Xavier Vercruysse – Mélissa Fontesse

Deploying ProxiCall

How-to guide

Table of Contents

[ProxiCall CRM 2](#_Toc9001479)

[Database 2](#_Toc9001480)

[Sendgrid 2](#_Toc9001481)

[Azure Active Directory 2](#_Toc9001482)

[Web app 3](#_Toc9001483)

[ProxiCall Bot 4](#_Toc9001484)

[Luis 4](#_Toc9001485)

[Web app bot 4](#_Toc9001486)

[Microsoft Teams 5](#_Toc9001487)

[ProxiCall Directline 6](#_Toc9001488)

[Cognitive Speech 6](#_Toc9001489)

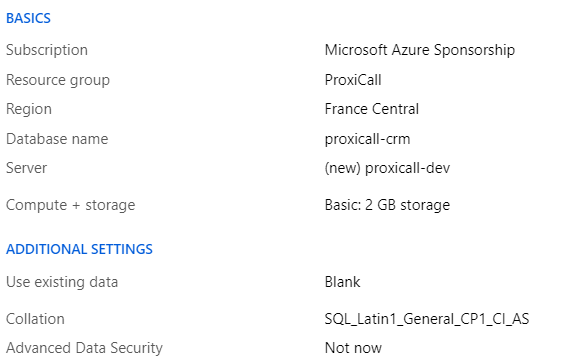
[Twilio 6](#_Toc9001490)

[Web app 6](#_Toc9001491)

# ProxiCall CRM

## Database

Create a new *sqlserver* database on Azure.



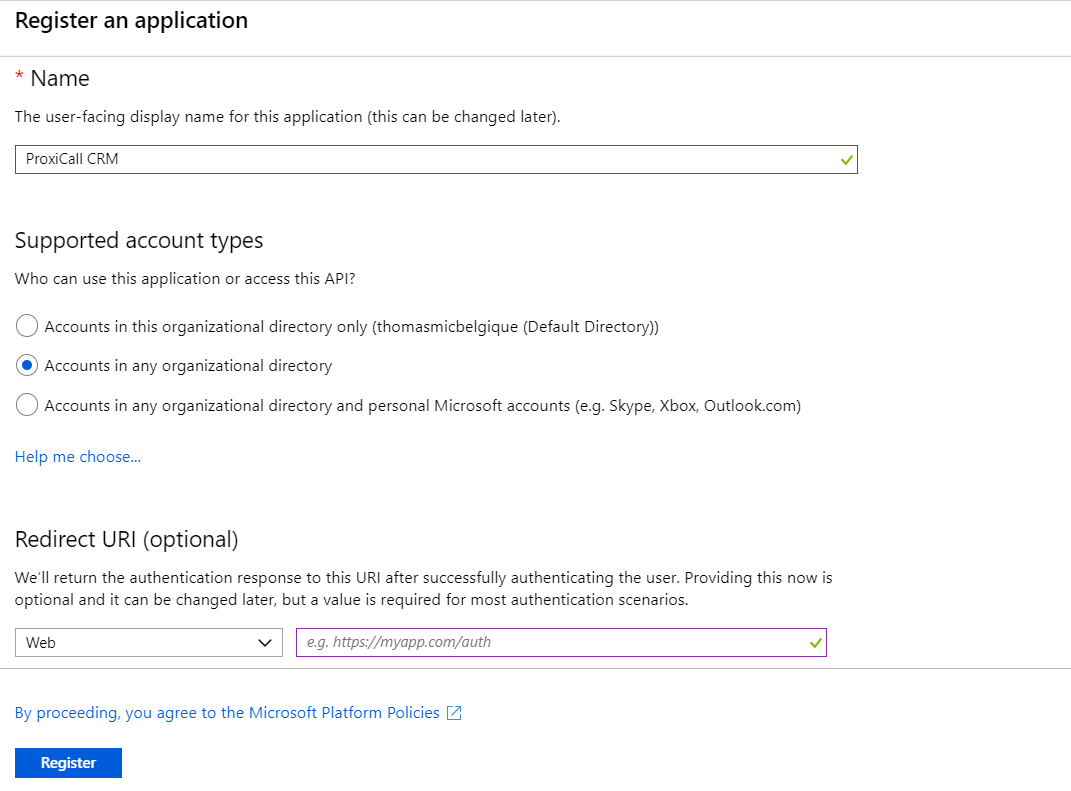
When it is created, retrieve the connection string from the *Overview* page, insert your username and password and save it for later.

