Rent Ready Technical Assessment

Cloud .NET Developer

Preparation

This assignment is designed to demonstrate:

1. Your .NET Development expertise.
2. Ability to integrate non-standard services. In our case, you will use Dynamics 365 Dataverse.
3. Ability to learn new things. We do not require you to know Azure Cloud or Dynamics, but you can demonstrate how you integrate knowledge.

**Expected outputs:**

The solution should emphasize how you work in real projects. Your future team members will evaluate if they want to work with a guy, who writes code, documents the solution in a way as they demonstrated in the test assignment.

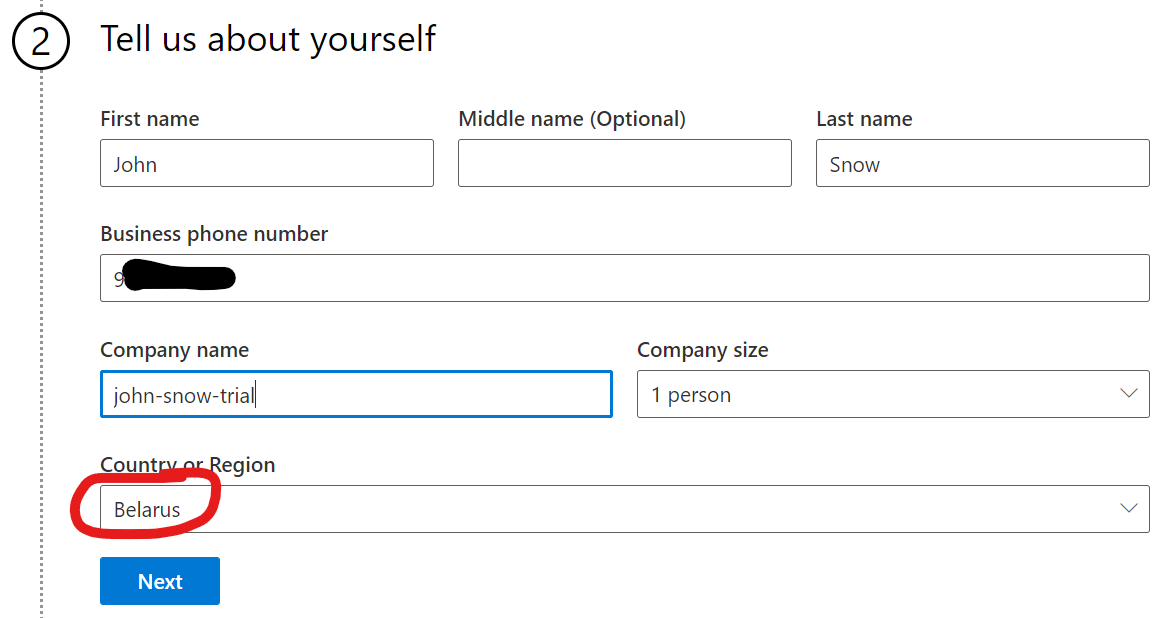
1. Public git repository link.
2. Unit tests. If you are not familiar with them, follow [the article](https://medium.com/vx-company/the-5-unit-testing-guidelines-f21d39c33e0b).
3. Readme.md should be provided in the root directory.
4. No build error and no warnings.
5. The solution should be deployable to Azure Cloud following provided instructions in Readme.md.
6. The solution should work in any Azure Cloud and Dataverse environments.

