Deployment and Operations Guide (Runbook)

Harkify Mobile Application

Milestone 3 – Bravo Team

Software Engineering Project

SWEN 670

August 05, 2021

Version 1.1

Project Plan Approvals

Name	Signature	Date

Revision History

Date	Version	Description	Author
07/19/2021	1.0	Initial Release	Team Bravo
07/29/2021	1.1	Azure Installation Guide	Team Bravo

Table of Contents

1. Introduction

1.1 Purpose

The purpose of the deployment and operations guide is to provide all technical information necessary to install and deploy Harkify, a mobile application designed to assist people who experience problems with short-term memory. If not mentioned by name any references to "the system," or "the application," refer to Harkify.

1.2 Intended Audience

This guide is designed for use by the technical stakeholders in the planning, setup, installation, and deployments of the application. The information contained in this document should allow said stakeholders to coordinate the deployment of Harkify and allow new developers to added to the project.

1.3 Technical Project Stakeholders

This section provides a list of all current stakeholders with an interest in this project:

Name	Role
Mir Asadullah	Product Owner
Roy Gordon	Stakeholder/PM Mentor
Johnny Lockhart	Stakeholder/DSO Mentor
Malcolm Freeney	Lead Project Manager
Raul Benavides	Bravo/Harkify PM
Madison Dunning	Lead Developer
Elshaday Mesfin	Business Analyst/Developer
Ben Cushing	Business Analyst/Developer
Tyler Puschinsky	Test Engineer/Developer
Alec Baileys	DSO Contact/Developer
Robert Wilson	DSO Engineer
Michelle Monfort	DSO Engineer
Jeroen Soeurt	DSO Engineer

2. Software Overview

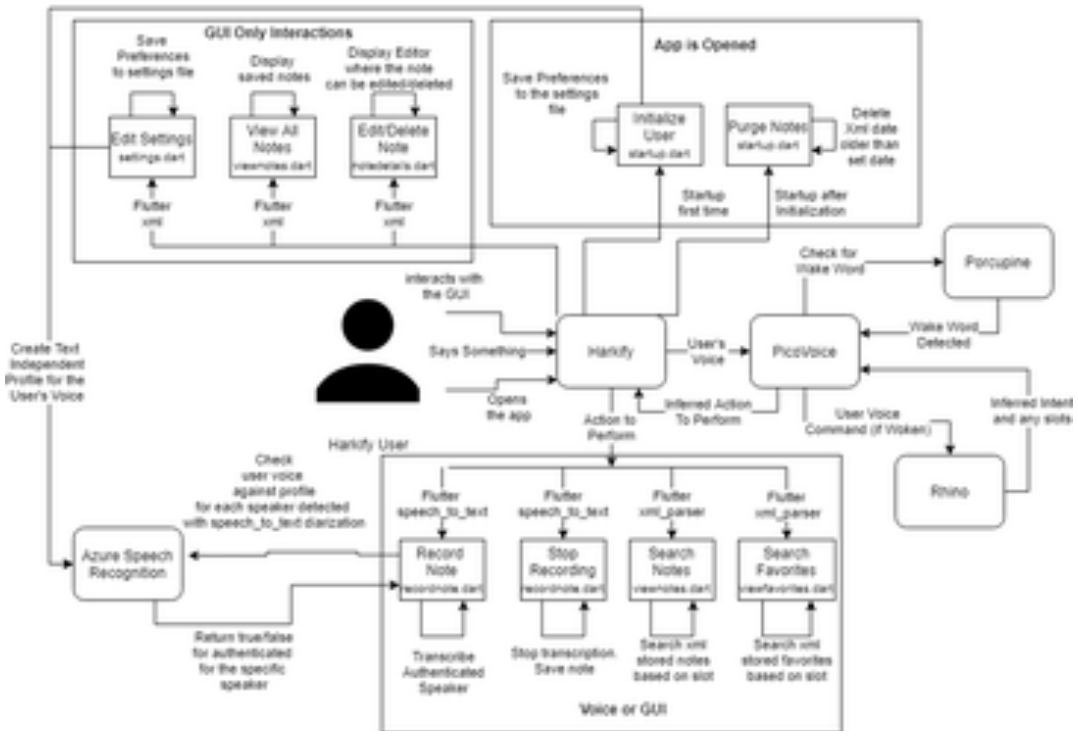


Figure 1 Harkify Screens and Services

2.1 Flutter

Harkify uses the Flutter framework to easily design an application that run on both Android and iOS. The application allows for user to record voice memos that the user can listen or view later to remember important details from conversations or the user's dictation.

The user can choose to record a new note or if the user speaks the wake word and the recording trigger, the system will automatically record the user and ignore any other speakers. The system also listens for triggers that signal Harkify to stop recording. This is achieved by identifying the user's voice when the user starts the application for the first time. When the user starts the application for the first time, Harkify presents the user with how-to videos about how to use the application. During this start up process, a sample of the user's voice will be stored, and the wake words the users will use will be defined. A settings page allows users to define new wake words and choose how long Harkify saves voice note.

All voice notes are to be encrypted, and user's important details such as address phone number, SSN, and Medicare number will be stored in a separate way than normal notes, and the user will be able to retrieve these notes by using Rhino to infer what notes the user is asking Harkify to find, and whether an action such as creating a future notification from the application for the user is necessary.

2.2 Picovoice

Picovoice is used to take recorded speech and convert it into text which is saved by the system. For searching notes, the Rhino NLU is employed to give users a more effective and easier to use solution to search for notes compared to a simple keyword search. Users can review notes, edit, or delete notes that are no longer relevant. A setting screen allows users to set how long notes should be saved, create, and edit voice triggers, and allows users to further train voice recognition.

Picovoice and Rhino are implemented via a local library that uses an API to handle requests, it does not need internet access to function. This also means that processing occurs much faster locally than having to send requests through an API. The only service that needs internet is Microsoft Azure which is used to create a profile of the Harkify user, so that Harkify will only transcribe the user's voice, avoiding issues with recording another speaker without their permission.

2.3 Azure Speech Recognition

Azure is one of the top speech recognition software that prioritizes user security. Azure provides a scalable product that identifies a user by their voice profile and allows the system to connect easily. While not part of the Harkify, Azure also provides many other services such as speech to text so help in audio services. Due to its robust system, scalability, and user security we decided to develop with Azure.

The voice sample that the user provides during initial setup of Harkify be sent to Azure to create a profile of the user's voice that can be used to compare to speakers to determine which speaker to record. When the audio is submitted, the service will reference the profile to verify the voice is from the user. Azure will send a result object with a result score from 0-1.0 and a result confirmation of accepted of denied. For security, the system saves the voice under an id, as opposed to the name of the user, and requires that Harkify save the id for the user. Requests are sent through REST API and internet connectivity is required for the initial process to create the profile when the user starts using Harkify.

3. Configurations

3.1 Flutter Framework

Harkify is an application developed using the Flutter designed to run on Android and iOS devices. The front-end employs various features of the Flutter framework to create an appealing and intuitive UX that simplifies the UI development process for developing Harkify. Harkify is developed using VS Code, and the code repository is hosted on GitHub. Git is required for cloning the repository and enabling other interactions with the GitHub repository through VS Code. Android studio is required for the bundled SDK that is required to build and run the application on Android emulators.

This project requires the following for developing Harkify using the Flutter framework (earlier versions may suffice, but may not have been verified):

- Flutter framework: 2.2.0 or later
- Android Studio: 4.2 or later
- Microsoft Visual Studio Code: 1.57 or later
- Git: 2.31 or later

Harkify requires the following packages to function (more information about these packages can be found at https://pub.dev/). The file pubspec.yaml holds all of the metadata on the packages used in Harkify:

- Path Provider: 2.0.2
- XML: 5.1.2
- Intl: 0.17.0
- Cupertino Icons: 1.0.3
- Flutter Search Bar: 3.0.0-dev.1
- Speech to Text: 4.2.1
- Highlight Text: 1.0.0
- Avatar Glow: 2.0.1
- Clipboard: 0.1.3
- Picovoice: 1.1.8
- Encrypt: 5.0.0
- HTTP: 0.13.3

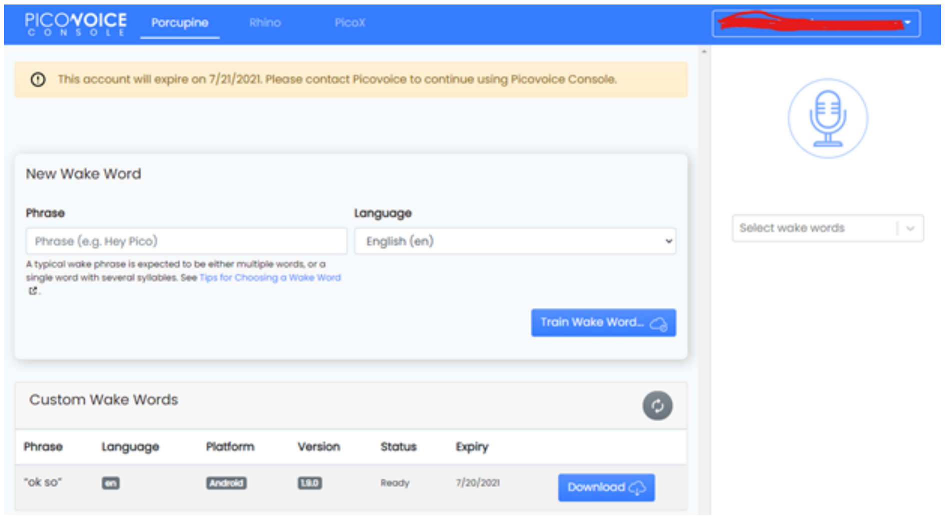
Harkify requires the following assets:

- note_taker.rhn
    - This is the file that holds the configuration for Rhino for Harkify specific interactions.
- ok_so.ppn
    - This is the file that defines the initial, default wake word used by Porcupine.

3.2 Picovoice

1. To create new wake words and to train Rhino, the Picovoice console is required. Log into the account for the project or create a new account to experiment with new features, note the free version of Picovoice Console is intended for a single user and has a limited free trial period.

2. After logging in you can set wake words under the Porcupine tab, or select existing wake words on the right:



This account will expire on 7/21/2021. Please contact Picovoice to continue using Picovoice Console.

