从零开始，轻松打造你的聊天机器人

我们即将进入对话及平台的时代，Conversation as a Platform，简称 CaaP，中文应该叫做“对话即平台”。不管是一个屏幕，一个桌子，一个冰箱，甚至是人，都是一个 app，都是一个对话的窗口，所有的交互都通过对话来完成。起床，你用自然语言唤起梳妆台的镜子显示今日的天气以及新闻。家里的小娜（Cortana）提醒你要赶紧出门，因为限号 4、9，根据历史的预测，路上会提早堵车，需要提前 20 分钟出发。

那这个时代，将是如何实现的呢？人工智能不是简单一个团队，一小段时间就能完成的，需要经过大量的科研人员，进行大量的研究才能实现的。那怎么样才能实现物物智能，智能众生的构想呢？微软将微软研究院多年的研究技术开放出来，让各个开发者，不管是大公司，小公司甚至是个人都可以借助”智能”的”洪荒之力”，实现智能对话平台的未来。

做一个 Bot 机器人主要涉及到两块东西：
1. 对话管理、会话转移等跟会话相关的东西
2. Bot 的智能能力，包括语义的理解、命令实体的识别、语音的识别等技术

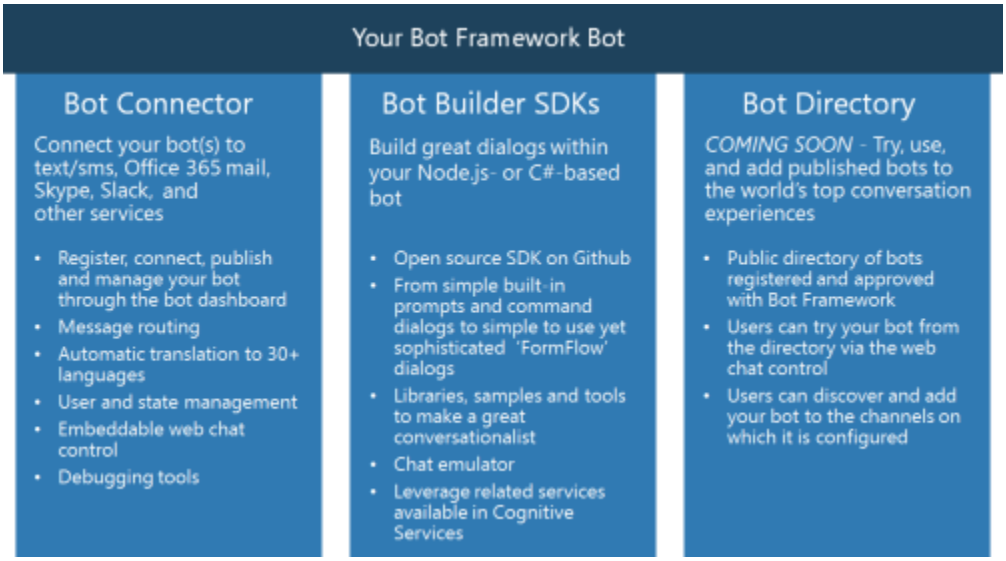
针对第一块东西，微软推出 Azure Bot Service（也叫做Microsoft Bot Framework），感兴趣的小伙伴可以打开网站，自己先钻研一下：https://azure.microsoft.com/zh-cn/services/bot-service/

针对第二块东西，微软推出 Microsoft Cognitive Service（认知服务），同样，感兴趣的小伙伴可以打开网站，自己先钻研一下：https://azure.microsoft.com/zh-cn/services/cognitive-services/directory/

不想钻研的小伙伴也没关系啦，接下来我将为大家概括这两块东西的核心内容。

Bot Framework

Bot Framework 就是帮你解决会话管理、会话转移等跟会话相关的事情，快速创建一个聊天机器人的后端，在各种终端和消息平台上提供服务。包括三大组件。



Bot Builder SDKs：

这个是 Bot 的生成器，快速生成一个 ASP.NET 和 Node.js 的后端服务，提供了像 Dialog、FormFlow 帮你管理与用户的会话。

Bot Connector：

这是个 Bot 的 Channel，帮你把你的服务快速发布到各个渠道，比如说 Skype，Facebook Messager 等等。这样用户就可以在 Skype 等 Channel 上使用你的服务了。

Bot Directory：

这个算是 Bot 的商店，在这里可以找到各个 Bot，你也可以把自己的 Bot 发布出来，从而大家都可以看到你的 Bot。

认知服务

认知服务是微软将研究院研究的技术以 API 和 SDK 的形式开放给开发者的一系列智能化服务。主要包括 5 大类的服务：视觉、语言、语言、知识和搜索。

这里面有好多特别好玩的服务，大家后面可以尝试去玩。本场 Chat 主要跟大家聊聊语言底下的“语言理解（Language Understanding Intelligent Service，简称 LUIS）”服务。这个服务主要是来做自然语言的意图识别和命名实体识别（NER）。