## Sendgrid

If you don’t have one already, create an account on [Sendgrid](https://sendgrid.com) and follow the instructions to obtain an api key. Save it for later.

## Azure Active Directory

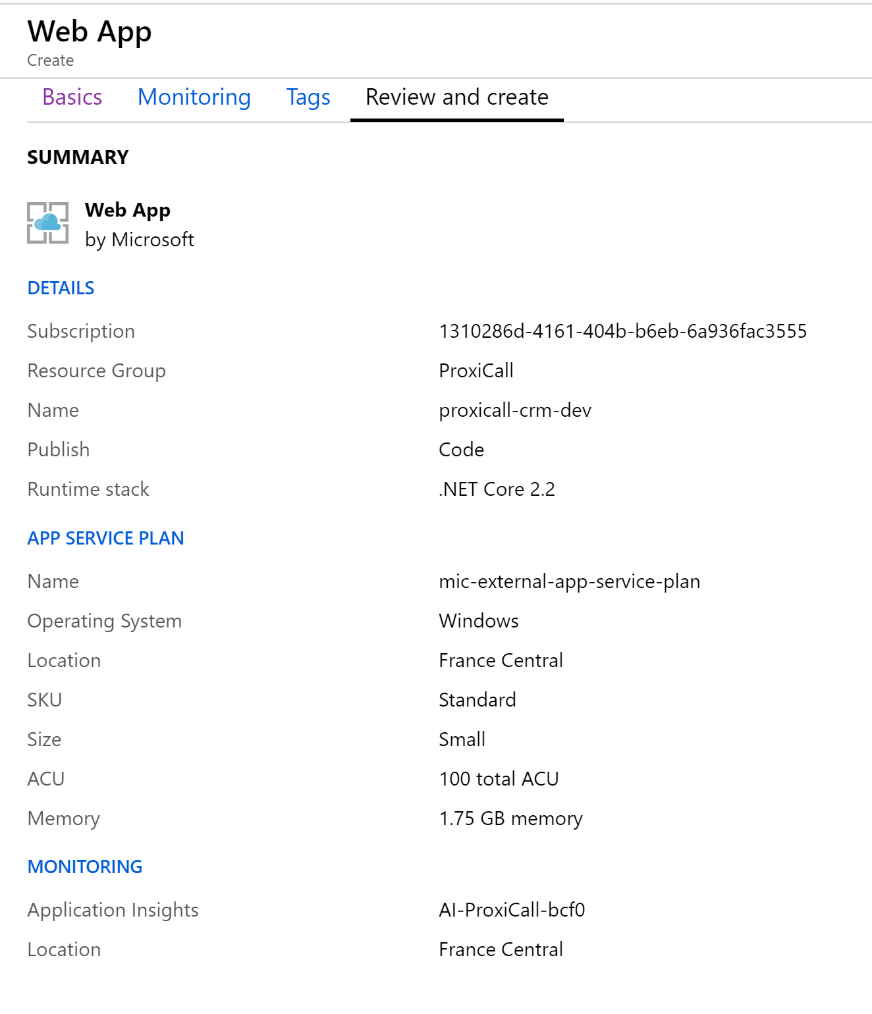
In your Azure Active Directory, register a new app. Leave the redirect uri empty for now.



When the app is registered, go on the *Certificates & Secrets* tab and create a new *Client secret*. Copy the newly created secret and save it for later, you won’t be able to see it afterwards.

## Web app

Create a web app resource on Azure.



Fill in the following settings in your *appsettings.json*, as well as in the *Configuration* tab of the CRM web app on Azure.

The *UserSettings* section represents the default admin account of the CRM.



Next, execute the *Update-Database* command in the *Package Manager Console*.

You can now publish the project on Azure.

# ProxiCall Bot

## Luis

Create a new *Language Understanding (LUIS)* resource. When it is created, save the api key for later and go on [luis.ai](eu.luis.ai) and sign-in with your Azure account.

Create a new luis application for each culture you want to support (only the en-US and fr-FR cultures are supported as of right now).