New Wake Word

Phrase: Phrase (e.g. Hey Pico)

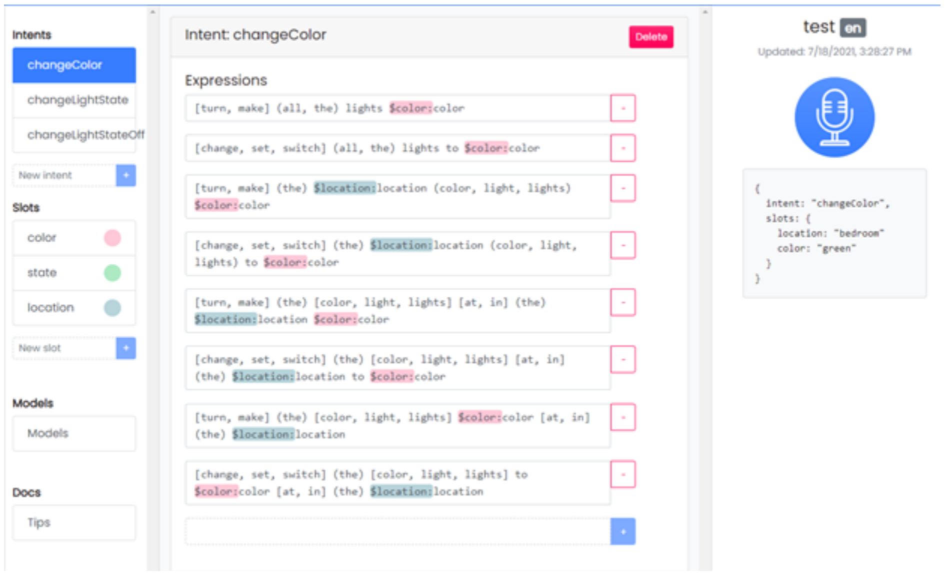
Language: English (en)

A typical wake phrase is expected to be either multiple words, or a single word with several syllables. See Tips for Choosing a Wake Word.

Custom Wake Words

Phrase	Language	Platform	Version	Status	Expiry
"ok so"	en	Android	1.9.0	Ready	7/20/2021

3. To train Rhino, intents must be created, and expressions added. The idea is to train Rhino to recognize the many ways that a user may express a certain intent. Add new expressions and test to see if Rhino recognizes them.



Intent: changeColor

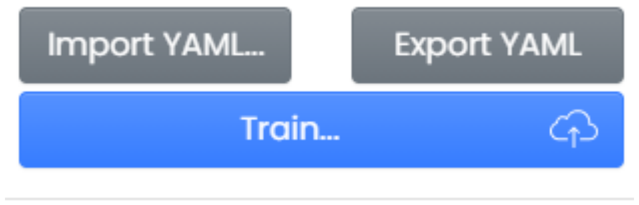
Expressions

```
[turn, make] (all, the) lights $color:color
[change, set, switch] (all, the) lights to $color:color
[turn, make] (the) $location:location (color, light, lights) $color:color
[change, set, switch] (the) $location:location (color, light, lights) to $color:color
[turn, make] (the) [color, light, lights] [at, in] (the) $location:location $color:color
[change, set, switch] (the) [color, light, lights] [at, in] (the) $location:location to $color:color
[turn, make] (the) [color, light, lights] $color:color [at, in] (the) $location:location
[change, set, switch] (the) [color, light, lights] to $color:color [at, in] (the) $location:location
```

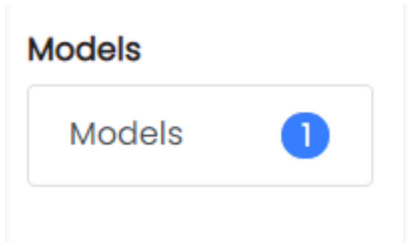
Updated: 7/18/2021, 3:28:27 PM

```
{
  intent: "changeColor",
  slots: {
    location: "bedroom"
    color: "green"
  }
}
```

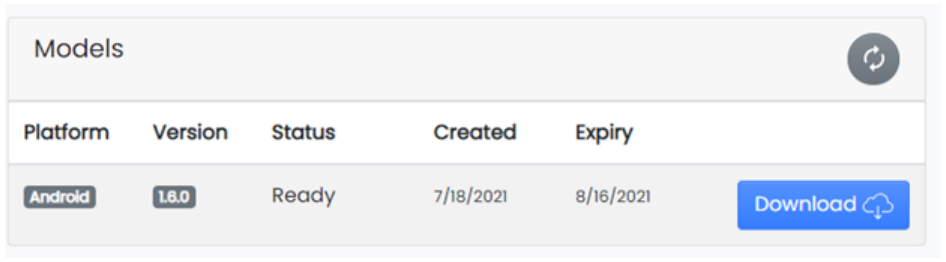
4. After creating and testing the expressions for the intents needed, select Train in the bottom-right.



5. After the training process select the model on the left-hand side.

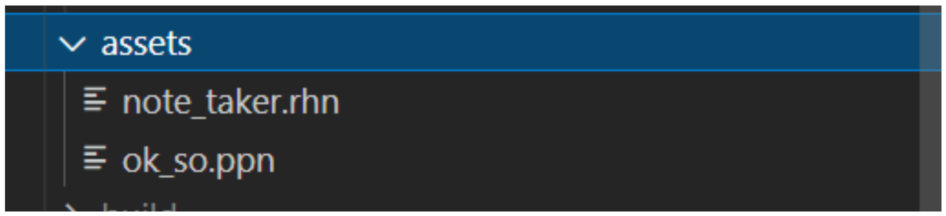


6. Download the Rhino model

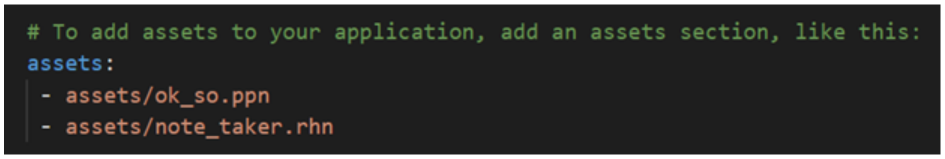

Models

Platform	Version	Status	Created	Expiry	
Android	1.6.0	Ready	7/18/2021	8/16/2021	Download

7. Create an assets directory within the Harkify directory and place the files.


```
assets
  note_taker.rhn
  ok_so.ppn
```

8. Provide the path to the assets in pubspec.yaml.



```
# To add assets to your application, add an assets section, like this:
assets:
  - assets/ok_so.ppn
  - assets/note_taker.rhn
```

3.3 Azure Speech Recognition

Harkify makes use of Azure Speech Recognition to identify the user of the application by their voice. Requests for Azure speech recognition are handled through HTTPS requests using a REST API. This is achieved by making a HTTP GET request to the API. Harkify receives a response in the form of a Json file, that is used by the application to create a voice profile of the Harkify user, comparing speakers in future recordings to only transcribe the user's voice into a note if the check returns true for the user's voice.

HTTPS requests also need to provide authentication for Azure. An endpoint and a key are required to provide authentication for Azure and are as follows.

- endpoint: https://teambravo.cognitiveservices.azure.com/
- key: 67f249403fd2443a81909fc8f89f34b0

4.Software Installation

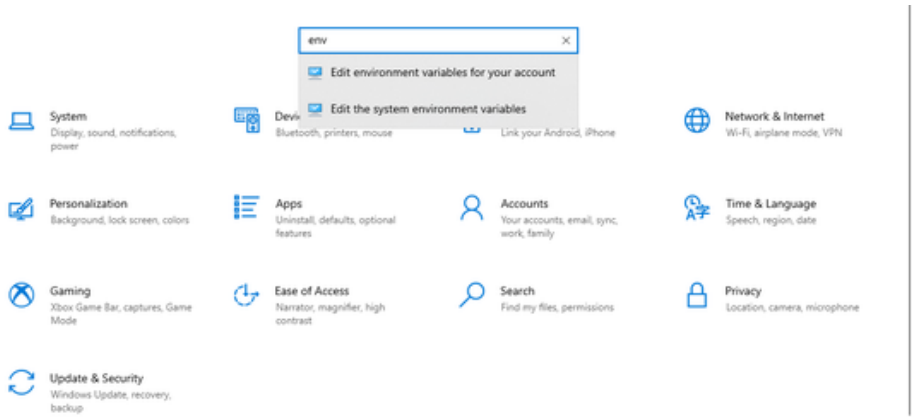
4.1 Flutter Framework

4.1.1 Installing Flutter and VS Code

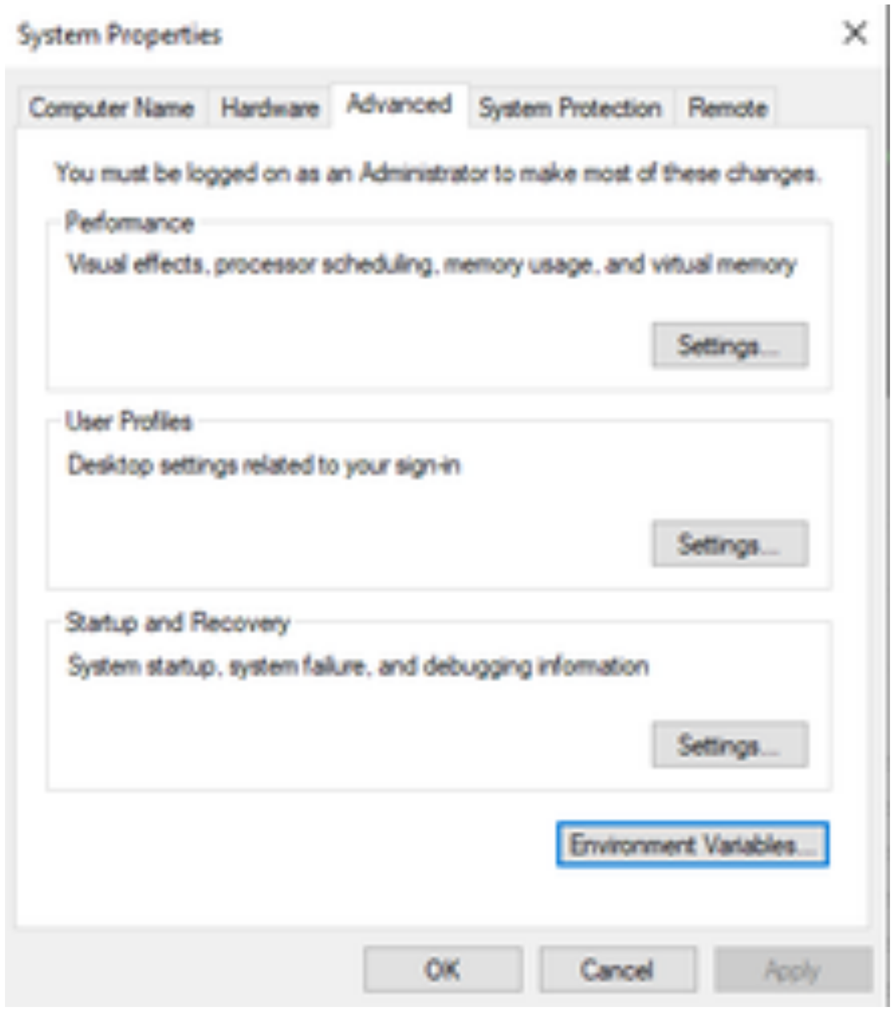
This section will cover how to set up Flutter on VS Code on Windows:

1. First to get the latest version of Flutter navigate to https://flutter.dev/docs/get-started/install/windows and choose the option to download the latest Flutter SDK. Please take note of the system requirements for Flutter from the Configurations section.
2. After the file has been downloaded, unzip the file into a directory named Flutter, and make sure not to place the "Flutter" directory in a location that requires additional privileges such as C:\Program Files\. Creating a folder in your user directory or directly on C:\ should suffice.

