Deployment Manual

Setup GOSH FHIR API

Please review the following deployment guide link which contains the instructions and source code for the FHIR standard access at Great Ormond St Hospital: https://github.com/goshdrive/FHIRworks_2020

Follow the instructions in the README of the above Git repo to set up the GOSH FHIR API on your localhost.

Credentials for the above .Net Application

"Instance": "https://login.microsoftonline.com/{0}",

"Tenant": "ca254449-06ec-4e1d-a3c9-f8b84e2afe3f",

"ClientId": "0f6332f4-c060-49fc-bcf6-548982d56569",

"ClientSecret": "ux@CJAaxCD85A9psm-Wdb?x3/Z4c6gp9",

"BaseAddress": "https://gosh-fhir-synth.azurehealthcareapis.com",

"Scope": https://gosh-fhir-synth.azurehealthcareapis.com/.default

Creating an Azure account

The next step needed is to create an Azure account. Azure is used in this project as an area to store all the generated Word documents containing patient data.

Visit this link and click 'Start for free' to create your own Azure account.

Sign in with your Microsoft or GitHub account or create a free Microsoft account.

On the About you page, select your correct country or region. Enter your first and last name, email address, and phone number. Depending on your country, you might see additional fields, such as a VAT number. Select Next to continue.

On the Identity verification by phone screen, select your country code, and type the number of a telephone to which you have immediate access.

You have the option of text or call back to obtain a verification code. Select the relevant button, type the code in the Verification code box, and select Verify code.
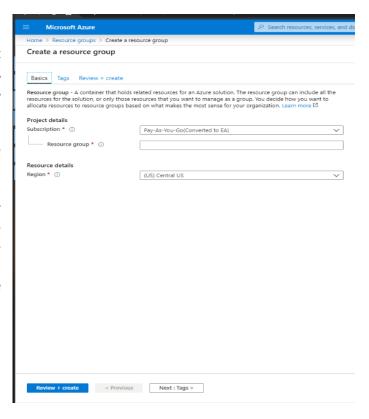
If the verification code is correct, you're asked to enter details of a valid credit card. You will not be charged anything for signing up to Azure as Azure operates on a pay as you go basis. Azure also provides you with £150 of free credit when you sign up for the first 30 days. Enter the card information and select Next.

The last step is to review the agreement and privacy statement then select Sign up.

You have now successfully set up a free account on Azure and should be on the Azure portal home page.

Create a Resource Group

Now you have an Azure account the next step is to set up a resource group. A resource group is a collection of resources that share the same lifecycle, permissions, and policies. To create a resource group first click on 'Resource groups' on the Azure portal home page. Then click on 'Create resource group'. Choose a subscription to use and enter the name you want to call your resource group into the field called 'Resource group'. Then for the field 'Resource details Region', select the appropriate region for you from the drop-down list. Then click 'Review and create', followed by 'Create'.

Creating Azure blob storage container

This link contains instructions on how to set up a storage account on Azure but there a few changes that need to be done. In the section 'Create a storage account', on step 7, for the field 'Account kind' don't leave it as 'StorageV2 (general-purpose v2)'. Instead change this to be BlobStorage. Step 8 can be skipped.

Other than the above changes, follow the steps in the link to create your storage account. You now need to create three containers to store the three different types of documents that can be generated. The containers will store patient feedback forms, patient details forms and patient health data forms. Go to the Overview page in your storage account resource. Click on 'Containers'. You then need to create a new container by clicking on the '+ Container' option. Name the container what you want but have it relate to what is going to be stored in it. Do this two more times to create another two containers to store the other document types in.

Fill in config.json files.

In both the PatientDocumentAPI and PatientDocumentGenerator directories there is an empty config file template that you need to fill in.

account_name is the name of the storage resource you created.

account_key is the storage account key. You can find this by clicking on Access keys under settings in the storage resource on Azure.

feedback_container_name is the name of the container you made to store feedback documents.

health_data_container_name is the name of the container you made to store health documents.

patient_info_container_name is the name of the container you made to store patient info documents in.

User Manual

Abstract

This project is something that would be used by GP clinics. Each clinic would set up their own Azure storage account and set up their own API that would use the details of their storage account. They could then use the GUI to type in patient IDs and get back a document they could print off and hand to the patient to collect patient feedback with. They could also generate a document with visualisations of the patient's health data such as weight, heart rate, BMI and respiratory rate.

