

## LuisDialog.cs

```
using RestaurantBot.Models;
using Microsoft.Bot.Builder.FormFlow;
using Microsoft.Bot.Builder.Dialogs;
using Microsoft.Bot.Builder.Luis;
using Microsoft.Bot.Builder.Luis.Models;
using System.Threading.Tasks;

[LuisModel("appid", "subscriptionkey")]
[Serializable]

public class LUISDialog : LuisDialog<UserOrder>
{
    private readonly BuildFormDelegate<UserOrder> OrderPizza;

    public LUISDialog(BuildFormDelegate<UserOrder> pizzaorder)
    {
        this.OrderPizza = pizzaorder;
    }

    [LuisIntent("")]
    public async Task None(IDialogContext context, LuisResult result)
    {
        await context.PostAsync("I'm sorry I don't know what you mean.");
        context.Wait(MessageReceived);
    }

    [LuisIntent("Greetings")]
    public async Task Greeting(IDialogContext context, LuisResult result)
    {
        context.Call(new RootDialog(), Callback);
    }

    private async Task Callback(IDialogContext context, IAwaitable<object> result)
    {
        context.Wait(MessageReceived);
    }

    [LuisIntent("Order")]
    public async Task PizzaOrder(IDialogContext context, LuisResult result)
    {
        var enrollmentForm = new FormDialog<UserOrder>(new UserOrder(),
this.OrderPizza, FormOptions.PromptInStart);
        context.Call<UserOrder>(enrollmentForm, Callback);
    }

    [LuisIntent("AvailableAmenities")]
    public async Task QueryAmenities(IDialogContext context, LuisResult result)
    {
        foreach (var entity in result.Entities.Where(Entity => Entity.Type ==
"Amenities"))
        {
            var value = entity.Entity.ToLower();
            if (value == "banquet" || value == "pool" || value == "garden" ||
value == "kidszone")
            {
                await context.PostAsync("Yes we have that!");
                context.Wait(MessageReceived);
                return;
            }
        }
    }
}
```

```
        else
        {
            await context.PostAsync("I'm sorry we don't have that.");
            context.Wait(MessageReceived);
            return;
        }
    }
    await context.PostAsync("I'm sorry we don't have that.");
    context.Wait(MessageReceived);
    return;
}
}
```