logo

WideBot internship task release notes

Ahmed Shaban Helwa

ahmedshabaan@outlook.com

**M E T A**

|  |  |
| --- | --- |
| Title: | WideBot Internship task release notes |
| Revision: | 1.3.1 |
| Date: | 18/8/2020 |

**Resources:**

**Task Assignment**

*https://drive.google.com/file/d/1BINgfwESr9naRQTPZYr4SIidYkNJEdvg/view*

**Task** **repository**

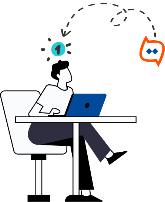
*https://github.com/AhmedShabaanHelwa/Task*

**Old repository**

*https://github.com/AhmedShabaanHelwa/Widebot-Task*

**Open Defects:**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Title | State | Phase |
| 1 | Null in JSON response | Recipient object in API response JSON returns null | Mockup 3 – Continues Integration |



|  |
| --- |
| **Task Analysis** |

**Statement**

*" Create ASP.NET Core API that integrate with bellow sample API and return response with Facebook send message API response to bot "*

That are mapped to the following

**Mocks**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Statement | Milestone | Details |
| *1* | *Create ASP.NET Core API* | Mockup 0 | Constructing the basic structure of an ASP.NET Core API project. |
| *2* | *that integrate with bellow sample API return response* | Mockup 1 | Integrating with a remote API and retrieve data in some bare form, and test it against API client. |
| *3* | *return response with facebook send message API response* | Mockup 2 | Mapping the response of remote API to Facebook Messenger response format. |
| *4* | *to bot* | Mockup 3 | Integrating with WideBot |

**SW Architecture**

inspired by *"the Mediator* " Design Pattern.

1 - HTTP GET Request

3 - HTTP POST Request in

Messenger Format

A Mediator API

2 - JSON Response

4 - WideBot response

(Expected)

**Solution Architecture**

Single project of Type: ASP.NET Core Web Application, scaffolded for Web API template.

|  |  |
| --- | --- |
| **Component** | **Functionality Description** |
| Controllers | Represents the **C**ontrollers for MVC Pattern |
| Models | Represents the internal domain entities as the **M**odel for MVC Pattern |
| DTOs | Represents the external domain, in replacement of **V**iew for MVC Pattern. |
| Profiles | Provides the mapping profiles to the Automapper, in order to map between Actual responses and The Data Transfer Objects that should be passed as response JSON payload. |

**Controllers**

Name:

* Integrator

Class code:

* IntegratorController.cs

Base Abstract Class:

* Microsoft.AspNetCore.Mvc.ControllerBase, it provides controller functionality without view support

API Operations:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| URL | HTTP | CRUD | Description | Success | Failure |
| api/Integrator | GET | READ | Getting all resources | OK - 200 | Not Found - 404 |
| api/Integrator/{id} | GET | READ | Getting single resource by ID | OK - 200 | Not Found - 404 |

Dependency Injection:

|  |  |  |
| --- | --- | --- |
| Dependency | Version | NuGet Package |
| AutoMapper | 8.0.1 | Extensions.Microsoft.DependencyInjection |
| JSON | 12.0.3 | Newtonsoft.Json |

Private methods:

//!Pseudo code

private UserDto ReadData(int id)

{

//!Pseudo: 1 – Inistantiate local variables to open an HttpClient and JSON payload

//!Pseudo: 2 – Read from API as HttpClient, retrieve the JSON raw

//!Pseudo: 3 – Parse the JSON raw to seriallize the string into JSON object.

//!Psedo: 4 – Deserialize the json to a Response DTO object

return user of UserDto;

}

**Models**

* Contains the internal domain of the entities.

Class code:

* MessengerTemplate.cs

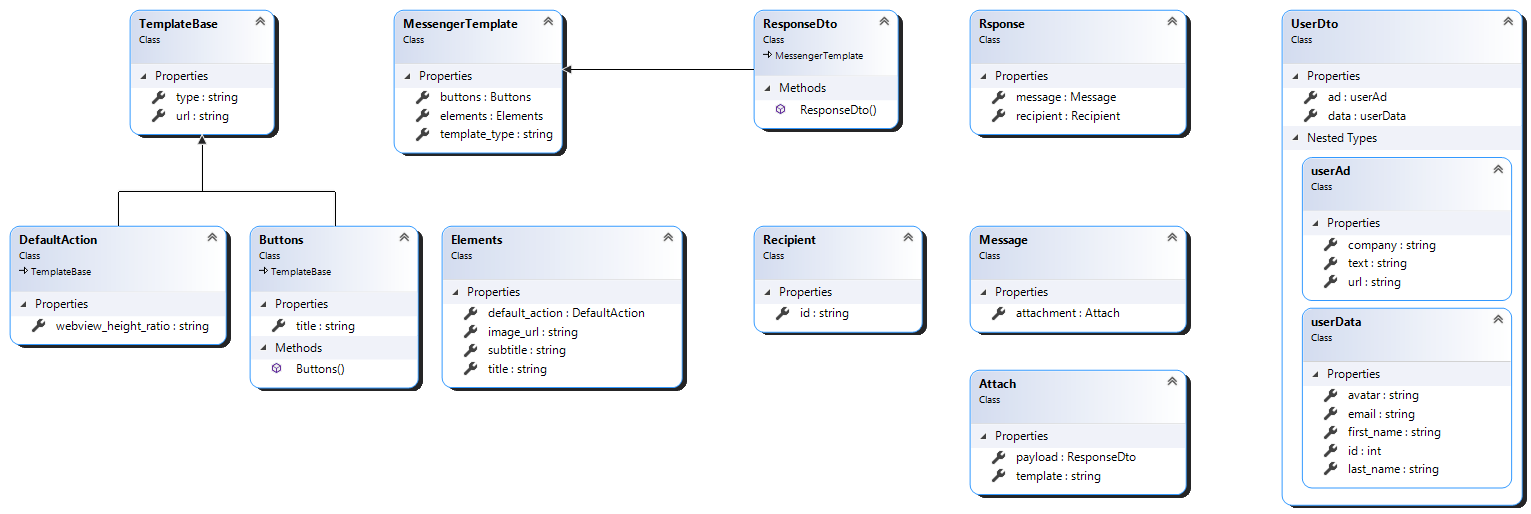
**DTOs**

* Contains the external domain of the entities.

Class Code:

* UserDto.cs
* ResponseDto.cs

*Class diagram for the entities:*



**Profiles**

Class code:

* IntegratorProfile.cs

**Startup Class**

|  |  |
| --- | --- |
| Method | Registeration |
| Service Configurations | AutoMapper |

|  |  |
| --- | --- |
| **Deployment** |  |



**Heroku Cloud hosting**

* Heroku is a cloud platform as a service (PaaS) supporting several programming languages. e.g; Java, Node.js, Scala, Clojure, Python, PHP, and Go.
* It provides an option for hosting containerized apps as well; which is the facility used to host our app: The Integrator API.

Application url:

*https://a7mah1.herokuapp.com/api/integrator/{id}*

****

**Docker**

A publishing version of the API is pushed into a linux container of Docker.

Currently deployed Container:

* Version 1.00

Resources

* Deploy ASP.Net Core to Heroku for free using Docker - Medium   
  *https://medium.com/@vnqmai.hcmue/deploy-asp-net-core-to-heroku-for-free-using-docker-bd6d6fc161ae*
* Deploy Asp.Net Core Website on Heroku using Docker - YouTube *https://youtu.be/gQMT4al2Grg*

|  |  |
| --- | --- |
| **Integration with WideBot** |  |

Facebook test page:

*https://www.facebook.com/%D8%A3%D8%AD%D9%85%D8%A7%D9%87-226060455405852/*

The bot should accept response in Messenger response generic format as the following example:

{

  "recipient":{

    "id":"<PSID>"

  },

  "message":{

    "attachment":{

      "type":"template",

      "payload":{

        "template\_type":"generic",

        "elements":[

           {

            "title":"Welcome!",

            "image\_url":"https://petersfancybrownhats.com/company\_image.png",

            "subtitle":"We have the right hat for everyone.",

            "default\_action": {

              "type": "web\_url",

              "url": "https://petersfancybrownhats.com/view?item=103",

              "webview\_height\_ratio": "tall",

            },

            "buttons":[

              {

                "type":"web\_url",

                "url":"https://petersfancybrownhats.com",

                "title":"View Website"

              },{

                "type":"postback",

                "title":"Start Chatting",

                "payload":"DEVELOPER\_DEFINED\_PAYLOAD"

              }

            ]

          }

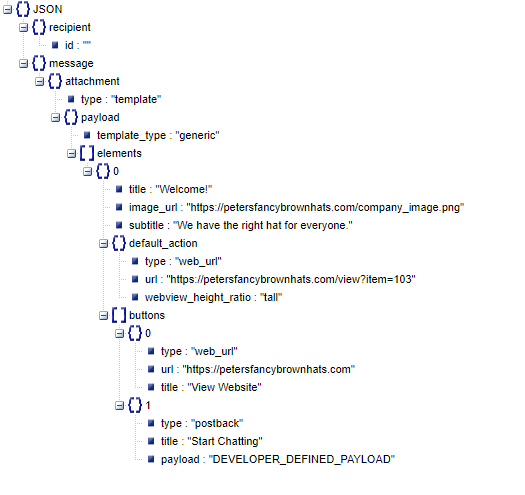
        ]

      }

    }

  }

}

That yield the JSON:  


**Appendix**

**Document History**

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 1.0 | 4/8/2020 | Initial release |
| 1.0.1 | 5/8/202 | More Formatting |
| 1.1 | 5/8/2020 | Solution Architecture  Task Analysis |
| 1.2 | 18/8/2020 |  |
| 1.3 | 19/8/2020 | Add Facebook Messenger response |
| 1.3.1 | 19/8/2020 | Add defect #1 |