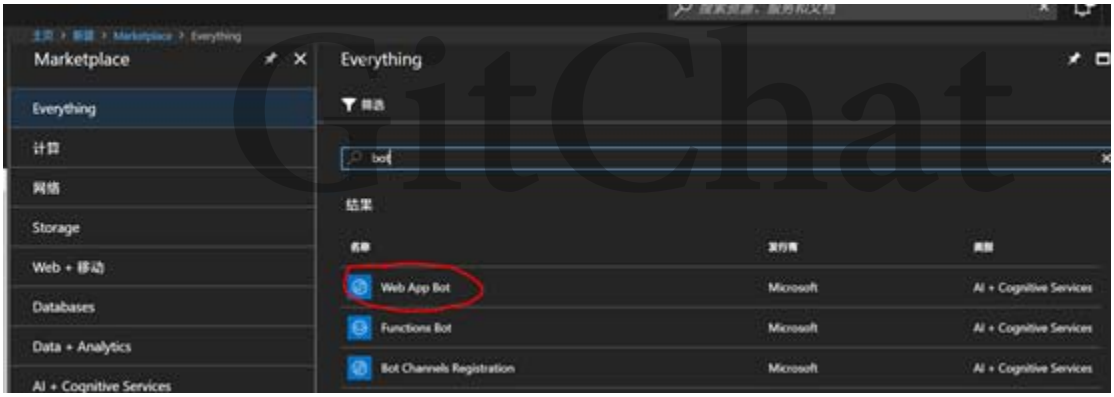
了解了以上基本原理，我们开始进入实战篇！

环境准备

- 下载 Visual Studio 2017
- 免费获取 Azure 全球版订阅 （https://azure.microsoft.com/zh-cn/free/ ）
- 使用微软账号注册 LUIS 网站（https://www.luis.ai/ ）

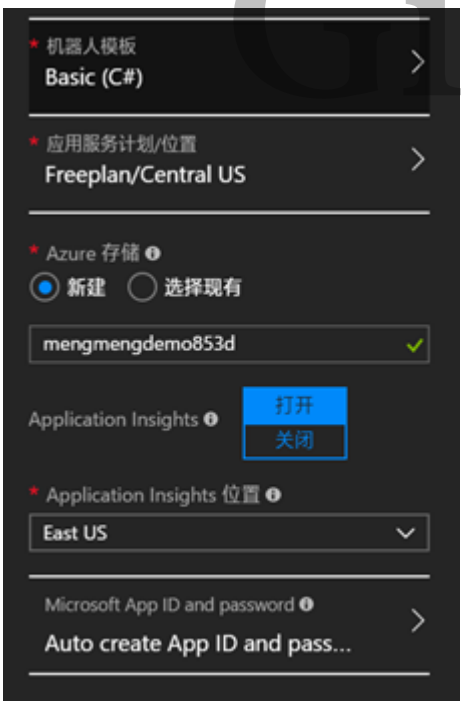
在 Azure 上创建 Bot 服务

咱们现在创建一个 Bot Application 的应用。我准备创建一个叫做”萌萌”的机器人。在 Azure 上创建 Bot 服务：搜索 Bot，选择”Web App Bot”。



