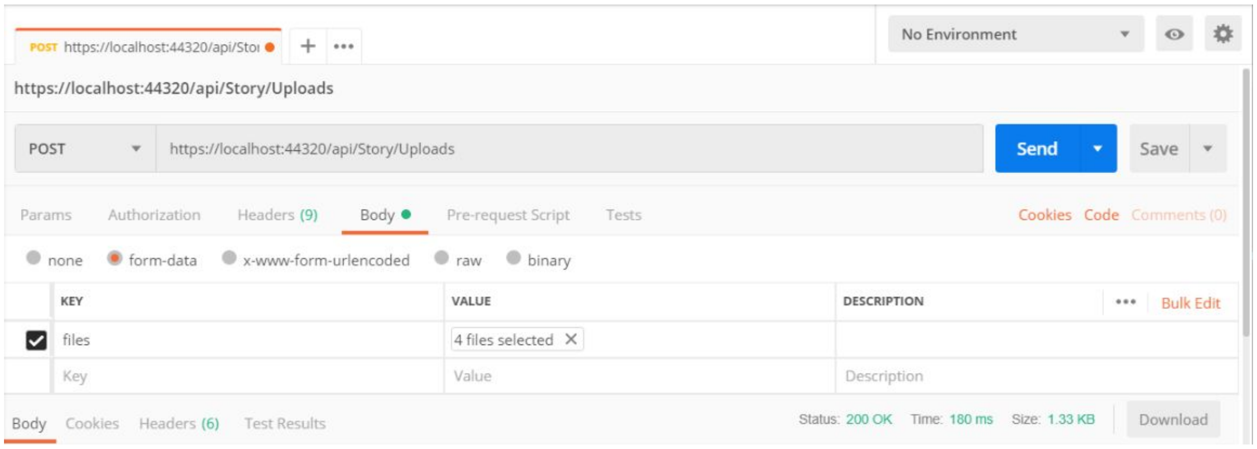
Lab 14. Final Project Lab

Task

Integrating Website, Webservice, and Webservice Client.

Solution

1. Add a WebApi and a WpfApp project to current Asp.Net Core MVC solution.
2. In the WebApi project, implement CRUD methods for the Stories Model and the related StoryBlocks Model, so that by running this project, we will have the ability to get, create, edit, and delete any of the story blocks or stories in our database. As a story contains multiple blocks, the we have to create a story as container before we can add blocks to it, and when we delete a story, all of its blocks will be deleted as well.
3. By typing the corresponding api url in the address bar or using postman, we can access our database with api.
4. So far, I had implemented basic Web APIs for CRUD and upload images to the wwwroot folder in the MVC project as well as calling the Get Api from the WPF client and will be working on the other APIs later.
5. Upload Image(s) to the custom folder with Web Api: upload 4 images to the folder "wwwroot/Stories" in the MVC project.

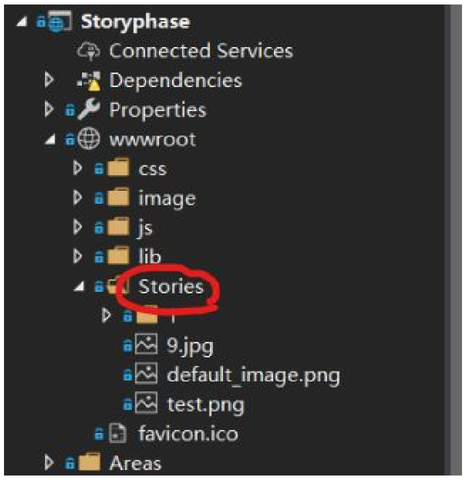

https://localhost:44320/api/Story/Uploads

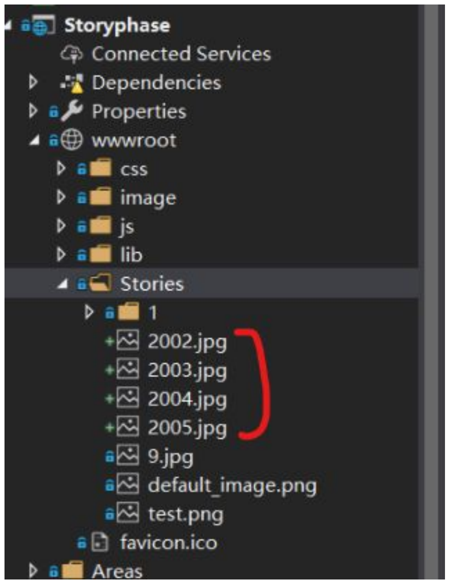
POST https://localhost:44320/api/Story/Uploads