3. Once you have created your Flutter directory navigate to System Properties (Windows 10 click settings and type “environment variables”).


env
Edit environment variables for your account
Edit the system environment variables
System
Display, sound, notifications, power
Devi
Bluetooth, printers, mouse
Link your Android, iPhone
Network & Internet
Wi-Fi, airplane mode, VPN
Personalization
Background, lock screen, colors
Apps
Uninstall, defaults, optional features
Accounts
Your accounts, email, sync, work, family
Time & Language
Speech, region, date
Gaming
Xbox Game Bar, captures, Game Mode
Ease of Access
Narrator, magnifier, high contrast
Search
Find my files, permissions
Privacy
Location, camera, microphone
Update & Security
Windows Update, recovery, backup

4. Under the Advanced tab, look for Environment Variables near the bottom of the screen and select it.



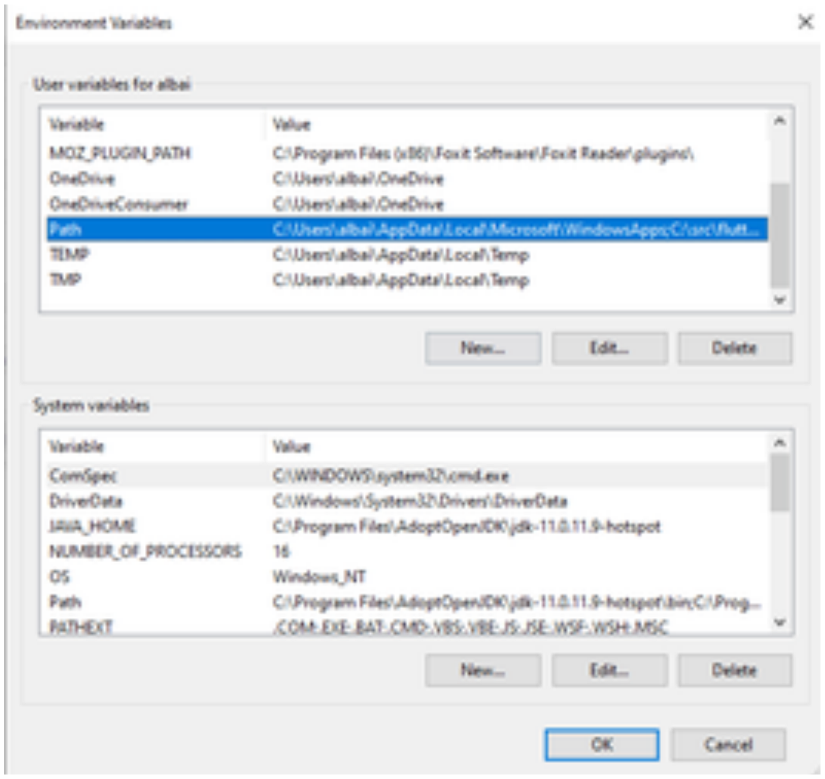
You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

5. Add a new path entry, in user variables with this your flutter/bin path such as C:\src\flutter\bin. Save the setting.



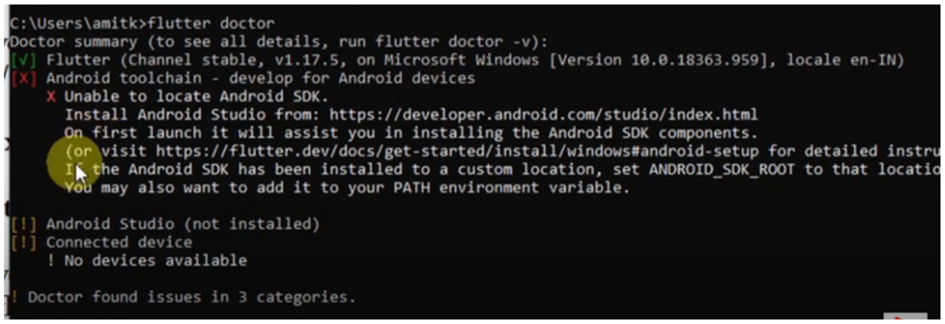
User variables for albai

Variable	Value
MOZ_PLUGIN_PATH	C:\Program Files (x86)\Foxit Software\Foxit Reader\plugins\
OneDrive	C:\Users\albai\OneDrive
OneDriveConsumer	C:\Users\albai\OneDrive
Path	C:\Users\albai\AppData\Local\Microsoft\WindowsApps;C:\src\flutt...
TEMP	C:\Users\albai\AppData\Local\Temp
TMP	C:\Users\albai\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot
NUMBER_OF_PROCESSORS	16
OS	Windows_NT
Path	C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin;C:\Prog...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC

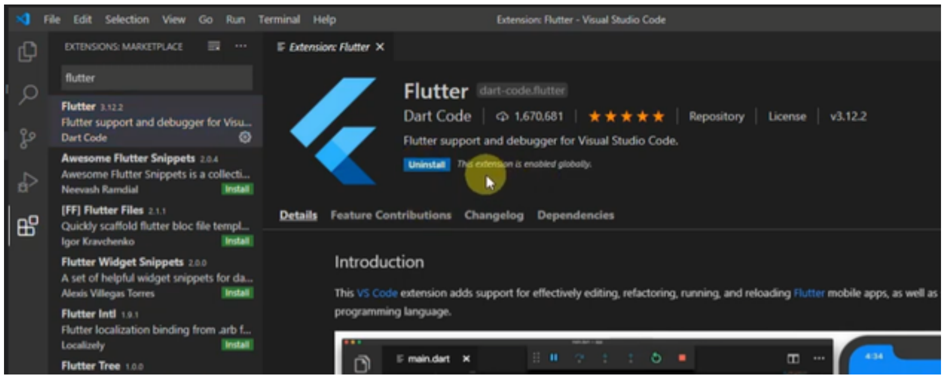
6. While you are here check if JAVA_HOME is set, if not set it to the location of a JDK that meets or exceeds the requirements set under the Configurations setting.
7. Open Command Prompt to verify flutter is set up properly. In the shell type "flutter doctor" and you should get a similar result



```
C:\Users\amitk>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, v1.17.5, on Microsoft Windows [Version 10.0.18363.959], locale en-IN)
[X] Android toolchain - develop for Android devices
    X Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/index.html
      On first launch it will assist you in installing the Android SDK components.
      (or visit https://flutter.dev/docs/get-started/install/windows#android-setup for detailed instru
      If the Android SDK has been installed to a custom location, set ANDROID_SDK_ROOT to that locatio
      You may also want to add it to your PATH environment variable.

[!] Android Studio (not installed)
[!] Connected device
    ! No devices available

! Doctor found issues in 3 categories.
```

8. In this screenshot the flutter doctor command is working but more steps are needed to set up Flutter in VS Code. Installation of Android Studio is required and is covered in 4.1.2 Installing Android SDK.
9. Install Microsoft VS Code from https://code.visualstudio.com/Download. Choose the proper download for your system.
10. Run the installer, leave the default installation options in place.

11. After installation open VS Code, look for the Extensions icon on the left-hand bar, click it, and type “flutter” as shown below, and install the flutter plug-in for VS Code.

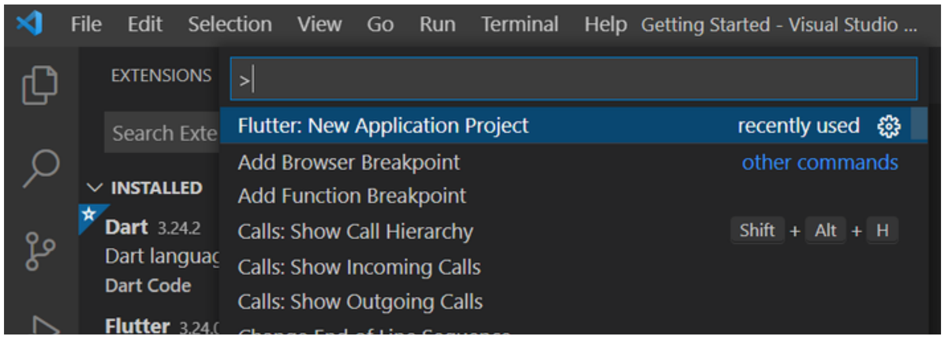

File Edit Selection View Go Run Terminal Help
Extension: Flutter - Visual Studio Code
EXTENSIONS: MARKETPLACE
flutter
Flutter 3.12.2
Flutter support and debugger for Visu...
Dart Code
Awesome Flutter Snippets 2.0.4
Awesome Flutter Snippets is a collecti...
Neevash Ramdial
Install
[FF] Flutter Files 2.1.1
Quickly scaffold flutter bloc file templ...
Igor Kravchenko
Install
Flutter Widget Snippets 2.0.0
A set of helpful widget snippets for da...
Alexis Villegas Torres
Install
Flutter Intl 1.9.1
Flutter localization binding from .arb f...
Localizely
Install
Flutter Tree 1.0.0
Extension: Flutter
Flutter dart-code.flutter
Dart Code | 1,670,681 | ★★★★★ | Repository | License | v3.12.2
Uninstall
This extension is enabled globally.
Details Feature Contributions Changelog Dependencies
main.dart

Flutter support and debugger for Visual Studio Code.

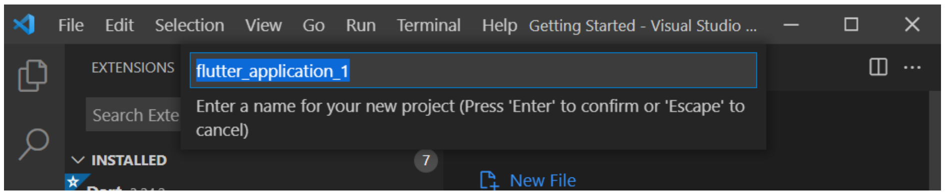
Introduction

This VS Code extension adds support for effectively editing, refactoring, running, and reloading Flutter mobile apps, as well as programming language.

