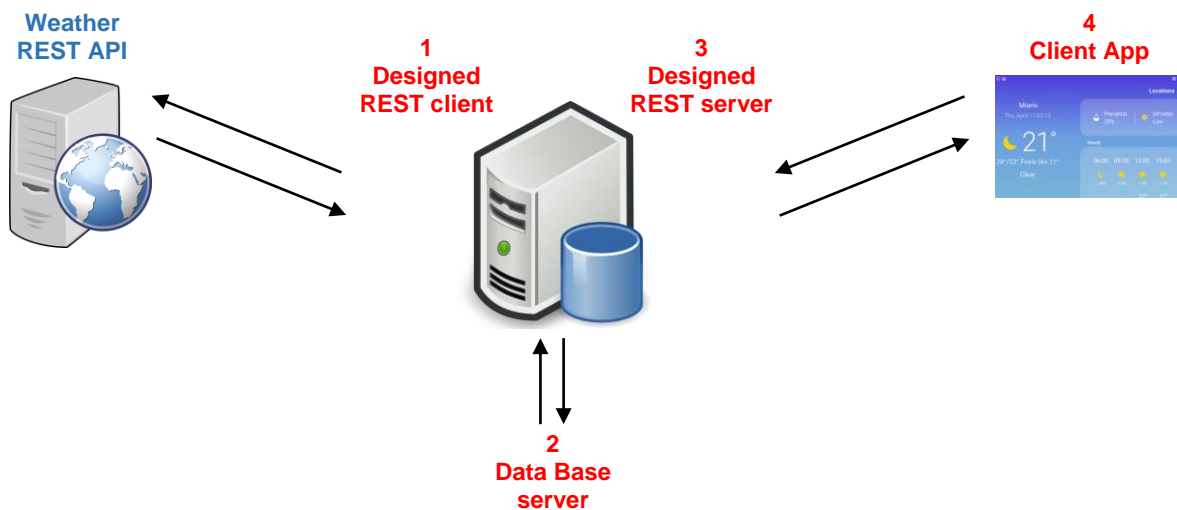
In a frame of the project there will be necessary to create an account on any of numerous weather API services (e.g. free “under some conditions” OpenWeather <https://home.openweathermap.org/>). Please read the documentation about:

- getting API key: https://home.openweathermap.org/api_keys
- pricing of free usage: <https://openweathermap.org/price#weather>
- tutorial for a quick start: <https://openweathermap.org/appid#get>
- complete API documentation: <https://openweathermap.org/api>
- requests for current wheatear: <https://openweathermap.org/current>
i.e. <http://api.openweathermap.org/data/2.5/weather?q=lodz,pl&APPID={KEY}>

Be careful about the temperature and pressure units!

Project description:

Design the system structure as depicted in the following figure:



The data flow is as follow. The Client App (4) can print to the user some data stored on Data Base (2). To separate the DB resources the Client App can communicate only through the private REST server (3) which process the client request and serve the data in a special form. The data stored in DB server (2) are taken from public REST server via the designed REST client.

To meet the project conditions implement the 4 modules with the following features:

1. Designed REST client (1):

- implement the mechanism to download from the public REST service the current weather conditions (e.g temperature in [Celcius degrees], pressure in [hPa], humidity in [%], precipitation in [mm], wind speed in [m/s] and direction in [degrees]) for 8 cities: Warszawa, Łódź, Wrocław, Szczecin, Rzeszów, Kraków, Gdańsk and Suwałki (in URL do not use polish letters),

- implement a scheduler to automatically run the previous function at least two times per day.

(NOTE!! If you are not able to leave the server to work 24h, you can run the function manually two times in a day to build a weekly data collection before designing next modules. Nevertheless, remember that the implementation of the scheduler is obligatory for higher grade).

2. Data base (2):

- run a data base server using available freeware solutions: mySQL, PostgreSQL or Heroku (in a cloud)
- design a table in your data base having the fields: ID (as primary key), date, temperature, pressure, humidity, precipitation, wind_speed, wind_direction.

3. REST server (3):

- implement REST client POST-based request processing:
 - ✓ /dbsize – to get number of days the weather conditions were collected for,
 - ✓ /average?c={city}&p={property}&d={days} - to provide an average value of the property (temperature, pressure etc.) for the given city for {days} days,
 - ✓ /poland?p={property}&d={days} – to provide an average value of the property (temperature, pressure etc.) for the whole Poland for {days} days,
- the response sent to client should be in JSON format,
- implement protection to serve only known clients; use a KEY.

4. Client App (4):

- just after running, APP (4) sends to REST server (3) set of requests to acquire date,
- next, prints the data (all average values) in a table like below

	Warszawa	Lodz	Szczecin	Wrocław	Gdańsk	Kraków	Suwałki	Rzeszów	Poland
temperature [Celsius dg]									
pressure [hPa]									
humidity [%]									
precipitation [mm]									
wind speed [m/s]									
wind direction [dg]									

Other remarks

- The project should meet all conditions mentioned above written with:
 - the black colour for assessment 3,
 - the blue colour for assessment 4,
 - the green colour for assessment 5;
 the functionality defined for assessments 4 and 5 may be considered alternate,
- The errors and exceptions handling especially for incorrect requests,
- The programming language and programming environment are arbitrary
- The readable and clean code.

Good luck!