Body: form-data

KEY	VALUE	DESCRIPTION
✓ files	4 files selected	
Key	Value	Description

Status: 200 OK Time: 180 ms Size: 1.33 KB

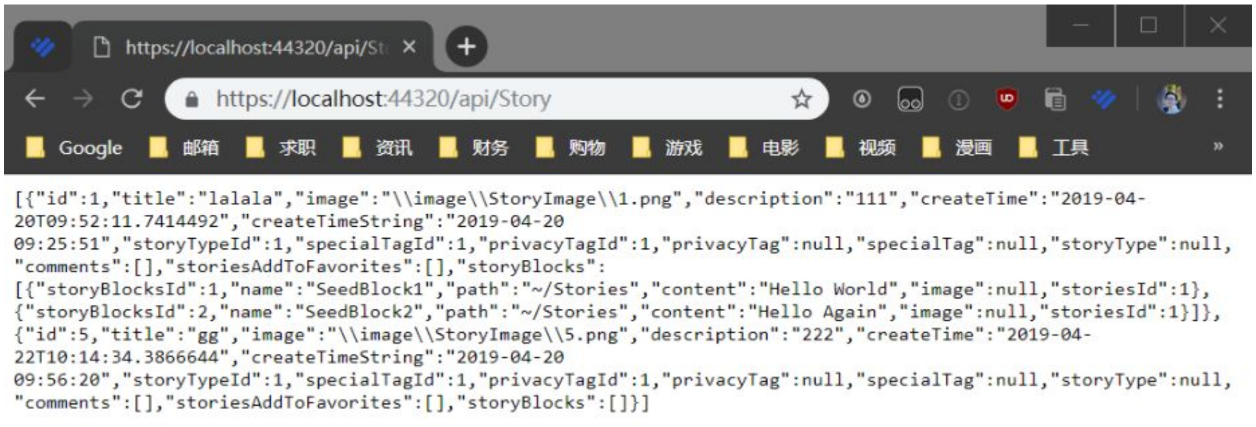




(Before) (After)

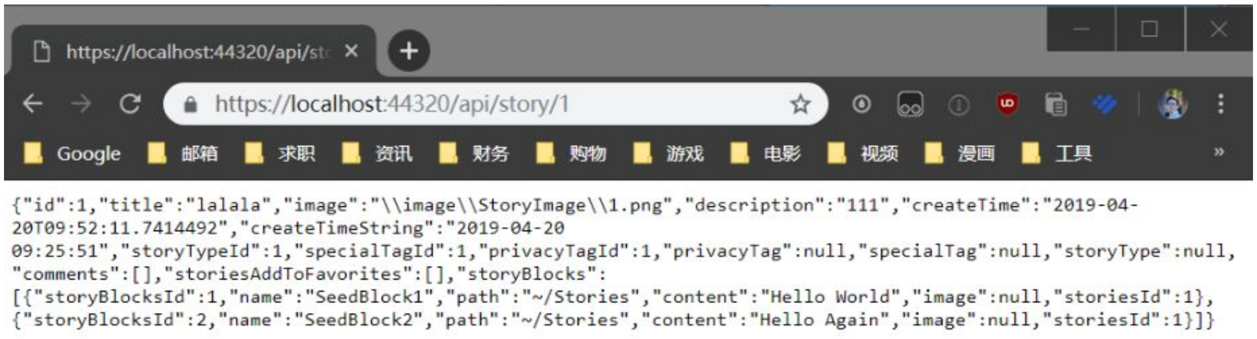
6. Take the GET APIs for Stories Model as an example:

URL: https://localhost:44320/api/Story (Get full story list, 2 records are retrieved)


https://localhost:44320/api/St
https://localhost:44320/api/Story
Google
邮箱
求职
资讯
财务
购物
游戏
电影
视频
漫画
工具

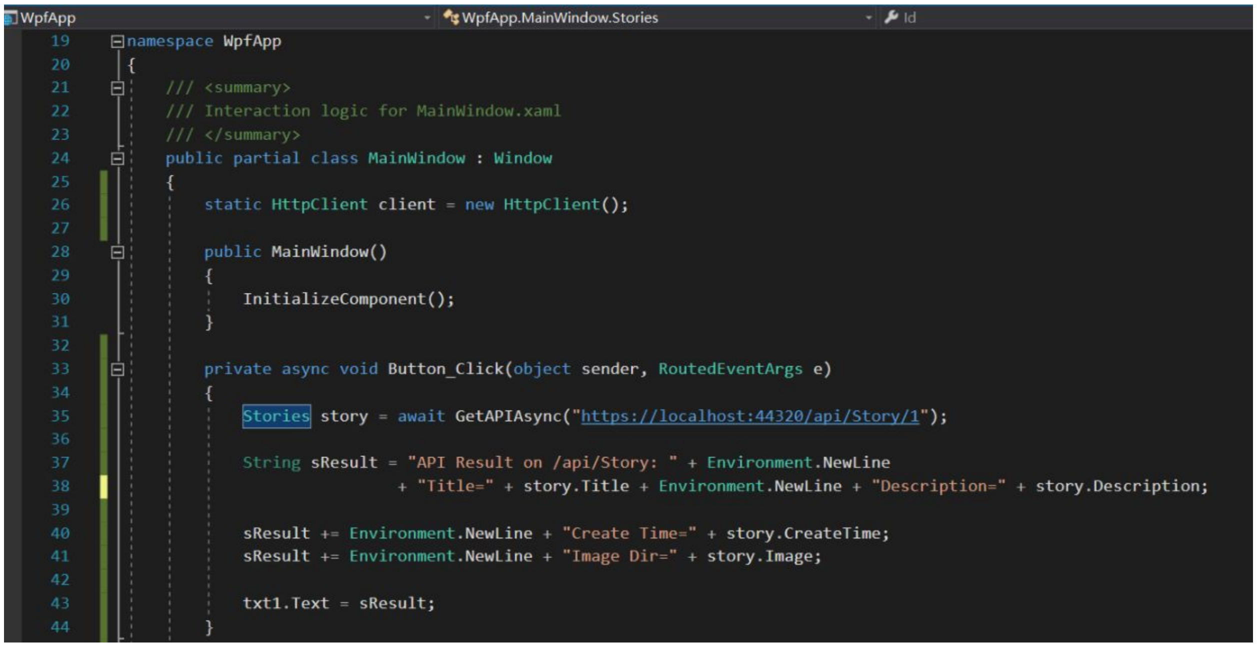
```
[{"id":1,"title":"lalala","image":"\\image\\StoryImage\\1.png","description":"111","createTime":"2019-04-
20T09:52:11.7414492","createTimeString":"2019-04-20
09:25:51","storyTypeId":1,"specialTagId":1,"privacyTagId":1,"privacyTag":null,"specialTag":null,"storyType":null,
"comments":[],"storiesAddToFavorites":[],"storyBlocks":
[{"storyBlocksId":1,"name":"SeedBlock1","path":"~/Stories","content":"Hello World","image":null,"storiesId":1},
{"storyBlocksId":2,"name":"SeedBlock2","path":"~/Stories","content":"Hello Again","image":null,"storiesId":1}]},
{"id":5,"title":"gg","image":"\\image\\StoryImage\\5.png","description":"222","createTime":"2019-04-
22T10:14:34.3866644","createTimeString":"2019-04-20
09:56:20","storyTypeId":1,"specialTagId":1,"privacyTagId":1,"privacyTag":null,"specialTag":null,"storyType":null,
"comments":[],"storiesAddToFavorites":[],"storyBlocks":[]}]
```