12. To create a new project and verify configuration is successful, select View from the top bar and select Command Palette, in the box that pops up type “Flutter: New Application Project.”


File Edit Selection View Go Run Terminal Help Getting Started - Visual Studio ...
EXTENSIONS
>|
Search Exte
INSTALLED
Dart 3.24.2
Dart languag
Dart Code
Flutter 3.24.
Flutter: New Application Project
recently used
Add Browser Breakpoint
other commands
Add Function Breakpoint
Calls: Show Call Hierarchy
Shift + Alt + H
Calls: Show Incoming Calls
Calls: Show Outgoing Calls

13. Select a directory for the project.
14. Select a name for the project. Flutter project names should be all lowercase, with underscores to separate words.

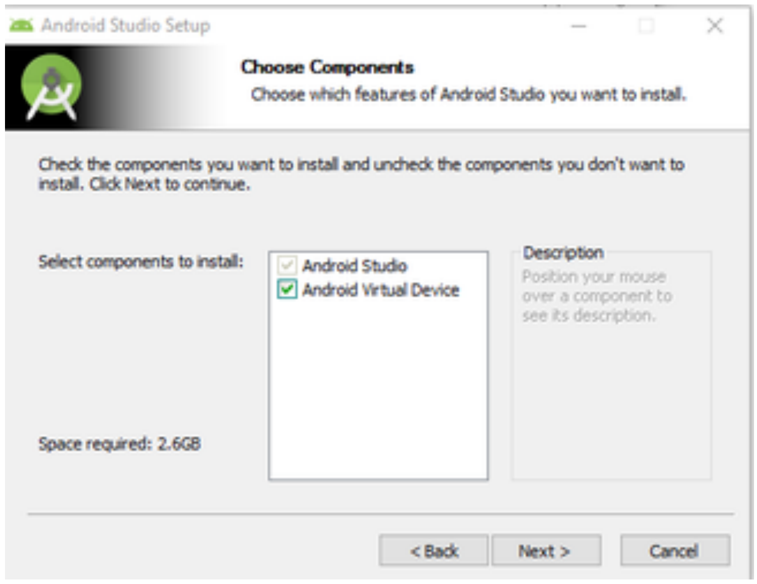

File Edit Selection View Go Run Terminal Help Getting Started - Visual Studio ...
EXTENSIONS
Search Exte
INSTALLED
flutter_application_1
7
New File

Enter a name for your new project (Press 'Enter' to confirm or 'Escape' to cancel)

4.1.2 Installing Android SDK

Android Studio is required for the SDK that allows VS Code's Android emulator to function. This guide will document how to download the latest version of Android Studio.

1. Navigate to https://developer.android.com/studio and choose Download Android Studio.
2. Run the installer, choose to install Android Studio and the Android Virtual Device.



Android Studio Setup

Choose Components

Choose which features of Android Studio you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

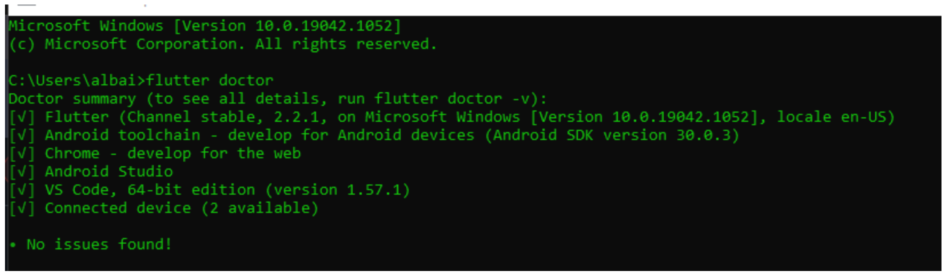
Select components to install: [✓ Android Studio] [✓ Android Virtual Device]

Description: Position your mouse over a component to see its description.

Space required: 2.6GB

4.1.3 Verify with Flutter Doctor

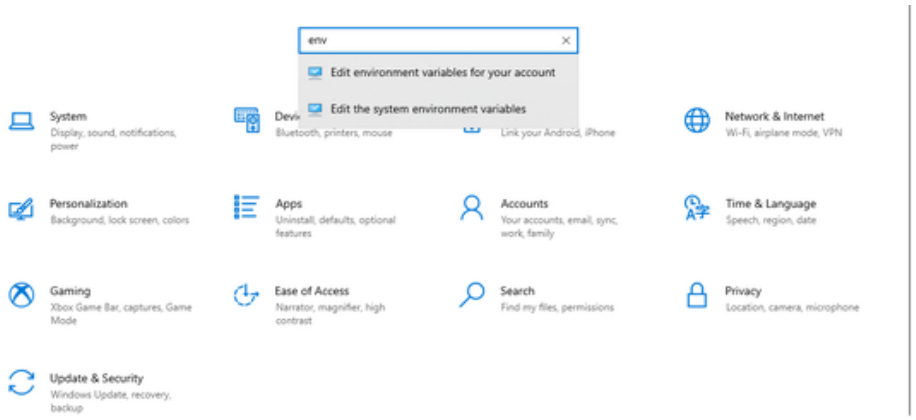
1. Run Flutter Doctor again.
2. The goal is for all items to have a green checkmark. You may want to install Chrome if you do not already have it. If there is an X for Android Studio or VS Code refer to 4.1.1 and 4.1.2 respectively. For Flutter refer to 4.1.1. Android toolchain may still have a red X and will be covered in the following steps.



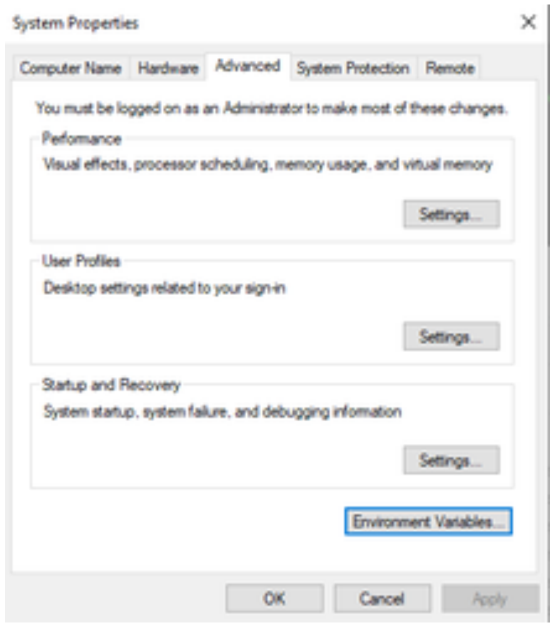
```
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Users\albai>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 2.2.1, on Microsoft Windows [Version 10.0.19042.1052], locale en-US)
[√] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[√] Chrome - develop for the web
[√] Android Studio
[√] VS Code, 64-bit edition (version 1.57.1)
[√] Connected device (2 available)

• No issues found!
```

3. Navigate to C:\Users\*yourprofilename*\AppData\Local\Android\Sdk\cmdline-tools and copy the path.

4. Navigate to System Properties (Windows 10 click settings and type "environment variables").


env
Edit environment variables for your account
Edit the system environment variables
System
Display, sound, notifications, power
Devi
Bluetooth, printers, mouse
Link your Android, iPhone
Network & Internet
Wi-Fi, airplane mode, VPN
Personalization
Background, lock screen, colors
Apps
Uninstall, defaults, optional features
Accounts
Your accounts, email, sync, work, family
Time & Language
Speech, region, date
Gaming
Xbox Game Bar, captures, Game Mode
Ease of Access
Narrator, magnifier, high contrast
Search
Find my files, permissions
Privacy
Location, camera, microphone
Update & Security
Windows Update, recovery, backup

5. Under the Advanced tab, look for Environment Variables near the bottom of the screen and select it.


System Properties

Computer Name | Hardware | Advanced | System Protection | Remote

You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...

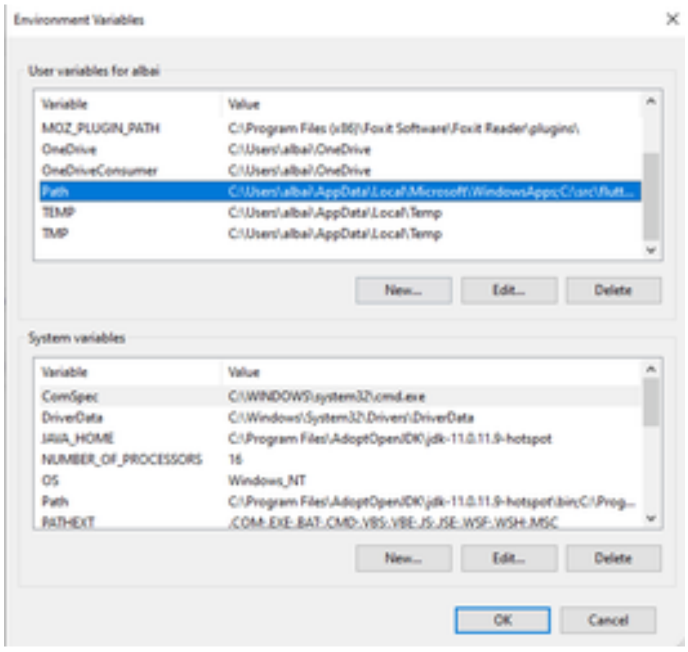
User Profiles
Desktop settings related to your sign-in
Settings...

Startup and Recovery
System startup, system failure, and debugging information
Settings...

Environment Variables...

OK Cancel Apply

6. Select “Path” under user variables and select Edit, as shown below, create a new path entry, with the path you copied from Step 3, C:\Users\*yourprofilename*\AppData\Local\Android\Sdk\cmdline-tools. Apply the change and exit.



