

Interview Task - .NET Developer

Introduction

We want you to create a sample application which is familiar to things we are actually doing, but also allows us to answer a few questions:

- How do you work?
- What skill do you have?
- What good and bad practices are you following?
- How do you solve the problems you encounter?

Getting started

The store manager noticed that in some stores, customers were waiting in the checkout line for a long time. To solve this problem, the manager decided to start collecting statistics about people visiting his store. Therefore, each entry to the store is noted. Therefore, each entry to the store is recorded to optimize the operation plan of individual stores.

In order to implement these plans, the manager decided to prepare a system for collecting and visualizing the aforementioned information.

Task specification

- Backend:
 - Create ASP.NET application exposing REST API,
 - .NET Framework 4.6.1 or newer required,
 - Build endpoints using the specifications below,
 - Use the **Entity Framework** for data storage.
- Frontend:
 - Create application using Front-End Web Technologies,
 - The collected data should be presented using graph library with an appropriate license,
 - The selected date range must be defaulted, but also configurable by the end user,
 - Only modern browsers need to be supported.

Store request

Add store

Sample request

```
curl --location --request POST 'https://localhost/api/store' \  
  --header 'Content-Type: application/json' \  
  --data-raw '{ "name": "Store #1", "city": "Castle Douglas",  
"country": "Scotland" }'
```

Node	Description	Type
name	Store name	string
city	City	string
country	Country	string

Sample response

```
{
  "id": 1,
  "status": true,
  "message": null
}
```

Node	Description	Type
id	Store id	integer
status	Action status	boolean
message	Error message	string

Get store

Sample request

```
curl --location 'https://localhost/api/store/{id}'
```

Node	Description	Type
id	Store id	integer

Sample response

```
{
  "id": 1,
  "name": "Store #1",
  "city": "Castle Douglas",
  "country": "Scotland",
  "status": true,
  "message": null
}
```

Node	Description	Type
id	Store id	integer
name	Store name	string
city	City	string
country	Country	string
status	Action status	boolean
message	Error message	string

Update store

Sample request

```
curl --location --request PUT 'https://localhost/api/store/{id}' \
  --header 'Content-Type: application/json' \
  --data '{ "name": "Store #1", "city": "Castle Douglas", "country":
"Scotland" }'
```

Node	Description	Type
id	Store id	integer
name	Store name	string
city	City	string
country	Country	string

Sample response

```
{
  "id": 1,
  "status": true,
  "message": null
}
```

Node	Description	Type
id	Store id	integer
status	Action status	boolean
message	Error message	string

Delete store request

Sample request

```
curl --location --request DELETE 'https://localhost/api/store/{id}'
```

Node	Description	Type
id	Store id	integer

Sample response

```
{
  "status": true,
  "message": null
}
```

Node	Description	Type
status	Action status	boolean
message	Error message	string

Entry requests

Add entry

Sample request

```
curl --location --request POST 'https://localhost/api/store/entry' \
  --header 'Content-Type: application/json' \
  --data-raw '{ "storeId": 1, "entryDate": "2021-01-01T08:00:00.222Z" }'
```

Node	Description	Type
storeId	Store id	integer
entryDate	Entry date	datetime in yyyy-MM-ddTHH:mm:ssZformat

Sample response

```
{
  "status": true,
  "message": null
}
```

Node	Description	Type
status	Action status	boolean
message	Error message	string

Statistics requests

Get statistics

Sample request

```
curl --location 'https://localhost/api/store/statistics/{storeId}?
startDate={yyyy-MM-dd}&endDate={yyyy-MM-dd}'
```

Node	Description	Type
storeId	Store id	integer
startDate	Start range date	date in yyyy-MM-ddformat
endDate	End range date	date in yyyy-MM-ddformat

Sample response message

```
{
  "id": 1,
  "name": "Store #1",
  "city": "Castle Douglas",
  "country": "Scotland",
  "statistics": [
    {
      "date": "2021-01-01",
      "count": 1543
    },
    {
      "date": "2021-01-02",
      "count": 2531
    },
    {
      "date": "2021-01-03",
      "count": 1725
    }
  ],
  "status": true,
  "message": null
}
```