When the application is created, go on the *Manage* tab, then *Keys and Enpoints* and assign the azure resource you created earlier.

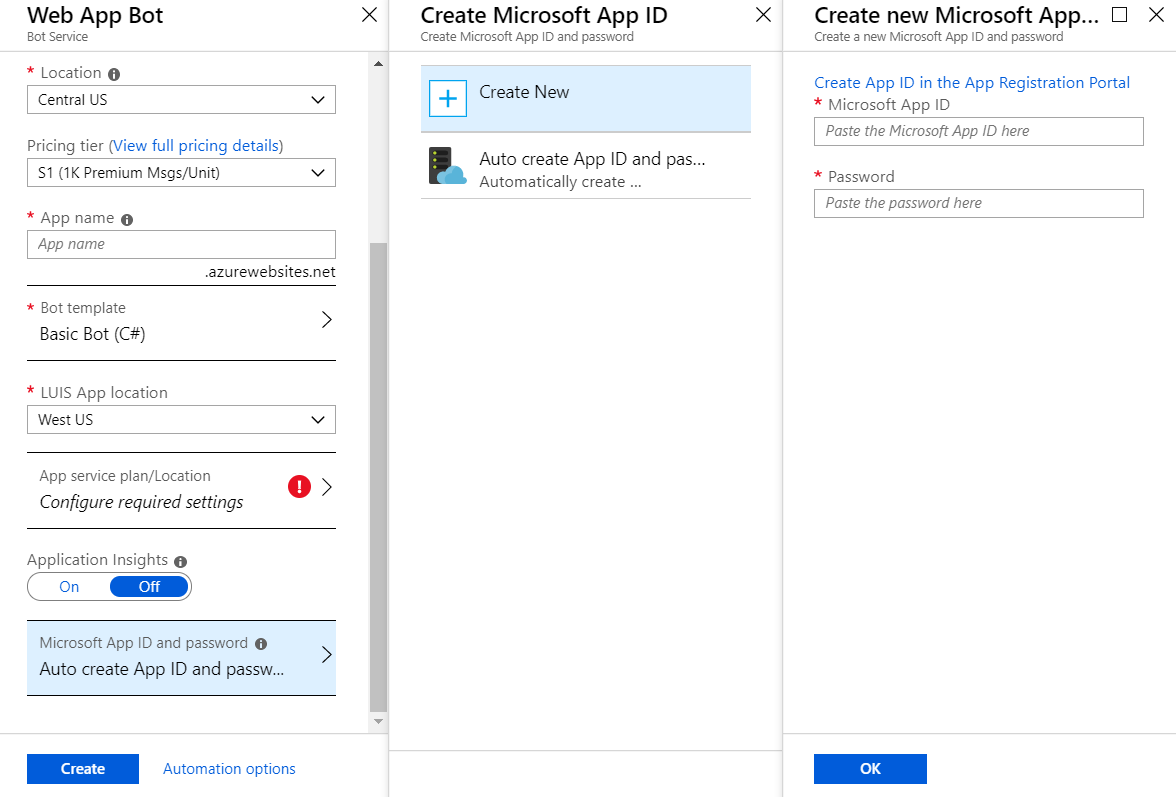
On the *Application Information* tab, save the ApplicationId for later

To add the model in the application, go on the *Versions* tab and import the file *proxicall-luis-LANGUAGE\_CODE-model.json*.

Finally, train and publish the app.

## Web app bot

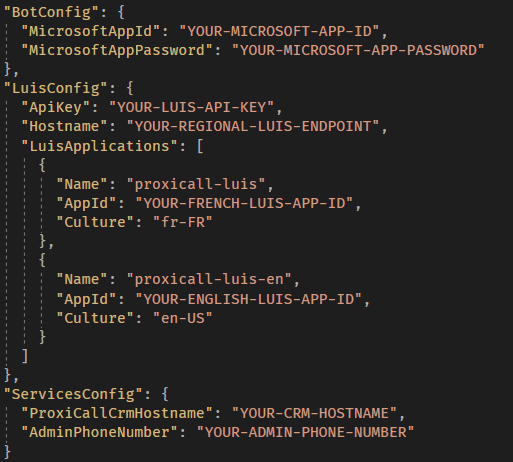
Create a new *Web App Bot* resource on azure.