Endpoints

/FormFiller/feedback?id=*patient_id*

This endpoint will find the name and address of the patient whose ID was provided and will create a Word document asking that patient for feedback on the services they received. This Word document will then be uploaded to the Azure account and is stored there. The endpoint will return JSON containing the patient data and questions used to make the patient feedback document as well as creating the document on Azure.

/FormFiller/feedbackDocumentData?id=*patient_id*

This endpoint will return JSON containing patient data and questions needed to generate a Document or form asking that patient for feedback on the services provided to them. It will not generate the word document and store it on Azure like the above endpoint.

The JSON response for the above two endpoints looks similar to what's shown below:

```
{
  "address": {
    "address_lines": [
      "506 Herzog Byway Apt 99"
    ],
    "city": "Barre",
    "country": "US",
    "postcode": "01005",
    "state": "Massachusetts"
  },
  "name": {
    "first_name": "Abby752",
    "last_name": "Beatty507",
    "prefix": "Ms."
  },
  "questions_and_messages": {
    "clinic": "What is the name of the clinic/department where you were
treated?",
    "comment": "With regards to your response to the previous question,
what is the main reason you feel this way?",
    "hospital": "What is the name of the hospital where you received
treatment?",
    "intro": "We welcome all feedback on the services we provide to tell us
what we are doing right and where we can improve.",
    "recommendation": "Based on your recent experience of our services, how
likely are you to recommend us to friends or family if they needed similar
care or treatment?"
  }
}
```

/report/rawData?id=*patient_id*

This endpoint will return JSON regarding patient health data. It will return the values of readings, date of those readings and units of the readings for weight, heart rate, BMI , diastolic blood pressure, systolic blood pressure and respiratory rate.

/report/patientReport?id=*patient_id*

This endpoint not only returns JSON containing information on the readings and dates of those readings for patient health data regarding weight, heart rate, BMI, systolic blood pressure, diastolic blood pressure and respiratory rate, but also creates a word document containing visualisations of the data and saves it to an Azure storage

Both of the endpoints that start with /report will return the same JSON response containing information on the patient's health data. A sample of the JSON data returned by these endpoints is shown below.

```
{
    "Body Mass Index": {
        "Dates": [
            "2013-10-01 21:27:12.215000+01:00",
            "2014-10-07 21:27:12.215000+01:00",
            "2015-10-13 21:27:12.215000+01:00",
            "2011-09-20 21:27:12.215000+01:00",
            "2010-09-14 21:27:12.215000+01:00",
            "2017-10-24 21:27:12.215000+01:00",
            "2016-10-18 21:27:12.215000+01:00",
            "2012-09-25 21:27:12.215000+01:00"
        ],
        "Values": [
            21.68,
            22.21,
            22.6,
            20.37,
            20.18,
            23.4,
            22.97,
            21.0
        ],
        "Unit": "kg/m2"
    }
}
```

/info/infoDocument?id=*patient_id*

This endpoint creates a document with all of the patient's personal details, asks them to check if they are correct and up to date and if not, provides lines on which they can write the correct details. The JSON response returned is a message saying if the file was created successfully or not. The JSON response look like this:

```
[
  {
    "message": "Document created successfully"
  },
  200
]
```

Running the project

If you followed the steps in the deployment manual then you should have the GOSH FHIR API running on localhost:5001. If not please run this API by following the instructions in the README of the Git repo provided at the start of the deployment manual.

You then need to open a terminal/command prompt in the 'PatientDocumentAPI' directory. Type 'python FormAPI.py' and hit enter to run the document generator API on localhost:5010.

Next open a terminal/command prompt in the 'PatientDocumentGenerator' directory. Type 'python GUI.py' and hit enter to run the GUI tool. In the patient ID field enter a patient ID such as: 8f789d0b-3145-4cf2-8504-13159edaa747

The right side of the GUI contains buttons for generating reports that will be stored on Azure. These reports can be retrieved from Azure by using the buttons on the left-hand side of the GUI. The 'Get' buttons will download the word documents from Azure into the PatientDocumentGenerator directory.

Examples of what these files look like are provided in the GitHub repository:
https://github.com/ckoppula199/GOSH-FHIRworks2020-PatientDocumentGenerator