点击 创建 ， 填写相关的信息：

选择基本模板即可，其他默认选项即可。



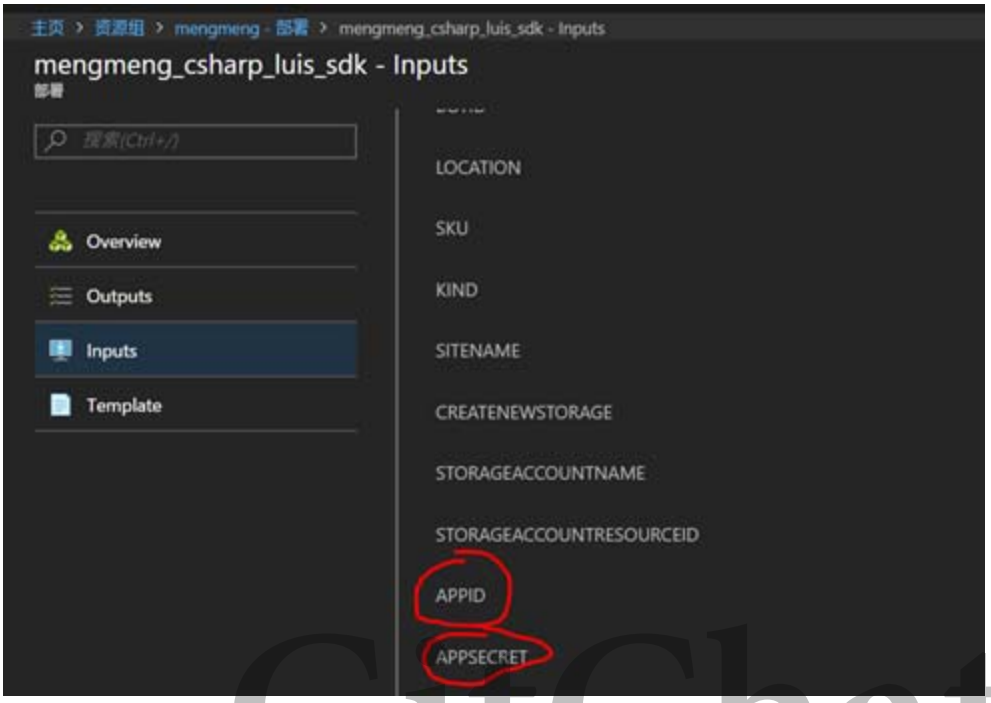
配置完成后，点击 创建 ，等待几分钟即可创建完成。

创建完成你可看到 Bot 服务帮你创建了 5 个相关服务。

点击第一个 Web App Bot 服务，即可配置 Bot 相关服务。

- Name：你的 bot 的名字，比如我的叫做“萌萌”。
- 句柄（Bot Handle）：比如我的写“mengmeng”，其实就是你的 Bot 的 id，记下来。
- Description：你的 Bot 的描述，会在你的 publish 之后主页上显示。
- 点击 信道，其实就是配置各个平台，里头可以看到“Web Chat”，这个是网页端的 channel，已经帮你写好的一个 iframe，咱们点击 `Edit`。
- 生成 Web Chat 的密钥之后，把密钥复制，点击 `I'm done configuring Web Chat`。

在资源组中，选择你创建的资源组，点击 部署：



请记下 APPID 和 AppSecret 两个参数，后面会用到。

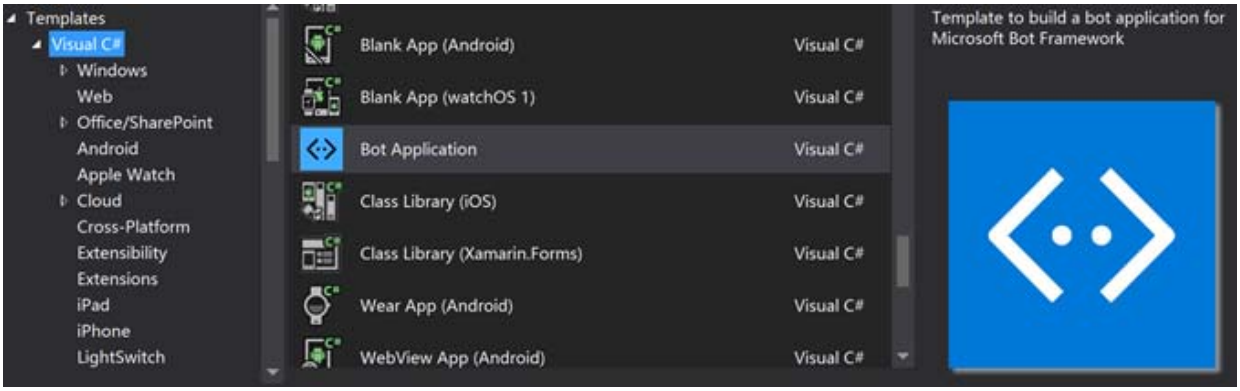
下载 Bot Framework 的 SDK

首先，请下载 Bot Framework 的 SDK，建议下载 Bot Framework 的 Visual Studio 的模板 Bot Application。

下载下来的模板（不用解压）请直接放置到

```
C:\Users\你的用户名\Documents\Visual Studio
2015\Templates\ProjectTemplates\Visual C#
```

下面，重新打开 VS 这样你在 C# 下面就可以看到有 Bot Application 的模板了。



如果是使用 NuGet 来下载 SDK，请参考：

1. 右键你的 C# 项目，选择“Manage NuGet Packages”。
2. 在 `Browse` 的 tab 页，输入“Microsoft.Bot.Builder”。
3. 点击 `Install` 的按钮。