URL: https://localhost:44320/api/Story/1 (Get the Story with id 1)

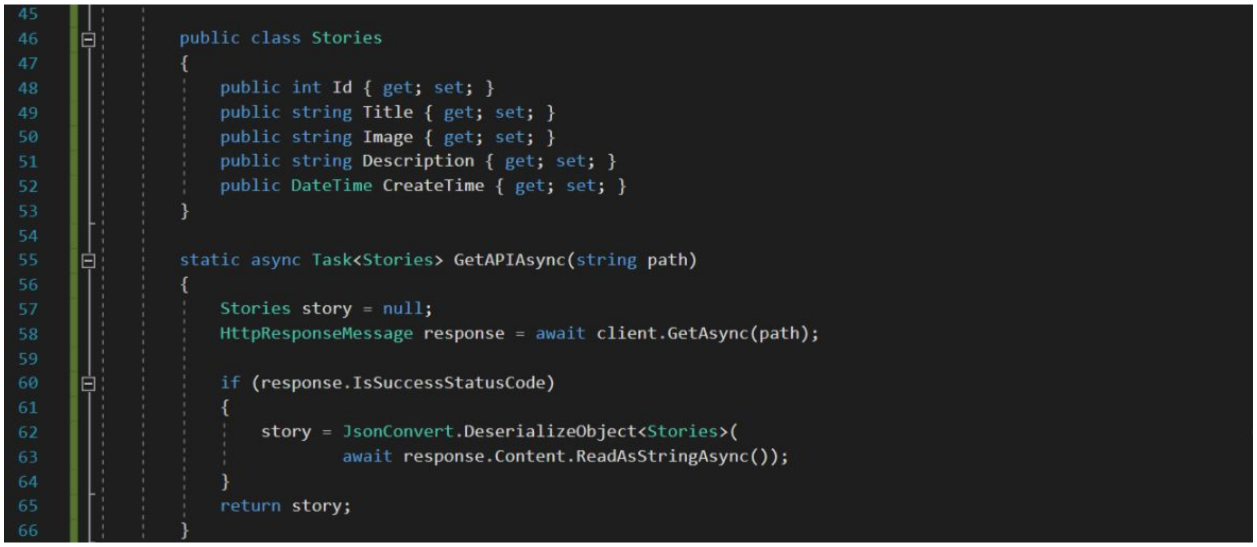

https://localhost:44320/api/st
https://localhost:44320/api/story/1
Google
邮箱
求职
资讯
财务
购物
游戏
电影
视频
漫画
工具

```
{"id":1,"title":"lalala","image":"\\image\\StoryImage\\1.png","description":"111","createTime":"2019-04-
20T09:52:11.7414492","createTimeString":"2019-04-20
09:25:51","storyTypeId":1,"specialTagId":1,"privacyTagId":1,"privacyTag":null,"specialTag":null,"storyType":null,
"comments":[],"storiesAddToFavorites":[],"storyBlocks":
[{"storyBlocksId":1,"name":"SeedBlock1","path":"~/Stories","content":"Hello World","image":null,"storiesId":1},
{"storyBlocksId":2,"name":"SeedBlock2","path":"~/Stories","content":"Hello Again","image":null,"storiesId":1}]}
```

7. After implemented APIs, we will now develop a WPF client project to consume the APIs in WebApi Project. As a quick example, I have added a button and a TextBlock to the main window and tested the Get api to display some of the attributes of story with id 1.

