

Assignment

For Implementation Engineer position

1. Assuming you have the following two requests that were sent to your server by a client:

```
{
  {
    {
      //a

      internal class Program
      {
        private static void Main(string[] args)
        {
          {
            "Age": 25,
            "Height": 170,
            "Weight": 80,
            "Name": {
              "First": "Charles",
              "Last": ""
            },
            "HairColour": "Brown",
            "Address": {
              "City": "Kansas city",
              "Street": "Blue street",
              "House": "547"
            },
            "IsEmployed": global::System.Boolean v = true;
          }
        }
      }
    }
  }
}
```

```
//B)
```

```

internal class Program
{
    private static void Main(string[] args)
    {
        {
            "Age": "Twenty Five",
            "Height": 170,
            "Weight": 80,
            "Name": {
                "First": "Charles",
                "Last": "TheMock"
            },
            "HairColour": "Brown",
            "Address": {
                "City": "Kansas city",
                "Street": global::System.String v = "Blue street"; global::System.String
v1 = ""House""; "547"
            },
            "MaritalStatus": "Single",
            "IsEmployed": true
        }
    }
}

```

That were sent to the following controller:

```

{
    {
        using Microsoft.AspNetCore.Mvc;
        using System;
        using System.Collections.Generic;
        using System.Linq;
        using System.Threading.Tasks;

        namespace Professional.Services.Assignment
        {
            public class AddCandidateController : Controller
            {
                [HttpPost]
            }
        }
    }
}

```

```

public IActionResult Index([FromBody] Candidate candidate)
{
    try
    {
        SaveCandidate(candidate);
        return Ok();
    }
    catch (Exception ex)
    {
        return StatusCode(500, ex);
    }
}

private void SaveCandidate(Candidate candidate)
{
    var firstName = candidate.Name.First.Trim();
    var lastName = candidate.Name.Last.Trim();
    var age = candidate.Age;
    var height = candidate.Height;
    var weight = candidate.Weight;
    var hairColour = candidate.HairColour.Trim();
    var maritalStatus = candidate.MaritalStatus.Trim();
    var isEmployed = candidate.IsEmployed;
    var city = candidate.Address.City.Trim();
    var street = candidate.Address.Street.Trim();
    var house = candidate.Address.House.Trim(); 37.
}
}

```

```

public class Candidate
{
    public Name Name { get; set; }
    public int Age { get; set; }
    public int Height { get; set; }
    public int Weight { get; set; }
    public string HairColour { get; set; }
    public Address Address { get; set; }
    public string MaritalStatus { get; set; }
    public bool IsEmployed { get; set; }
}

```

```

public class Name
{

```

```
        public string First { get; set; }
        public string Last { get; set; }
    }

    public record Address(
        string City, string Street, string House);
    }
}
```

And the following responses were received:

- A) 500 object was null, Professional.Services.Assignment.cs Line 32 B)
400 Bad request

Please write a clear and cordial email to the client explaining what is wrong with their requests and what they should do to fix it.

Hi <Customer Name>!

Sorry to hear that you are running into some trouble with <the product>. Other than that, I hope all is well. I did have a chance to review the email along with the errors you provided me. From my general knowledge of the description of the error, I am pretty sure there is something missing and I am certain we can get this resolved for you quickly.

I used some debugging tools to try to clean up or identify what the issue is. See previous thread for my "corrections". Nothing stood out that matched any of my documentation.

Additionally, I did have a chance to review this issue with my development team and they are confident that they know what the issue is. However, we want to make sure we are clear on the goal that you are trying to accomplish at this time.

We would like to set up a quick call just to finalize everything and make sure you are in a good place quickly. Would you 3:00PM EST work for you to meet with me and my developer?

Let me know if there is a better time and we will be sure to accommodate. I look forward to speaking with you soon.

Rick Nelson

Implementation Engineer | Jifiti

Error! Filename not specified.Error! Filename not specified.

-
2. Below is a request url and body. How would you set it up on Postman to get a response? Please make screenshots using Postman client that include all your edits on all the relevant screens.

```
Url: https://exmaple.com/api/send
Authentication: basic - username: abcd, password: 1234

Body:

{
  "GId": "weInfjregnrkhnj",
  "AId": "hjtfkjfthdrt",
  "LoanType": "1",
  "FirstName": "Mike",
  "LastName": "Smith"
}
```

-
3. Your client has embedded your JS snippet in their website and there's an error in the console when running the code. List the steps you would take using Chrome developer tools to investigate the issue.

-
- 4.
- How would you formulate a REST API request for retrieving details of a dog named "humi"? (write the exact request syntax in a standard format).
 - Apparently "humi" is not in the database, what http error code would you return?

-
5. Consider the Whatsapp feature of voice messages. Write a short documentation to explain the value of this feature and its main attributes.

-
6. Below are 2 tables - Merchants and Tokens. Tokens is a table that holds tokens for a merchant to use. Each merchant can have multiple tokens.

Table: Merchants Columns:

- Id (nvarchar, not null),
- Name (nvarchar, not null),
- PartnerId (nvarchar, not null),
- Group (nvarchar, not null),
- JoinDate (nvarchar, not null)

Table: Tokens Columns:

- Id (nvarchar, not null),
- InsertDate (DateTime, not null),
- MerchantId (nvarchar, not null),
- TokenType (int, not null),
- IsActive (bit, not null)

Write a query that will fetch merchants that have inactive tokens that were created after Aug 30, 2021 for merchants that joined after Jan 1, 2020. Required output: merchant name, merchant join date. Don't display rows with duplicate data.