如果你使用的是 Node.js，请使用以下命令：

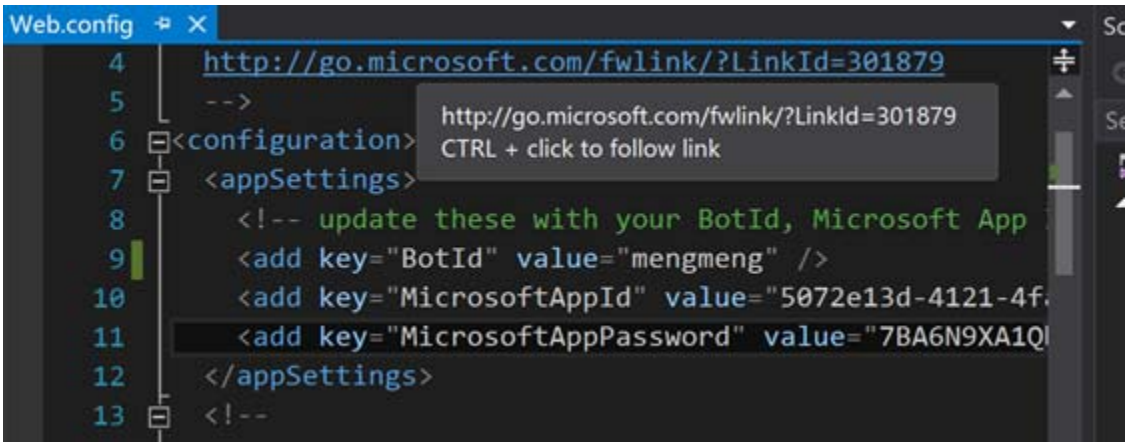
```
npm install botbuilder
```

准备你的机器人后端服务代码

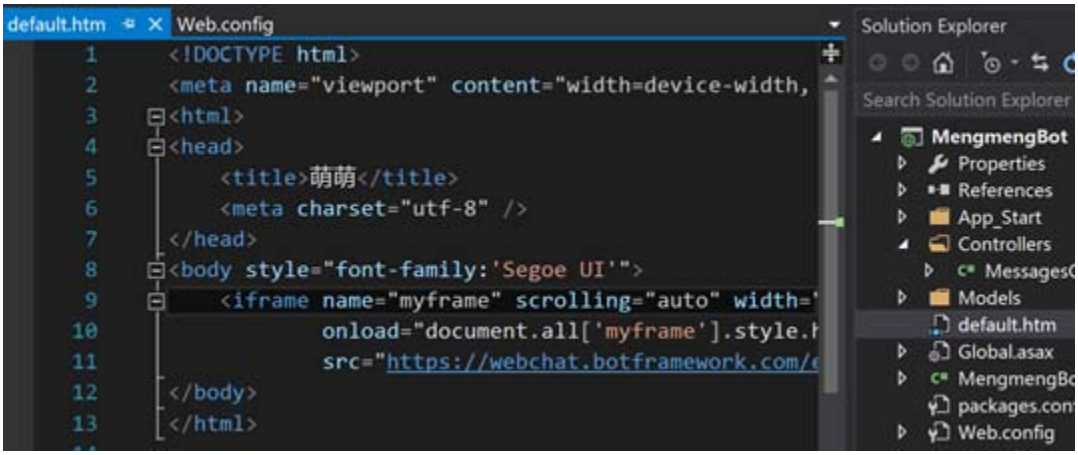
从我的 GitHub 地址下载代码：

> https://github.com/cheneyszp/MengmengBot

在你的 Web.config 里，填上你的 BotId，刚才创建的 App ID 和 AppSecret。



打开网站的起始页 default.htm：



复制以下代码：

```
    <!DOCTYPE html>
    <meta name="viewport" content="width=device-width, initial-
scale=1, maximum-scale=1">
    <html>
    <head>
    <title>萌萌</title>
    <meta charset="utf-8" />
    </head>
    <body style="font-family:'Segoe UI'">
    <iframe name="myframe" scrolling="auto" width="100%"
height="100%"

onload="document.all['myframe'].style.height=myframe.document.bod
y.scrollHeight"
    src="https://webchat.botframework.com/embed/你自己的Bot句柄?s=
你 自 己 的 Web Chat 密 钥 " style="height: 502px; max-height:
502px;"></iframe>
    </body>
    </html>
```