Pick the *Echo Bot* template and select the *Auto Create* option in the *Microsoft App ID and password* field. Save the MicrosoftAppId and MicrosoftAppPassword added as *Application Settings* for later.

Go on the *Web app bot*, in the *Channel* tab and add the *Directline* channel. Save the Directline secret for later. You can also add the *Teams* channel for messaging.

Fill in the following settings in your *appsettings.json*, as well as in the *Configuration* tab of the CRM web app on Azure.



You can now publish the bot on Azure.

## Microsoft Teams

Go on the Teams App Store and install *App Studio*. When it is installed, go on the *Manifest Editor* tab and create a new app. Fill in the different fields, then go the bot tab and add an existing bot.

Add the Microsoft App Id that you saved earlier and select the personal scope.

Finally, scroll to the *Test and distribute* tab and install the app.

You can now chat with ProxiCall on Teams.

N.B.: You need to login at least once with your Microsoft Teams account on your CRM to be authenticated on the Teams bot.

# ProxiCall Directline

## Cognitive Speech

Create a new *Speech* resource on Azure. When it is created, retrieve your api key and save it for later, as well as the region.

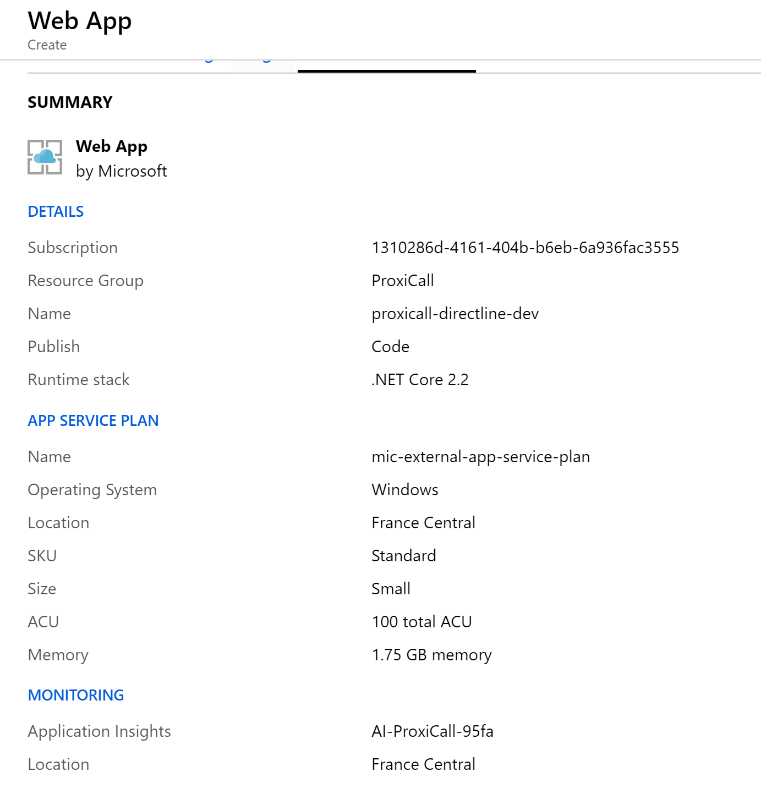
## Twilio

If you don’t have one already, create a new account on [Twilio](https://www.twilio.com) and create a new project with a *Phone Number* product.

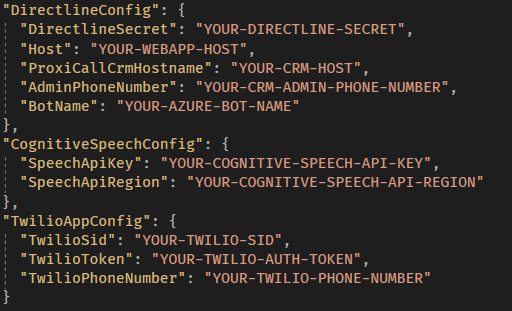
Finally, buy a phone number with *calling capabilities* and save it for later, along with your Twilio SID and Twilio Password (auth token).

## Web app

Create a new *Web App* on Azure.



Fill in the following settings in your *appsettings.json*, as well as in the *Configuration* tab of the web app.



You can now publish the web app.