Environment Variables

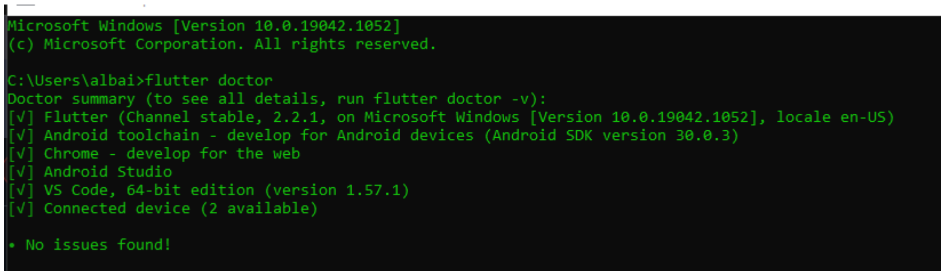
User variables for albai

Variable	Value
MOZ_PLUGIN_PATH	C:\Program Files (x86)\Foxit Software\Foxit Reader\plugins\
OneDrive	C:\Users\albai\OneDrive
OneDriveConsumer	C:\Users\albai\OneDrive
Path	C:\Users\albai\AppData\Local\Microsoft\WindowsApps;C:\src\flutt...
TEMP	C:\Users\albai\AppData\Local\Temp
TMP	C:\Users\albai\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot
NUMBER_OF_PROCESSORS	16
OS	Windows_NT
Path	C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin;C:\Prog...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC

7. Run Flutter Doctor again to verify that all items have checkmarks.



```
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Users\albai>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 2.2.1, on Microsoft Windows [Version 10.0.19042.1052], locale en-US)
[√] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[√] Chrome - develop for the web
[√] Android Studio
[√] VS Code, 64-bit edition (version 1.57.1)
[√] Connected device (2 available)

• No issues found!
```

4.2 Picovoice

1. To install Picovoice reference it in pubspec.yaml like any other package.

```
dependencies:
  picovoice: ^<version>
```

2. Picovoice needs permission to record audio from the hardware's microphone. On Android open AndroidManifest.xml and add this line.

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

3. For iOS, open inflo.plist and add this line.

```
<key>NSMicrophoneUsageDescription</key>
<string>[Permission explanation]</string>
```

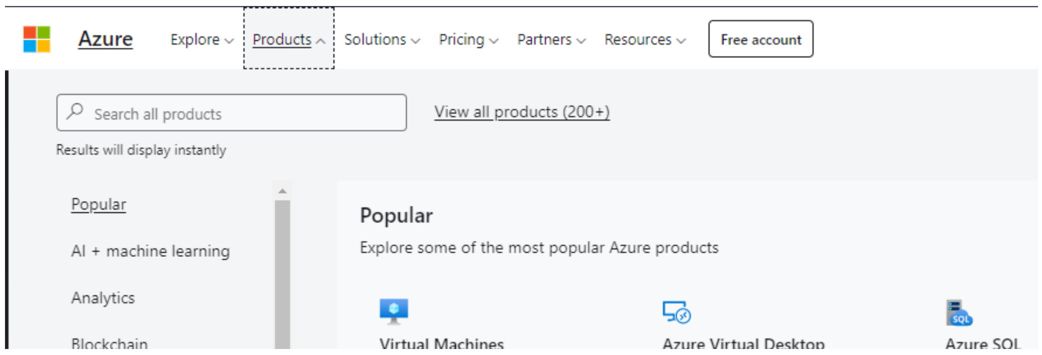
4. To create an instance of Picovoice Manager, needed for audio recording, use the PicovoiceManager.create constructor. For more information, please check the configuration section and the Programmer's Guide on how to create widgets using Picovoice.

```
import 'package:picovoice/picovoice_manager.dart';
import 'package:picovoice/picovoice_error.dart';

_picovoiceManager = PicovoiceManager.create(
    "/path/to/keyword/file.ppn",
    _wakeWordCallback,
    "/path/to/context/file.rhn",
    _inferenceCallback);
```

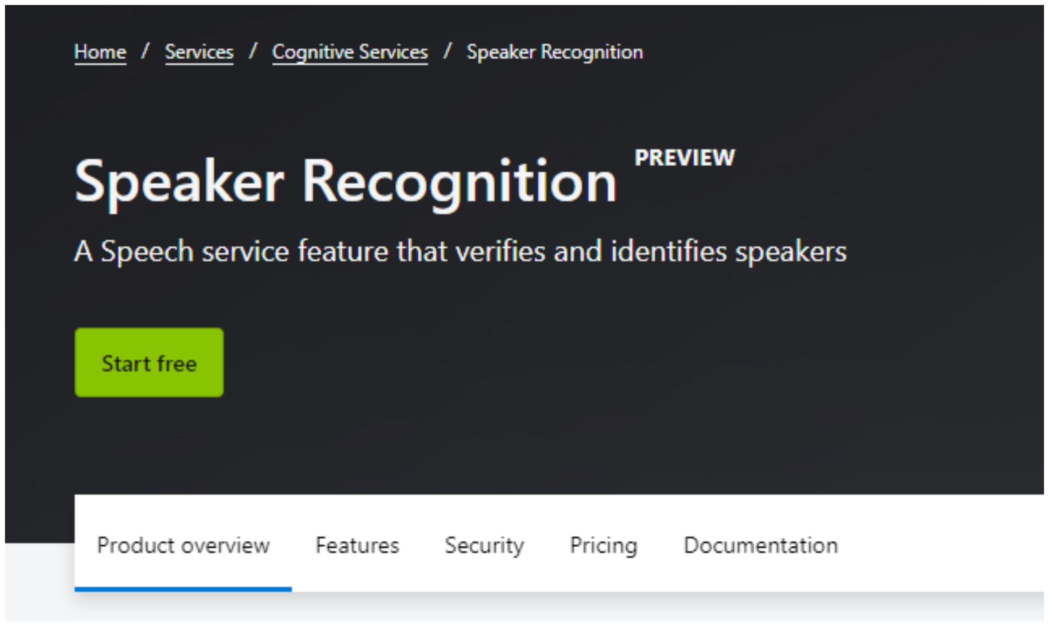
4.3 Azure Speech Recognition

Azure Speech Recognition will be accessed using REST API to create a profile of the user's voice. No packages need to be imported for the Harkify application. Please ensure that a valid Azure account is being used and that the license for the trial version have not expired. If a new account is required create one at https://azure.microsoft.com/en-us/ .

1. Creating an account will require a Microsoft account and a way to verify your identity, using a one-time code sent to your phone or using a credit card for verification, which ever method is chosen, a credit card is still required to create an Azure account, but does not cost anything for our purposes.
2. After creating an account select Product from the top and search for "Speaker Recognition,"

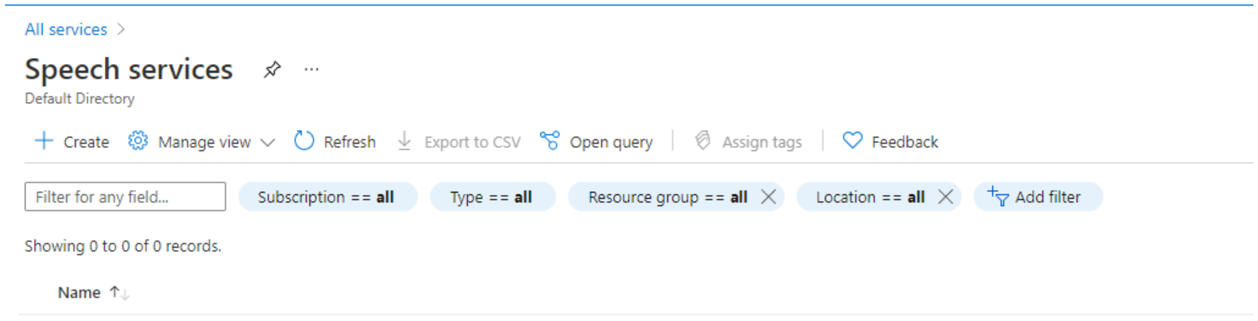


3. On the Speaker Recognition Page, get started by selecting Start Free. Note that documentation for this service can also be found here. On the next screen select Start Free again.


Home / Services / Cognitive Services / Speaker Recognition
Speaker Recognition PREVIEW
A Speech service feature that verifies and identifies speakers
Start free
Product overview
Features
Security
Pricing
Documentation

4. Choose between Pay as you go or use existing account, we chose to use existing account.
5. Now you will be sent to the Microsoft Azure portal, at the top search for "Speech Services.

6. At the Speech Services directory select Create.



All services >

Speech services

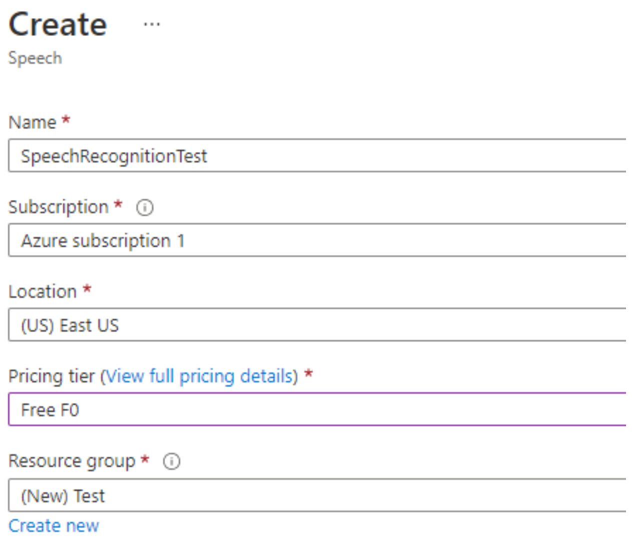
Default Directory

Filter for any field... | Subscription == **all** | Type == **all** | Resource group == **all** | Location == **all** | Add filter

Showing 0 to 0 of 0 records.

Name ↑↓

7. Fill out the fields to create the service, make sure to choose the free pricing tier currently.



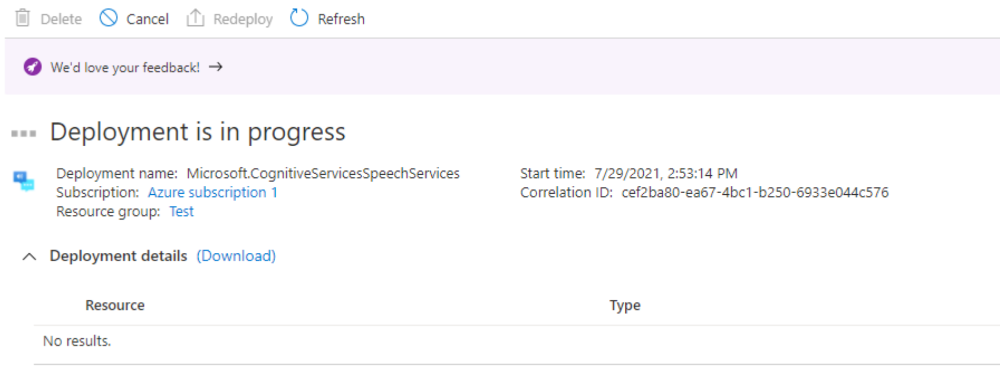
Create ...