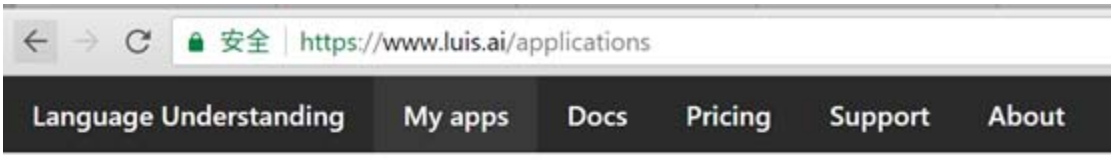

然后右键项目工程，点击 publish，之后你就可以拥有一个简单的 Bot 啦。

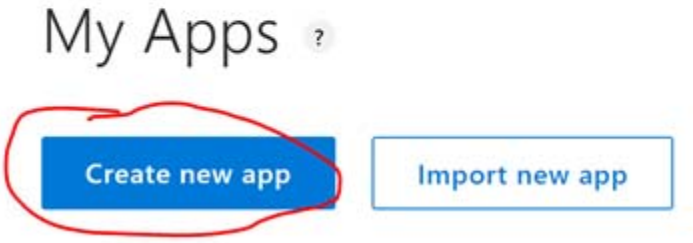
新建 LUIS 服务

请先登录：https://www.luis.ai/ ，这是语言理解服务的 portal。如果还没注册的话用 live id 注册以下就可以了。

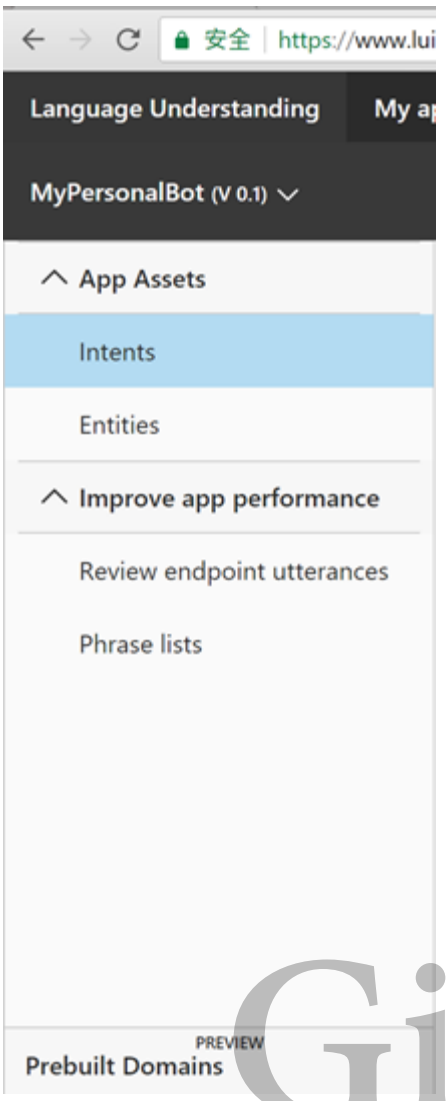
点击 App 这一栏，咱们先点击 new App 新建一个 app。

新建完成后，点击应用的名称，进入编辑这个应用。





我们先看以下左边的 tab，可以看到有仪表盘，意图（Intents），实体（Entities），功能等。



Intents：就是意图，比如咱们现在要提供天气查询的服务，那么咱们就创建一个“查询天气”的 Intent。

实体里头有两类：

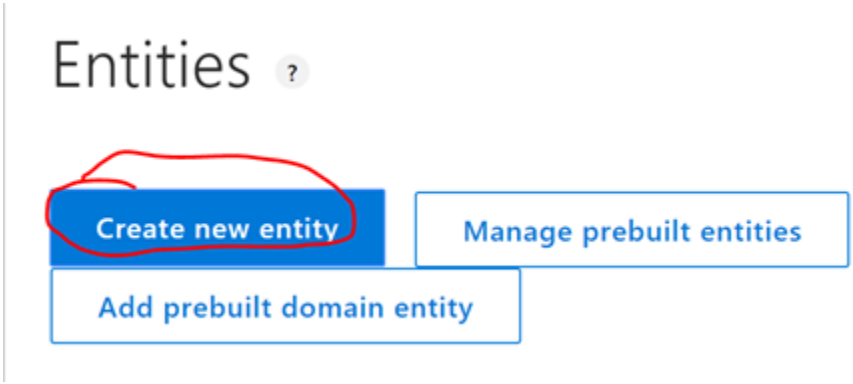
- Entities：实体，比如在查询天气的时候需要有地理位置信息，需要把用户的语言里头的地点提取出来，这个地点就是这个句子里头的实例，咱们创建一个”地点”的实例。
- Pre-built Entities（预建实体）：这个是预置好的实例，比如说时间，数字等等，我加了一个 datetime 的预置实例。

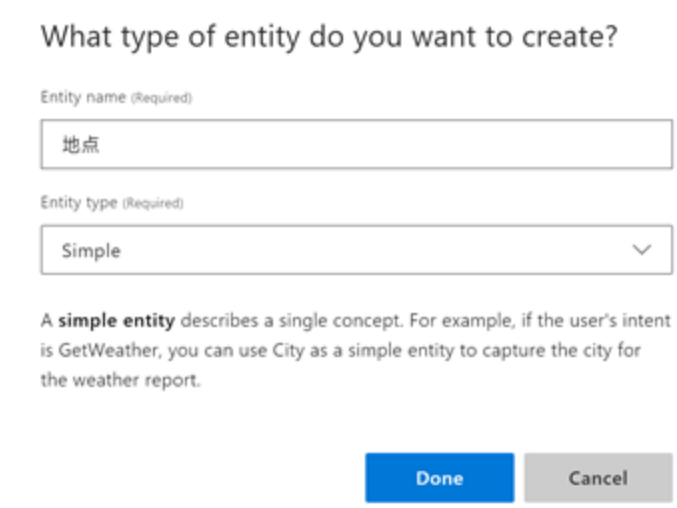
在功能里头会有：

- Phrase List（短语列表功能）：固定的一些短语，能够直接识别，比如说航空公司的名字等已知信息
- Pattern Features（模式功能）：正则表达式，可以匹配出相应的一些字段，比如说航班号。