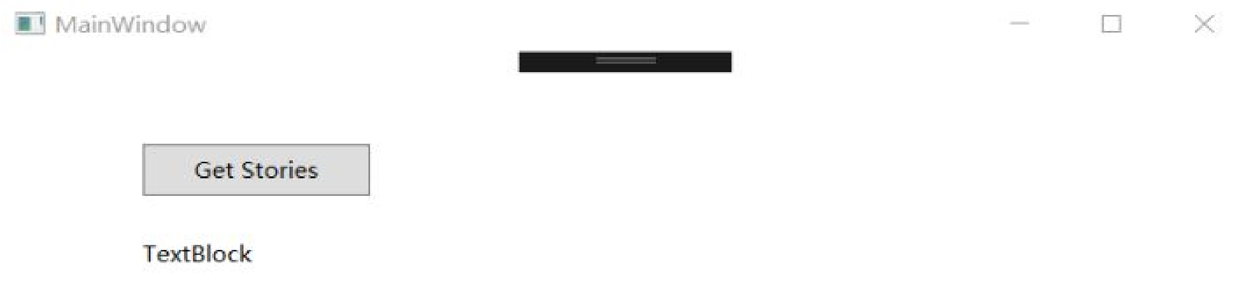


```
namespace WpfApp
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        static HttpClient client = new HttpClient();

        public MainWindow()
        {
            InitializeComponent();
        }

        private async void Button_Click(object sender, RoutedEventArgs e)
        {
            Stories story = await GetAPIAsync("https://localhost:44320/api/Story/1");

            String sResult = "API Result on /api/Story: " + Environment.NewLine
                            + "Title=" + story.Title + Environment.NewLine + "Description=" + story.Description;

            sResult += Environment.NewLine + "Create Time=" + story.CreateTime;
            sResult += Environment.NewLine + "Image Dir=" + story.Image;

            txt1.Text = sResult;
        }
```

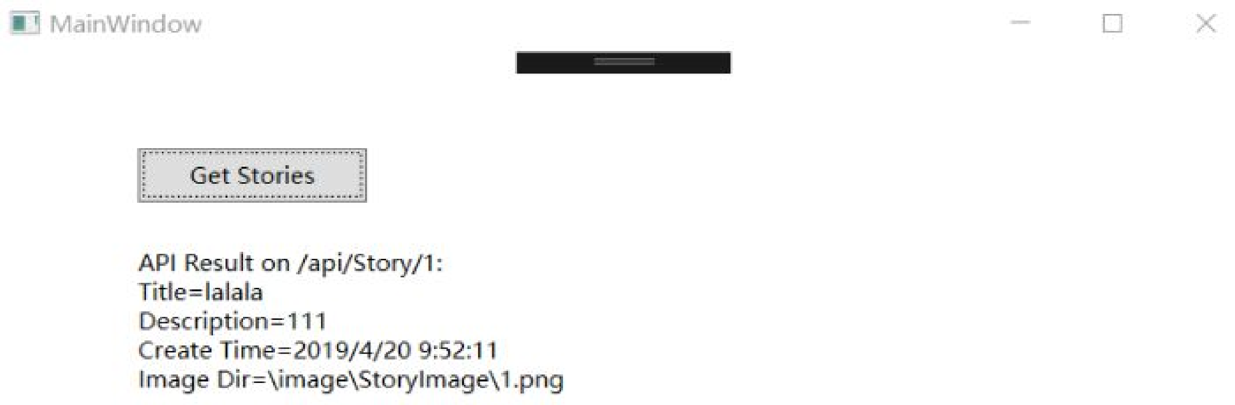


```

        public class Stories
        {
            public int Id { get; set; }
            public string Title { get; set; }
            public string Image { get; set; }
            public string Description { get; set; }
            public DateTime CreateTime { get; set; }
        }

        static async Task<Stories> GetAPIAsync(string path)
        {
            Stories story = null;
            HttpResponseMessage response = await client.GetAsync(path);

            if (response.IsSuccessStatusCode)
            {
                story = JsonConvert.DeserializeObject<Stories>(
                        await response.Content.ReadAsStringAsync());
            }
            return story;
        }
```

8. Run the WebApi Project first and then WpfApp. Before clicking the button:



When the window shows up, click on "Get Stories" button and the following result will be displayed in the TextBlock:



```
API Result on /api/Story/1:
Title=lalala
Description=111
Create Time=2019/4/20 9:52:11
Image Dir=\image\StoryImage\1.png
```