Task 1. Setting up a Dataverse trial

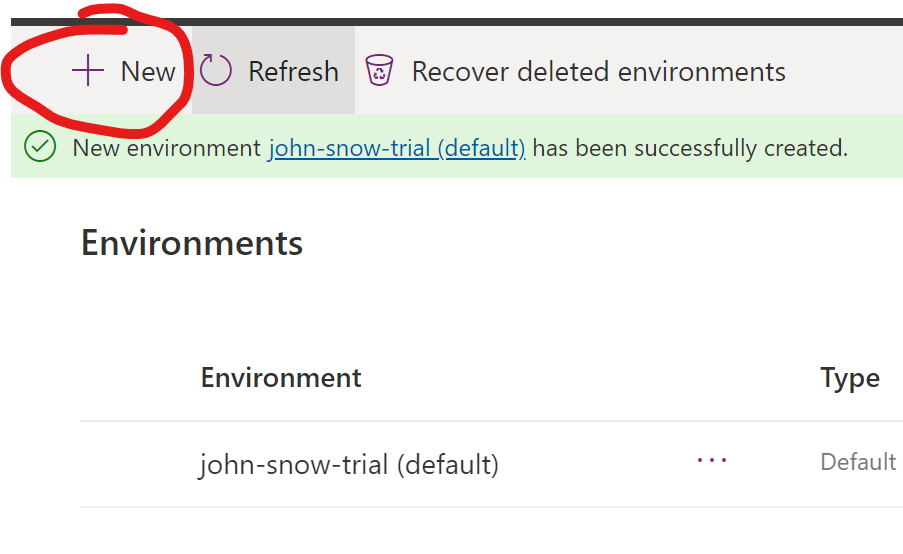
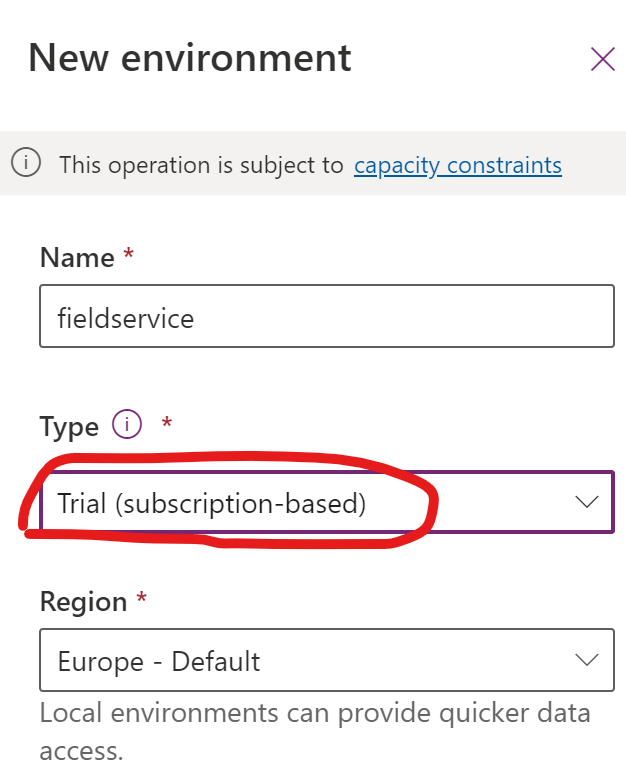
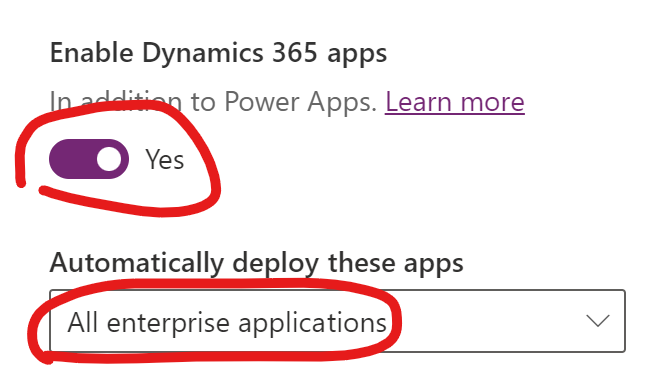
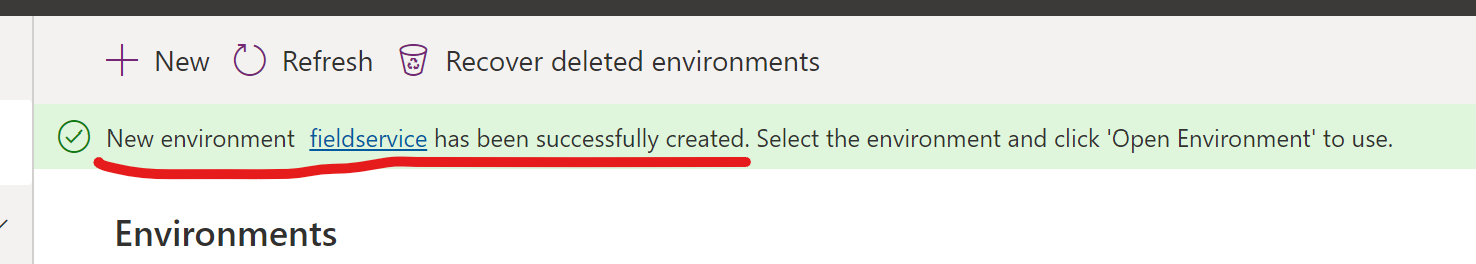
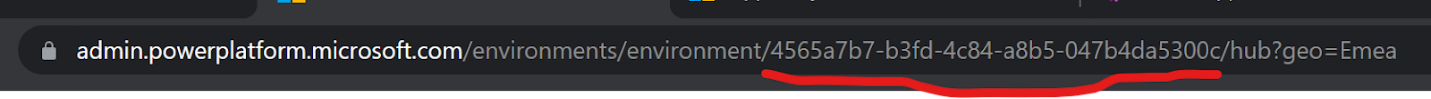
The **outputs** from this exercise:

1. Username
2. Password
3. Environment URL
4. Link to the table “Time Entries”

**Steps:**

1. Follow the [link](https://signup.microsoft.com/get-started/signup?ali=1&products=5a589020-9713-4df5-a155-8761fbb7e419&ru=https%3A%2F%2Fadmin.powerplatform.microsoft.com%2Fenvironments%3Fopen%3Dnew%26type%3DTrial%26template%3DD365_FieldServicePremiumTrial).
2. Step 1: Enter your personal email, click ‘Next’. Confirm ‘Set up account’ when prompted.
3. Step 2: If you are living in Russia, you should specify Country or Region = “**Belarus**” (NOT Russia, otherwise, you will not be allowed to proceed because of Russian local laws), your phone without a country code, first name, and last name. Click ‘Next’.  
   
4. Step 2: Set the correct country code. If you selected Belarus, just update the country code back to the correct one. Click ‘Send verification code’ and confirm it.
5. Step 3: Set **username** and **password**. Store them somewhere. Click ‘Next’.
6. Step 4: When prompted ‘Enter your UNP registration number:’, enter 6300000000.

Congratulations! You set up your trial account. Now you will create an environment for your task.

1. Go to <https://admin.powerplatform.microsoft.com/environments>
2. Log in with the username and password you created on Step 3. The environments list will be opened.
3. Click ‘+New’.  
   
4. Enter the environment name and set Type to ‘Trial (subscription-based)’. Click ‘Next’  
   ’
5. When the next step is loaded, set ‘Enable Dynamics 365 apps’ to ‘Yes’ and ‘Automatically deploy these apps’ to ‘All enterprise applications’ or ‘FieldService Premium Trial’. Click ‘Save’.  
   
6. Wait 1-2 minutes, click “Refresh” until you see that the environment has been successfully created.  
   
7. Select the environment and find the “**Environment URL**” value. Copy it. This will be necessary in SDK.
8. Find the **environment guid** in the address bar. Copy it.  
   
9. Open “https://make.powerapps.com/environments/**environment guid**/entities/**environment guid**/msdyn\_timeentry#data”. This will be your “Time Entry” URL. There is no Data. You will use it to see how your logic pushes data here.

Task 2: Create an azure function

Summary

The goal of this task is to:

* Provision a trial Azure account. You can use your existing account, or start a free trial using the username and password you created in the previous step.
* Create an Azure function that can be easily deployed to the Cloud.
* Build a business logic that interacts with timeentries in Dataverse. You can use <https://github.com/microsoft/PowerPlatform-DataverseServiceClient>. For simplicity, you can use username/password authentication.
* Add unit tests.

 The entity to interact with is the **msdyn\_timeentry** entity.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Schema Name** | **Field Type** | **Additional Comments** |
| Start | Msdyn\_start | DateTime |  |
| End | Msdyn\_end | DateTime |  |

Functional Details

Interact with the Time Entry Entity in Dataverse.

The azure function should accept the payload:

{

  "$schema": "http://json-schema.org/draft-04/schema#",

  "type": "object",

  "properties": {

    "StartOn": {

      "type": "string",

      "format": "date"

    },

    "EndOn": {

      "type": "string",

      "format": "date"

    }

  },

  "required": [

    "StartOn",

    "EndOn"

  ]

}

The logic should create an *msdyn\_timeentry* record (msdyn\_start=msdyn\_end) for every date in the date range from *StartOn* to *EndOn*. The logic should also ensure that there are no duplicate time entry records created per date.