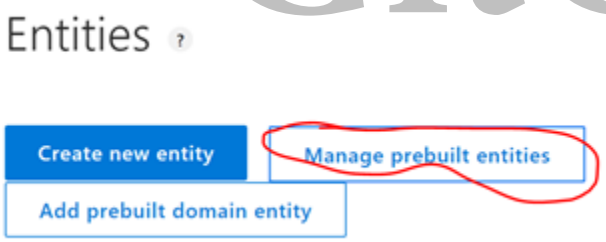
咱们现在来创建一个能够识别查询天气的语言理解服务。

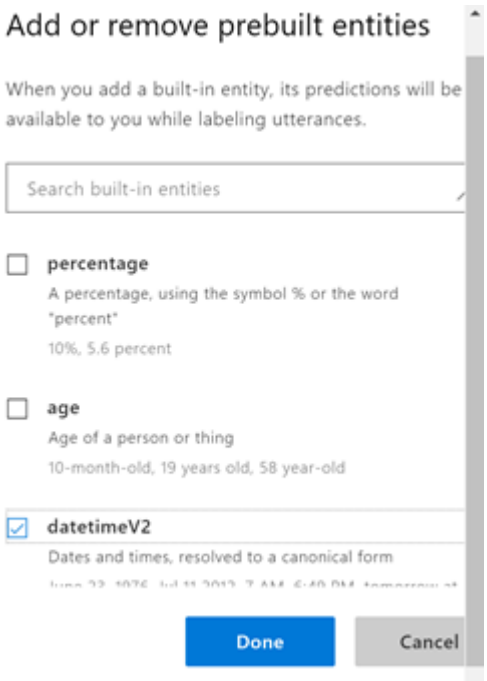
首先，查询天气需要地点信息，咱们先创建一个“地点”的实例。点击到 实体 里头， 添加自定义实体：



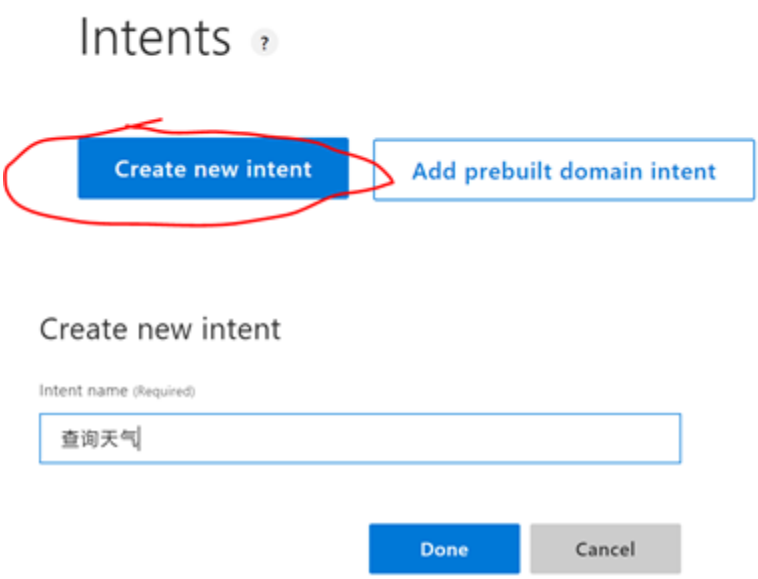


添加一个“预建实体” datetime：





再创建一个叫做“查询天气”的 Intent。

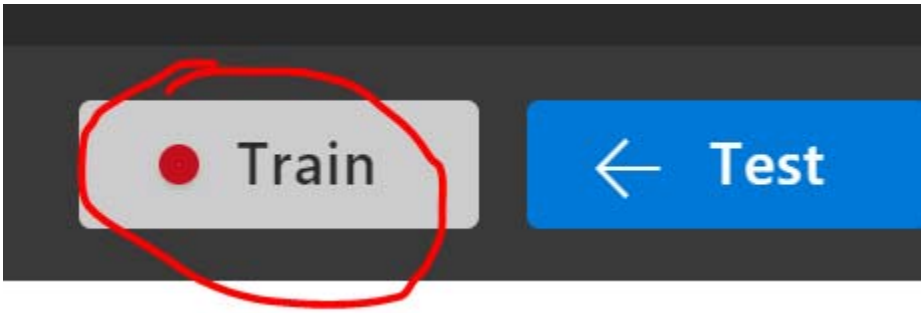


点击 Done 之后，点击 查询天气 会进入以下界面：

输入几个例子，比如说“北京天气怎么样”，可以多输入几个句子的类型，比如“北京今天有雾霾吗？”等等，每输入完一句按一下回车。

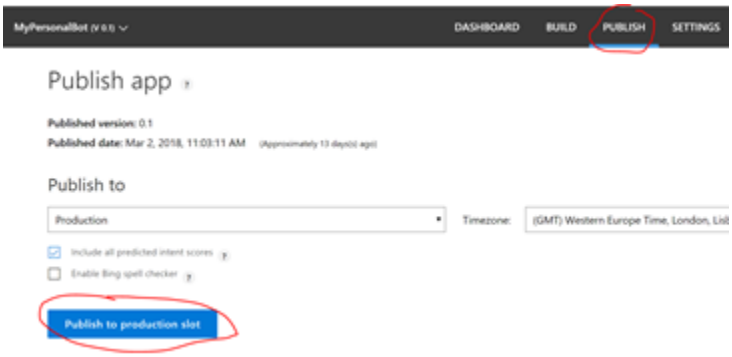
如果北京等地点信息没有显示标记的话，选中北京两个字，然后选择 地点 标注。然后点击 保存 。