Speech

Name *	SpeechRecognitionTest
Subscription *	Azure subscription 1
Location *	(US) East US
Pricing tier (View full pricing details) *	Free F0
Resource group *	(New) Test

Create new

8. Wait for deployment to complete.



We'd love your feedback! →

Deployment is in progress

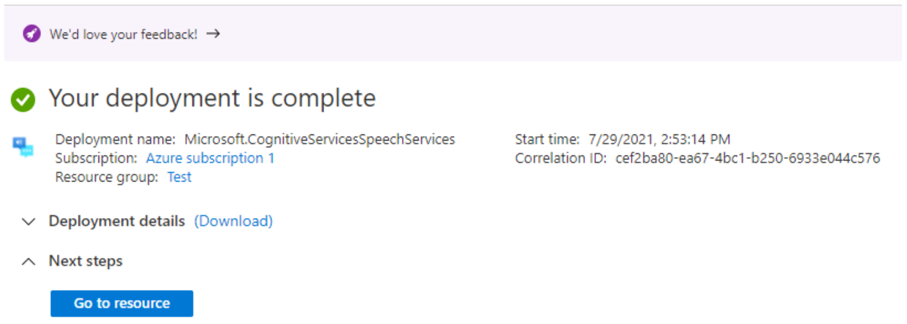
Deployment name: Microsoft.CognitiveServicesSpeechServices
Subscription: Azure subscription 1
Resource group: Test

Start time: 7/29/2021, 2:53:14 PM
Correlation ID: cef2ba80-ea67-4bc1-b250-6933e044c576

Deployment details (Download)

Resource	Type
No results.	

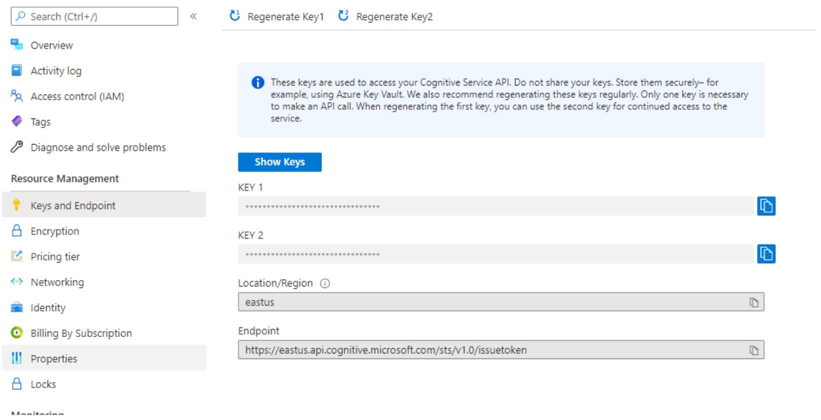
9. After deployment is complete a button titled “Go to resource” will appear, select it now.



Your deployment is complete

Deployment name: Microsoft.CognitiveServicesSpeechServices
Subscription: Azure subscription 1
Resource group: Test
Start time: 7/29/2021, 2:53:14 PM
Correlation ID: cef2ba80-ea67-4bc1-b250-6933e044c576

10. On the next screen select Keys and Endpoint, the keys and endpoint are how Harkify makes requests to Azure.



These keys are used to access your Cognitive Service API. Do not share your keys. Store them securely– for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

KEY 1: ••••••••••••••••••••••••••••••••

KEY 2: ••••••••••••••••••••••••••••••••

Location/Region: eastus

Endpoint: https://eastus.api.cognitive.microsoft.com/sts/v1.0/issuetoken

4.4 Git and GitHub

4.4.1 Installing Git

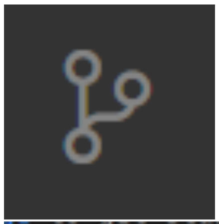
Git is the source control software used for development of the Harkify application. Git Bash is the command window for executing Git commands. Both are necessary for interacting with the application’s codebase.

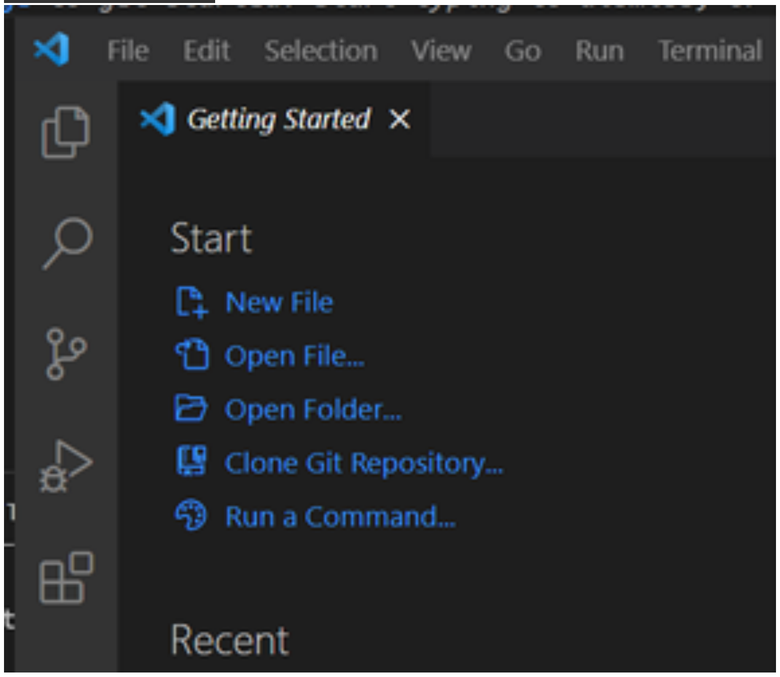
1. Download URL: https://git-scm.com/download/win .
2. Download the installation package from the above URL, running it to install Git and Git Bash.

4.4.2 Clone Repository

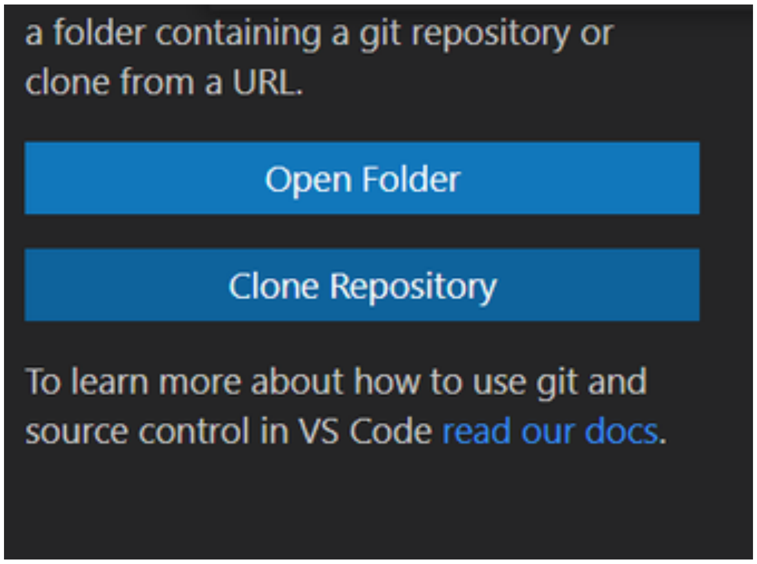
This guide covers how to clone the GitHub repository, create a new branch and push to main.

1. In VS Code look for this icon on the left-hand bar and click it, this is the Source Control menu.





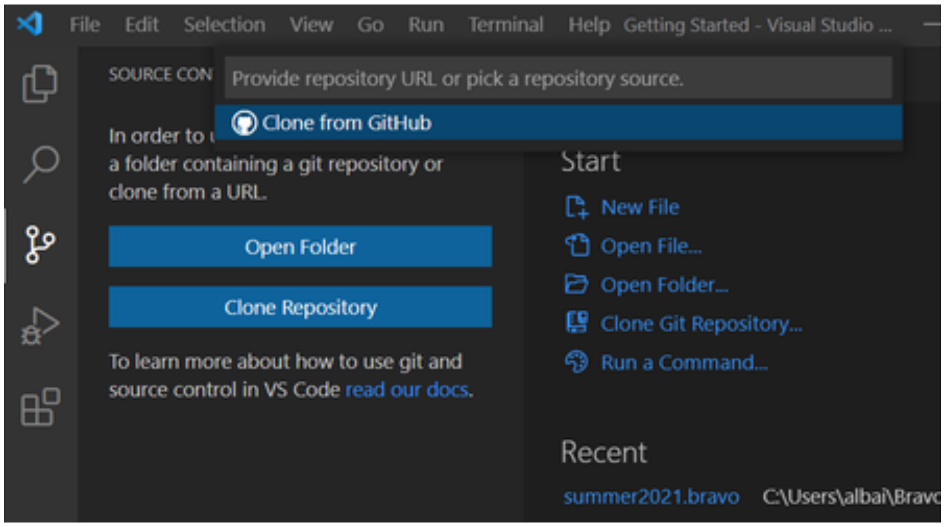
2. Select Clone Repository.



a folder containing a git repository or clone from a URL.

To learn more about how to use git and source control in VS Code read our docs.

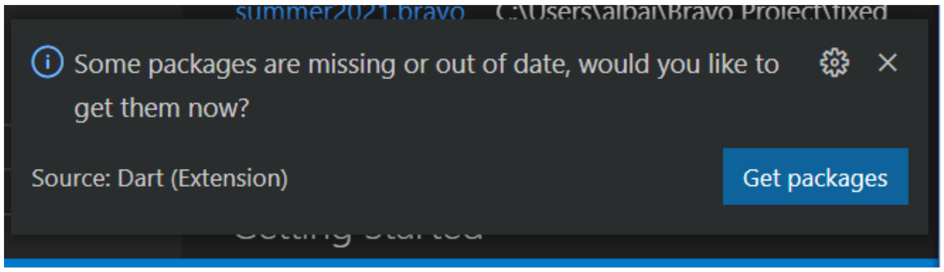
3. Enter the repository URL, currently https://github.com/umgc/summer2021.bravo, in the box near the top of the screen.



In order to a folder containing a git repository or clone from a URL.

To learn more about how to use git and source control in VS Code read our docs.

4. Choose a directory for the cloned repository, and choose to open the directory now, if this pop-up appears select Get Packages.



Some packages are missing or out of date, would you like to get them now?

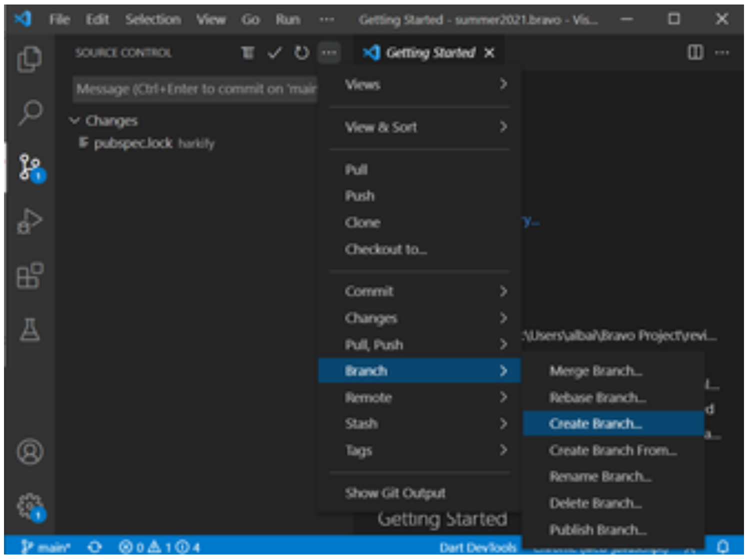
Source: Dart (Extension)

5. Examine the project especially if you missed the previous pop-up in there are numerous errors you may need to run "dart pub get" in a shell. Make sure to change directory to the folder in the project that contains pubspec.yaml.
6. If there are still numerous errors, you may need to migrate packages to null safety please consult https://dart.dev/null-safety/migration-guide for more information.

