Backend

Situation

You are a developer who is responsible for implementing backend services. This includes business logic, data access logic and service interfaces.

The services you implement will be used by your colleagues who are responsible for frontend development.

Requirements

Implement some web services for sending SMS, viewing sent SMS and aggregated statistics.
The data shall be stored in a relational database.
During send an SMS the mobile country code of the receiver needs to be identified and stored within the SMS record (list of countries and the codes you find below).
The logic for sending the SMS will be implemented later, so you need to send the message to a dummy implementation, which just dumps the SMS to a log file.
A later exchange of the dummy implementation with the real logic should not require changes on the code you wrote.

All your services should be able to return the responses in JSON and XML format. The services need to fit the interface definition at the end of this document.

If possible write and use non-blocking code.

Check and create the required database schema.

Your solution should provide a simple way to deploy the web services to Docker machine.

Write a short "How-To" which describes the deployment and initial configuration.

Write a short summary doc of your implementation (e.g. a diagram, difficulties, possible optimisations, etc.).

List of countries you should support

Name	Mobile Country Code	Country Code	Price per SMS
Germany	262	49	0.055
Austria	232	43	0.053
Poland	260	48	0.032

Technology stack at Mitto

- Microsoft .NET Core 3.1
- Microsoft Visual Studio 2019
- C# 7.3
- ServiceStack V5.1 (free starter edition limits are ok for this task)

- MySQL 5.7

Service Interface Definition

Service::SendSMS

Parameters

- from: string [the sender of the message]
- to: string [the receiver of the message]
- text: string [the text which should be sent]

Returns

- state: enum (Success, Failed)

URL format

- GET /sms/send.json?from=The+Sender&to=%2B4917421293388&text=Hello+World
- GET /sms/send.xml?from=The+Sender&to=%2B4917421293388&text=Hello+World

Service::GetCountries

Parameters

- None

Returns

- array of country ○ mcc: string [the mobile country code of the country, e.g. “262” for Germany] ○ cc: string [the country code of the country, e.g. “49” for Germany] ○ name: string [the name of the country, e.g. “Germany”]
  ○ pricePerSMS: decimal [the price per SMS sent in this country in EUR, e.g. 0.06]

URL format

- GET /countries.json
- GET /countries.xml

Service::GetSentSMS

Parameters

- dateTimeFrom: dateTime [format: “yyyy-MM-ddTHH:mm:ss”, UTC]
- dateTimeTo: dateTime [format: “yyyy-MM-ddTHH:mm:ss”, UTC]
- skip: integer [skip n records]
- take: integer [take n records]

Returns

- totalCount: int [the total count of all items matching the filter]
- items: array of SMS ○ dateTime: dateTime [the date and time the SMS was sent, format: “yyyy-MMddTHH:mm:ss, UTC]
    - mcc: string [the mobile country code of the country where the receiver of the SMS belongs to]
    - from: string [the sender of the SMS] ○ to: string [the receiver of the SMS]

    ○ price: decimal [the price of the SMS in EUR, e.g. 0.06] ○ state: enum (Success, Failed)

URL format

- GET /sms/sent.json?dateTimeFrom=2018-03-01T11:30:20&dateTimeTo=2018-0302T09:20:22&skip=100&take=50
- GET /sms/sent.xml?dateTimeFrom=2018-03-01T11:30:20&dateTimeTo=2018-0302T09:20:22&skip=100&take=50

Service::GetStatistics

Parameters

- dateFrom: date [format: “yyyy-MM-dd”]
- dateTo: date [format: “yyyy-MM-dd”]
- mccList: string list [a list of mobile country codes to filter, e.g. “262,232”] ○ if list is empty this means: include all mobile country codes

Returns

- array of statistic record ○ day: date [format: “yyyy-MM-dd”] ○ mcc: string [the mobile country code]
    - ○ pricePerSMS: decimal [the price per SMS in EUR, e.g. 0.06] ○ count: integer [the count of SMS on the day and mcc]
    - ○ totalPrice: decimal [the total price of all SMS on the day and mcc in EUR, e.g. 23.64]

URL format

- GET /statistics.json?dateFrom=2018-03-01&dateTo=2018-03-05&mccList=262,232
- GET /statistics.xml?dateFrom=2018-03-01&dateTo=2018-03-05&mccList=262,232