随手点击该界面右上角的 Train 按钮进行训练。



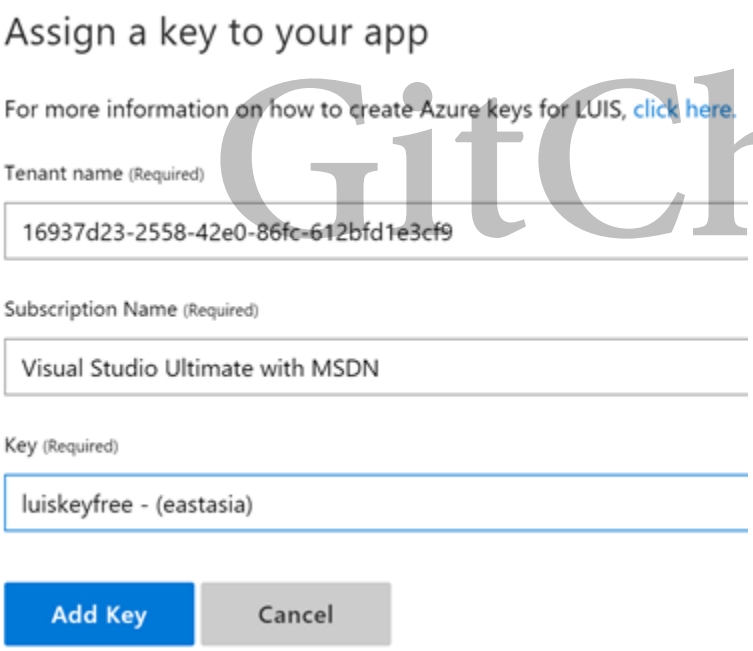
可以点击 Test 进行测试。

随手点击 Publish 的 tab 页面，进入到一下界面，点击 Publish to production slot：



点击 Publish 之后，你可以在 Publish 的 tab 页面底下，可以看到 Resources and Keys ，试用的话会给你自动生成一个 Starter Key，右边的链接就是 Publish 出来的链接地址啦。如：https://westus.api.cognitive.microsoft.com/luis/v2.0/apps/LUIS_APP_ID?subscription-key=LUIS_subscription-key&timezoneOffset=0.0&verbose=true&q=

如果商用的话，需要在 Azure Portal 创建一个 LUIS 服务，然后点击 Add Key ，把 LUIS 的 Key 添加进来。



url 后面加上查询语句就是 API 了。
https://westus.api.cognitive.microsoft.com/luis/v2.0/apps/LUIS_APP_ID?subscription-key=LUIS_subscription-key&timezoneOffset=0.0&verbose=true&q=北京今天天气怎么样?

把 LUIS 的 ID（就是 apps 后面那串字符）和 subscription-key 记下来，后面需要用到。在 URL 后面可以输入相关的语句，然后回车，就可以看到返回的 json 字符串了。

在 Bot 中集成 LUIS 服务

现在我们就把 LUIS 集成到你的 Web 服务里头来就可以了。咱们用 Dialog 的方式来完成，比较好管理会话。在 MessagesController.cs 里头的 public class MessagesController : ApiController 添加 MengmengDialog 的类。

```
    [LuisModel("你LUIS的ID", "你LUIS的subscription-key ")]
    [Serializable]
    public class MengmengDialog : LuisDialog<object>
    {
    public MengmengDialog()
    {
    }
    public MengmengDialog(ILuisService service): base(service)
    {
    }
    [LuisIntent("")]
    public async Task None(IDialogContext context, LuisResult
result)
    {
    string message = $"萌萌不知道你在说什么，面壁去。。。我现在只会查询股
票和查询天气。。T_T" + string.Join(", ", result.Intents.Select(i =>
i.Intent));
    await context.PostAsync(message);
    context.Wait(MessageReceived);
    }
    public bool TryToFindLocation(LuisResult result, out String
location)
    {
    location = "";
    EntityRecommendation title;
    if (result.TryFindEntity("地点", out title))
    {
    location = title.Entity;
    }
    else
    {
    location = "";
    }
    return !location.Equals("");
    }

    [LuisIntent("查询天气")]
    public async Task QueryWeather(IDialogContext context,
LuisResult result)
    {
    string location = "";
    string replyString = "";
    if (TryToFindLocation(result, out location))
```