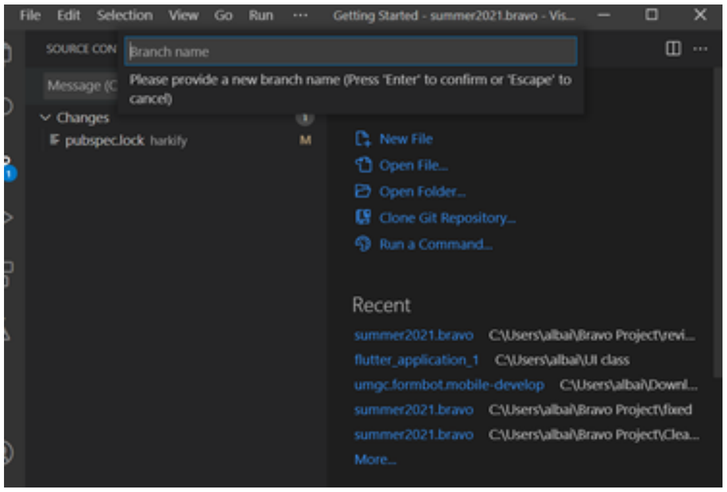
4.4.3 Create Branch

You should save your work to a branch until it is ready to be merged to main. This guide covers how to create a branch and commit changes.

1. Navigate to Source Control, click on the three dots, navigate to Branch, and select Create Branch.

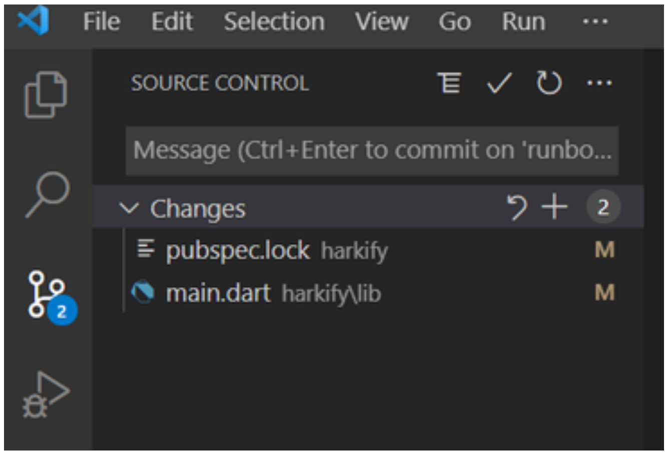


2. Enter the name for the branch and press enter.

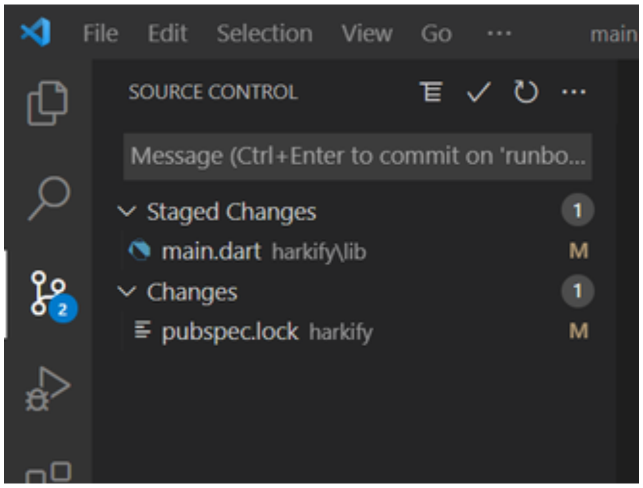

File
Edit
Selection
View
Go
Run
Getting Started - summer2021.bravo - Vis...
SOURCE CON
Branch name
Message (C
Changes
pubspec.lock harlify
M
New File
Open File...
Open Folder...
Clone Git Repository...
Run a Command...
Recent
summer2021.bravo C:\Users\albai\Bravo Project\revi...
flutter_application_1 C:\Users\albai\UI class
umgc.formbot.mobile-develop C:\Users\albai\Downl...
summer2021.bravo C:\Users\albai\Bravo Project\fixed
summer2021.bravo C:\Users\albai\Bravo Project\Clea...
More...

Please provide a new branch name (Press 'Enter' to confirm or 'Escape' to cancel)

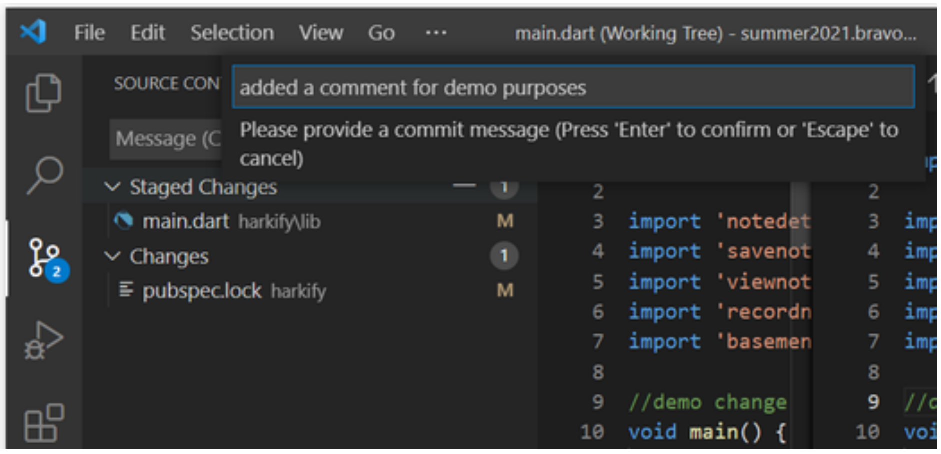
3. To publish changes to the branch, review the changes in Source Control.


File Edit Selection View Go Run ···
SOURCE CONTROL
Message (Ctrl+Enter to commit on 'runbo...
Changes 2
pubspec.lock harkify M
main.dart harkify\lib M

4. Review the changes and use the "+" to stage changes you want to publish in the branch.



5. Click on the checkmark to commit changes to the branch. Please be sure to provide a commit message.

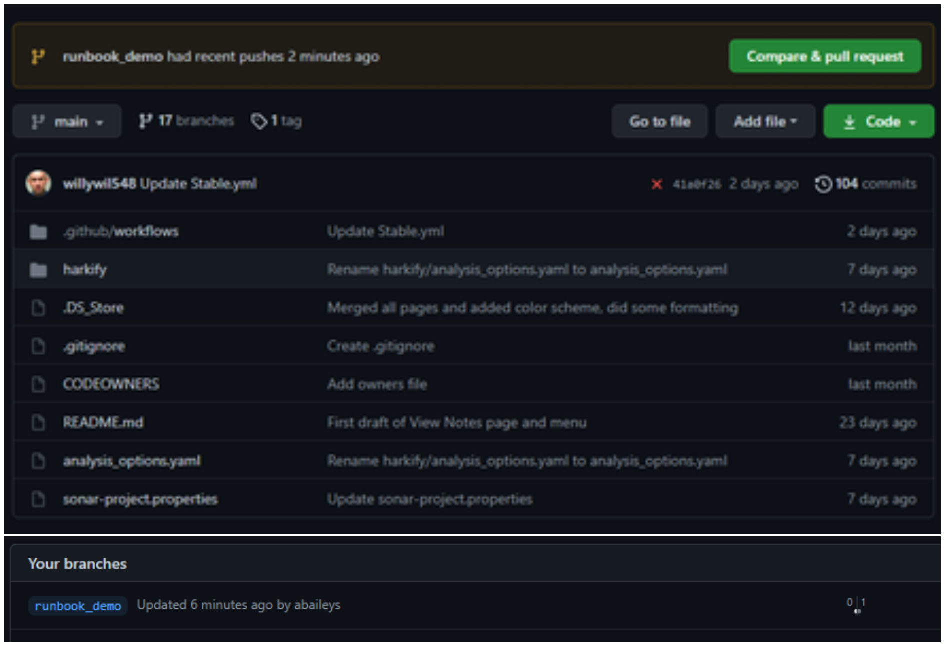


added a comment for demo purposes

Please provide a commit message (Press 'Enter' to confirm or 'Escape' to cancel)

```
import 'notedet
import 'savenot
import 'viewnot
import 'recordn
import 'basemen

//demo change
void main() {
```

6. Check your branches for your created branch for verification.


runbook_demo had recent pushes 2 minutes ago

willywil548 Update Stable.yml — 41a0f26 2 days ago — 104 commits

.github/workflows	Update Stable.yml	2 days ago
harkify	Rename harkify/analysis_options.yaml to analysis_options.yaml	7 days ago
.DS_Store	Merged all pages and added color scheme, did some formatting	12 days ago
.gitignore	Create .gitignore	last month
CODEOWNERS	Add owners file	last month
README.md	First draft of View Notes page and menu	23 days ago
analysis_options.yaml	Rename harkify/analysis_options.yaml to analysis_options.yaml	7 days ago
sonar-project.properties	Update sonar-project.properties	7 days ago

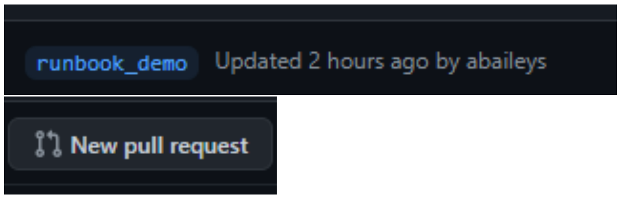
Your branches

runbook_demo Updated 6 minutes ago by abaileys 0 | 1

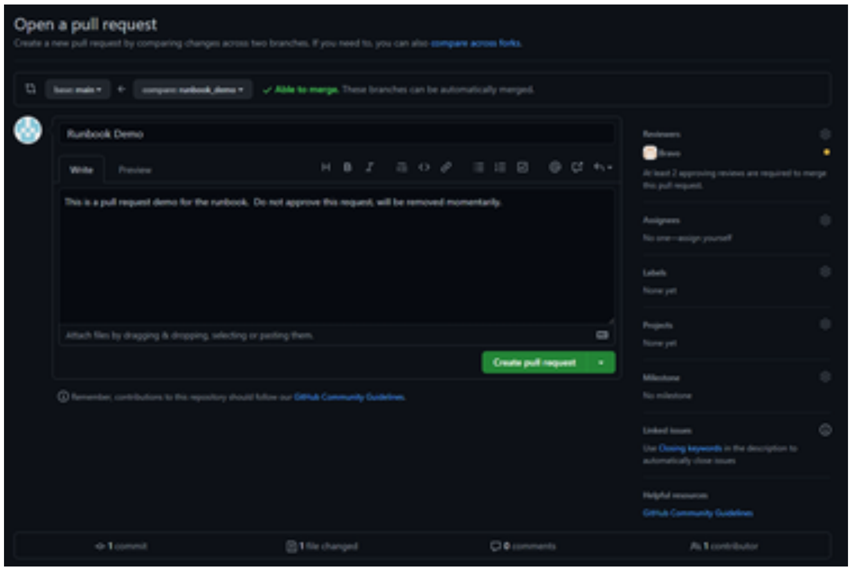
4.4.4 Create Pull Request

For main to be modified, a pull request is necessary. The pull request must be approved by a member of Bravo and DevSecOps.

