90322:
”SOLID Programming in C#”

Lab Manual

Wincubate ApS

11-02-2022

V1.0

Table of Contents

Prerequisites

The present labs require the course files accompanying the course to be extracted in some directory path, e.g.

```
C:\Wincubate\90322
```

with Visual Studio 2019 (or later) installed on the PC.

We will henceforth refer to the chosen installation path containing the lab files as *PathToCourseFiles* .

Module 1: "The SOLID Principles in C#"

Lab 1.1: "Write to both Console and File"

This step emulates **Sprint 1** after having done the initial application in the presentation.

- Open the starter project in
  PathToCourseFiles`\Labs\1.1\Starter` ,
  which contains the Visual Studio solution we produced in the presentation.

The story described on the Sprint 1 board reads:

"Allow writing results to both console storage and file storage"

- Implement the story in a SOLID manner.

Lab 1.2: "Send an SMS"

This step emulates Sprint 2 after having done the initial application in the presentation.

- Open the starter project in
  PathToCourseFiles\Lab\1.2\Starter ,
  which contains the Visual Studio solution we produced in the presentation.

The story described on the Sprint 2 board reads:

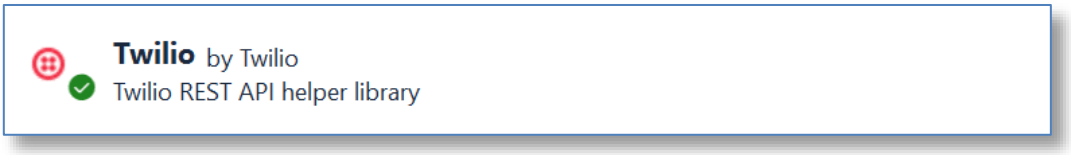
> ***"Implement functionality for sending an SMS with the result"***

- Implement the story in a SOLID manner
  - Create a SMS transmission strategy using the Twilio SMS API.
  - Test your implementation by sending yourself the results as SMS messages.
    - It may take up to 30 to 45 seconds for the SMS to arrive.
    - Note: Since Twilio is an international service, do remember to prefix your number with "+45"

Background

See the Twilio SMS API documentation at https://www.twilio.com/docs/api/messaging/send-messages .

In order to use the Twilio API for sending SMS messages you must include their nuget package into your project:



Once it is included, you can send an SMS message using the following code:

```
string _accountSid = "ACa5?64844f11c4152c5e4db4bc202c7??";
string _authToken = "b978????5570945b775bac117f5b7059";

TwilioClient.Init(_accountSid, _authToken);
MessageResource mr = await MessageResource.CreateAsync(
    to: new PhoneNumber("<phone number>"),
    from: new PhoneNumber("+4676???9439"),
    body: "<contents of SMS>"
);
```

Your instructor will supply you with the remaining digits substituting the missing ?'s in the above codes.

Lab 1.3: "Retry the SMS sends"

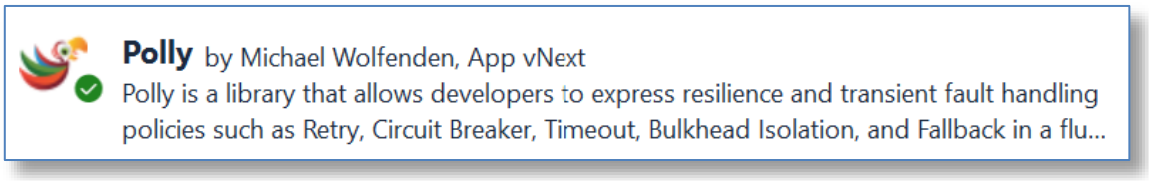
This step emulates Sprint 3 after having done the initial application in the presentation.

- Open the starter project in
  PathToCourseFiles`\Lab\1.3\Starter` ,
  which contains the Visual Studio solution we produced in the presentation.

The story described on the Sprint 3 board reads:

> ***"Implement a retry strategy for SMS sends"***

Many such strategies can be implemented in a very simple fashion using the Polly nuget package:



You can read more about it here: https://www.nuget.org/packages/Polly/

Your task is now:

- Implement the story in a SOLID manner
  - Do whichever strategy you find correct, e.g. *"retry 3 times and then fail"* or similar.

Lab 1.4: "Class Discussion"

When you have completed Labs 1.1, 1.2, and 1.3, your instructor will initiate a discussion in the classes addressing such issues as:

- How much additional work would it take to implement a potential Sprint 4 specification such as

  "Write results to files, log to Console, and send an SMS!
  And hey… Let's keep trying all until they all succeed…"

- What if we need to send an email?
- What if we want to retry sending the email?
- What if we wanted to have multiple read storages? Same or different serialization types?
- …
- What about unit testing?
- …
- In which order did labs 1.1, 1.2, and 1.3 need to be completed?
- …
- Agile vs. SOLID in general?