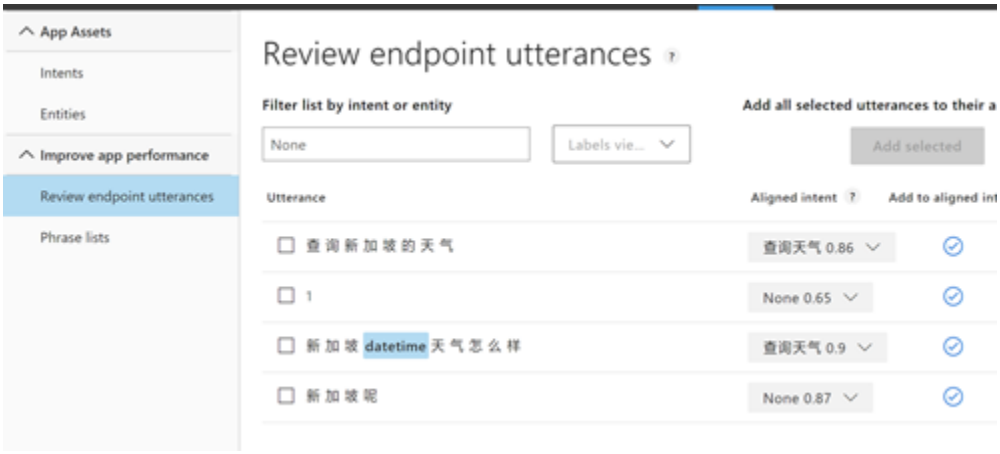
```
        {
        replyString = await GetWeather(location);
        await context.PostAsync(replyString);
        context.Wait(MessageReceived);
        }
        else
        {
        await context.PostAsync("亲你要查询哪个地方的天气信息呢，快把城市的名字发给我吧");
        context.Wait(AfterEnterLocation);
        }
        }
        }
```

GetWeather 这个函数需要自己实现，调用相关的 API 返回相应的天气信息。然后在 `Task<HttpResponseMessage> Post([FromBody]Activity activity)` 下创建 MengmengDialog 来处理对话的内容。

```
        public async Task<HttpResponseMessage> Post([FromBody]Activity activity)
        {
        if (activity.Type == ActivityTypes.Message)
        {
        await Conversation.SendAsync(activity, () => new MengmengDialog());
        }
        else
        {
        HandleSystemMessage(activity);
        }
        var response = Request.CreateResponse(HttpStatusCode.OK);
        return response;
        }
```

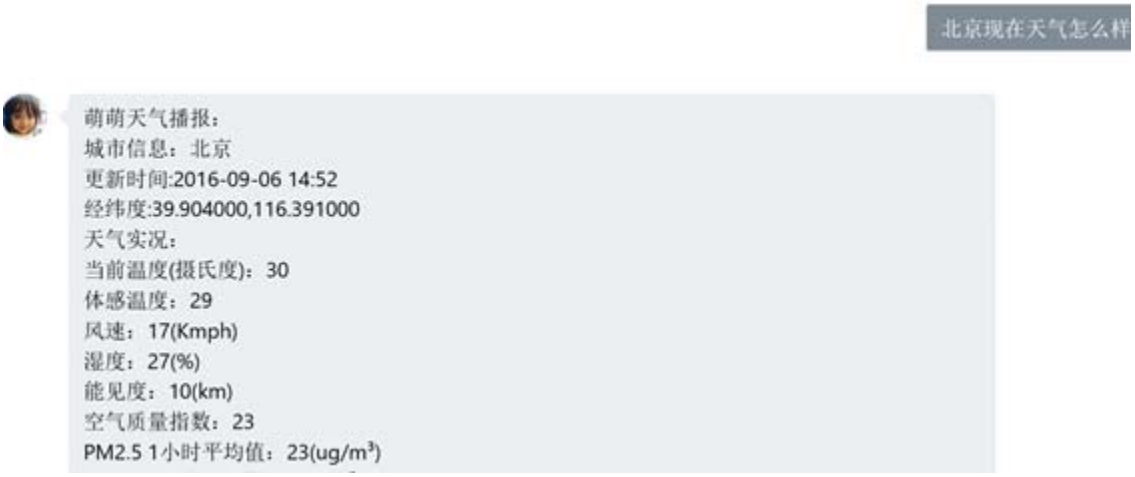

然后发布，这样我们的天气查询机器人就完成啦！测试一下：

换着问法问天气，发现都没问题，都能够正确地识别出来。在 LUIS Portal 里头，点击 `Review endpoint utterances`，可以看到所有调用 LUIS API 发过来的消息，也就是说，你可以在这里看到所有用户发来的消息。



你可以看看 LUIS 标注得对不对，如果错了勾选，然后选择“重新分配意图”，然后保存，之后 LUIS 会学习到这个知识，变得越来越准。重新标注之后记得重新 train 一下然后 publish 哈。比如说，输入“北京现在天气怎么样”，发现只识别出来 Intent，但是没有识别出来地点。

重新标注，train，重新 publish 一下。



发现已经能够正确识别。

试试吧，赶紧自己做个 Bot：）

代码已经发布到 GitHub，请参考：https://github.com/cheneyszp/MengmengBot

本次 Chat 文字分享到此结束，如果还有其它问题或建议，欢迎在读者圈留言，期待与各位线上交流！也欢迎各位加入微软专家兴趣群！