1. Navigate to https://github.com/umgc/summer2021.bravo/branches and find the branch you wish to create a pull request for. Select "New Pull Request" on the right.


runbook_demo Updated 2 hours ago by abaileys

2. Give the pull request a name and description. Assign reviewers and then create the pull request. Check to see if you are merging the correct branches. Your branch with the latest changes should be the compare dropdown, and base is your target for the latest changes.



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: main ← compare: runbook_demo ✓ Able to merge. These branches can be automatically merged.

Runbook Demo

This is a pull request demo for the runbook. Do not approve this request, will be removed momentarily.

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Reviewers: Bravo. At least 2 approving reviews are required to merge this pull request.

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

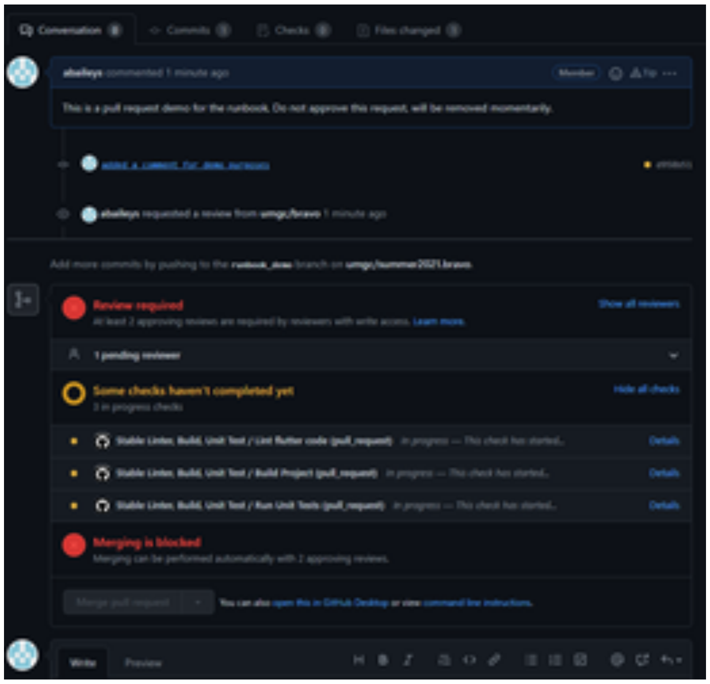
Milestone: No milestone

Linked issues: Use Closing keywords in the description to automatically close issues

Helpful resources: GitHub Community Guidelines

1 commit · 1 file changed · 0 comments · 1 contributor

3. After the pull request view the progress of the pull request by selecting Pull requests from https://github.com/umgc/summer2021.bravo/. Note that the request requires two reviews, one from Bravo and one from DevSecOps.



Conversation · Commits · Checks · Files changed

abaileys commented 1 minute ago

This is a pull request demo for the runbook. Do not approve this request, will be removed momentarily.

abaileys requested a review from umgc/bravo 1 minute ago

Add more commits by pushing to the runbook_demo branch on umgc/summer2021.bravo.

Review required

At least 2 approving reviews are required by reviewers with write access. Learn more.

1 pending reviewer

Some checks haven't completed yet

3 in progress checks

Stable Linter, Build, Unit Test / Lint flutter code (pull_request) In progress — This check has started... Details

Stable Linter, Build, Unit Test / Build Project (pull_request) In progress — This check has started... Details

Stable Linter, Build, Unit Test / Run Unit Tests (pull_request) In progress — This check has started... Details

Merging is blocked

Merging can be performed automatically with 2 approving reviews.

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

5. Known Bugs

All known issues and bugs with Harkify and the tools and services required by Harkify will be listed in this section.

- Harkify currently only runs on SDK versions under 30, with 28 being used in the current build.
- Some Flutter packages have been created before Dart 2.0 do not support null safety, please consult https://dart.dev/null-safety/migration-guide for how to migrate packages to null safety, better yet choose packages that already support null safety.

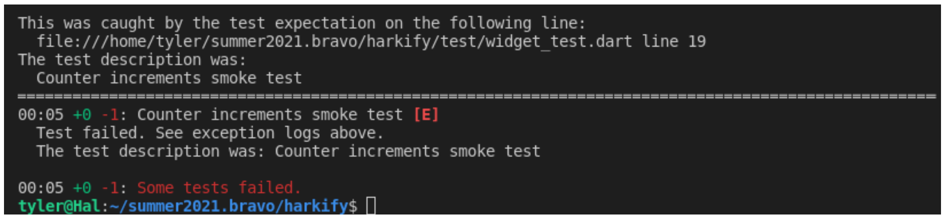
6. Testing

No additional software is required for testing the Harkify application. Flutter & Dart are used for all automated tests, and manual tests may be executed from either an emulated Android device, or from a physical device.

To execute unit and widget tests from the terminal, run the following command from the "harkify" directory in the project:

```
flutter test test/widget_test.dart
```

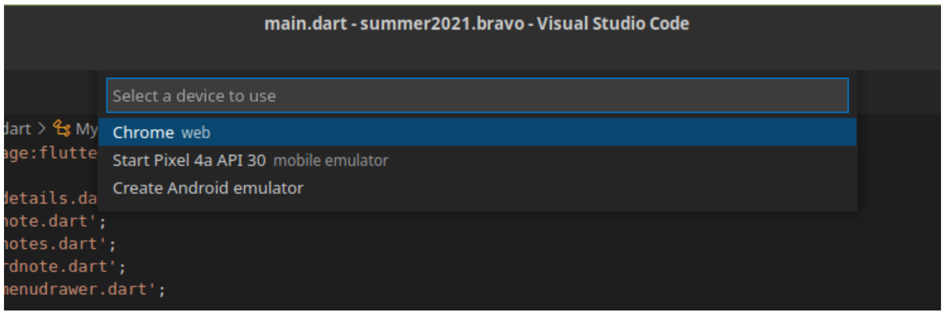
Output example:



```
This was caught by the test expectation on the following line:
  file:///home/tyler/summer2021.bravo/harkify/test/widget_test.dart line 19
The test description was:
  Counter increments smoke test
====================================================================================================
00:05 +0 -1: Counter increments smoke test [E]
  Test failed. See exception logs above.
  The test description was: Counter increments smoke test

00:05 +0 -1: Some tests failed.
tyler@Hal:~/summer2021.bravo/harkify$ 
```

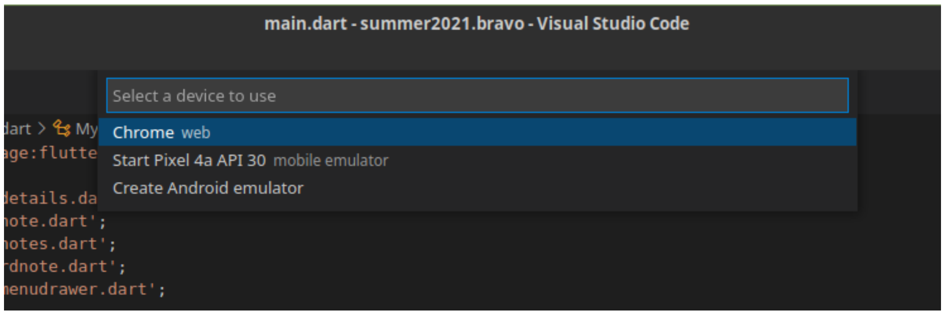
To test the application from an emulator or physical device, from VS Code:

1. Select the appropriate Device:



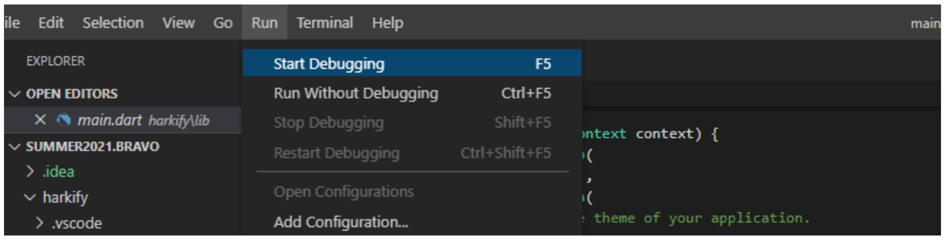
main.dart - summer2021.bravo - Visual Studio Code
Select a device to use
Chrome web
Start Pixel 4a API 30 mobile emulator
Create Android emulator

2. Wait for the device to start and connect:


main.dart - summer2021.bravo - Visual Studio Code
Select a device to use
Chrome web
Start Pixel 4a API 30 mobile emulator
Create Android emulator

```
dart > My
age:flutte

details.da
note.dart';
notes.dart';
dnote.dart';
menudrawer.dart';
```

3. Select Run > Start Debugging (or press F5).



```
ntext context) {
(
,
(
 theme of your application.
```

Once the application is launched on the emulator or device, testing may be conducted.

Appendix A: Acronyms and Abbreviations

API – Application Programming Interface

JDK – Java Development Kit

NLU – Natural-language understanding

PR – Pull Request

REST – Representational state transfer

SDK – Software Development Kit

SSN – Social Security Number

VS – Visual Studio

Appendix B: References

Picovoice Platform—Flutter API. (2021). Picovoice. https://picovoice.ai/docs/api/picovoice-flutter/

trevorbye, DCtheGeek, & v-demjoh. (2020, September 2). *Speaker Recognition overview—Speech service—Azure Cognitive Services*. Microsoft. https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/speaker-recognition-overview