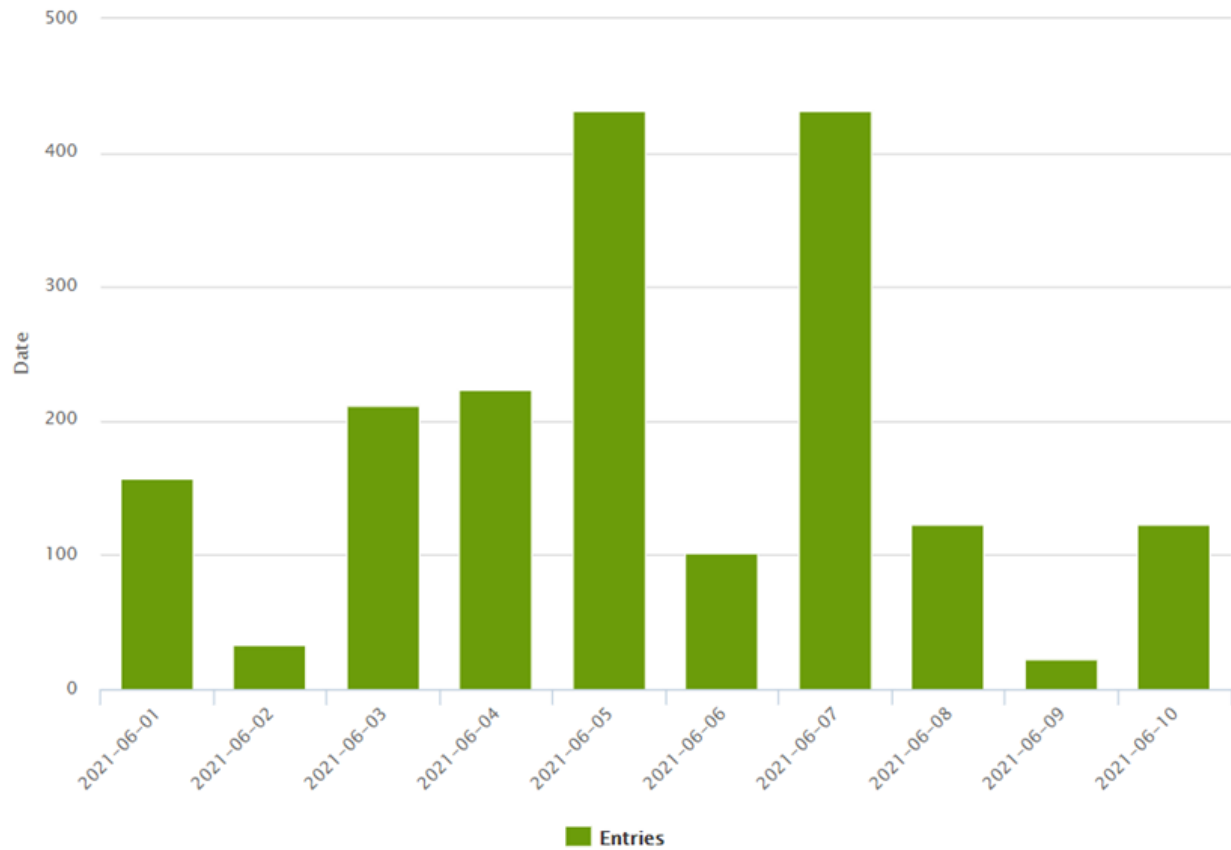
Node	Description	Type
id	Store id	integer
name	Store name	string
city	City	string
country	Country	string
date	Entries for date	date in yyyy-MM-ddformat
count	Entries count	integer
status	Action status	boolean

message	Error message	string
---------	---------------	--------

Quick overview

From: 01/06/2021 To: 10/06/2021

Entries summary



Extras

- If you like unit tests, feel free to show us your favorite libraries!
- We are big fans of **Unity** (dependency injection)!
- We really love the features **Query** offers.
- If you do not have access to a specific endpoint, create and use a sample JSON file.
- If you think this project needs improvement, do so and document it!