HOME SUBSCRIBE

# How to process Facebook Messenger callbacks using the Microsoft Bot Framework

18 APRIL 2017 on Microsoft, Microsoft Bot Framework, BotBuilder, NodeJS, Facebook, Facebook-Extension

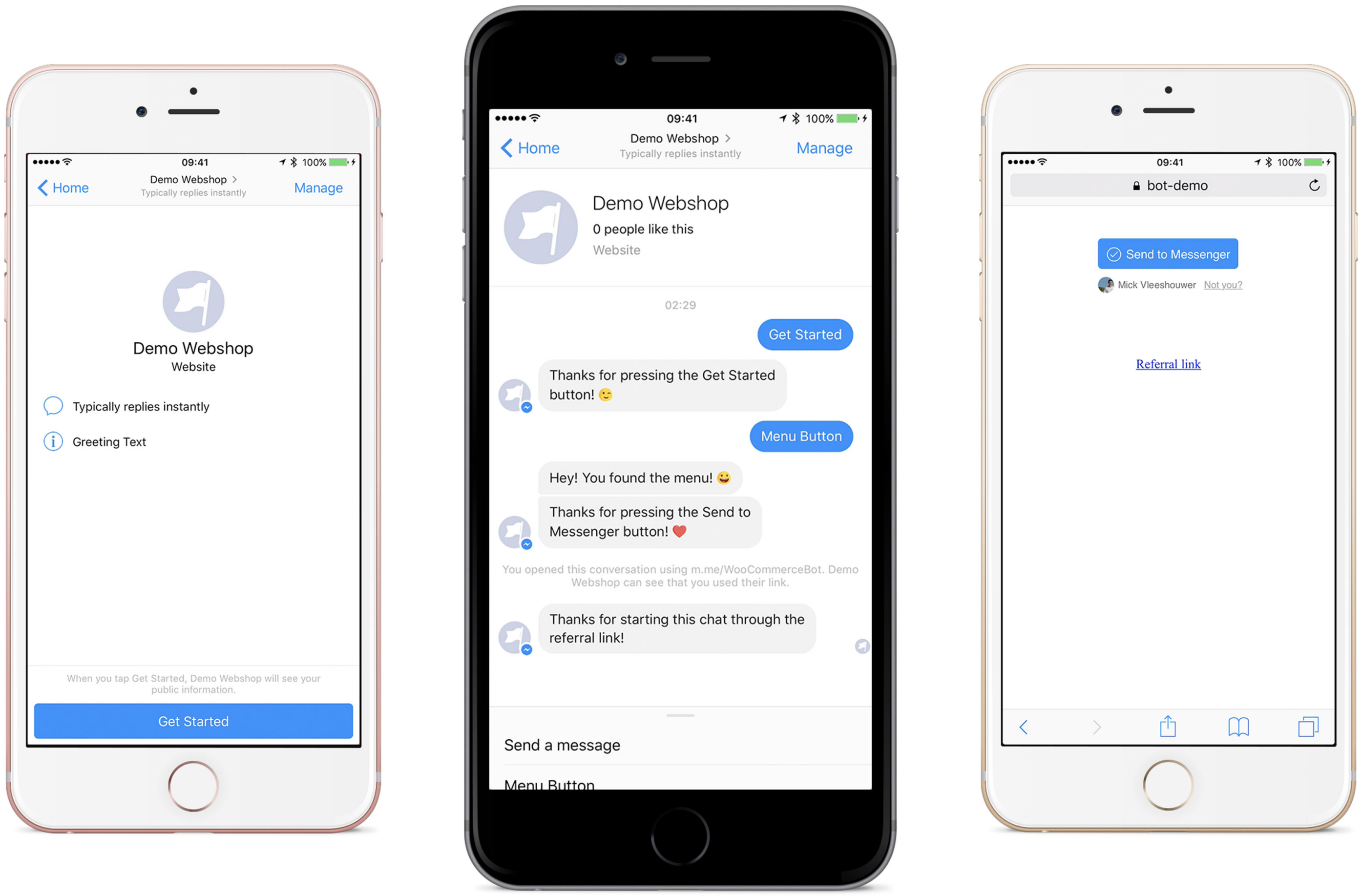
EMAIL TWITTER FACEBOOK LINKEDIN REDDIT

> *The Microsoft Bot Builder SDK is one of three main components of the Microsoft Bot Framework. The Microsoft Bot Framework provides just what you need to build and connect intelligent bots that interact naturally wherever your users are talking, from text/SMS to Skype, Slack, Office 365 mail and other popular services.*

*This is a post in a series about Facebook Messenger & Microsoft Bot Framework, view all posts in this series here.*

The Get Started button, Persistent Menu, Send to Messenger plugin and the referral link are ways to guide the user to a specific dialog without having to initiate a chat first. The BotBuilder SDK can't handle those callbacks by default, so I have written a recognizer class to handle this. With the CallbackRecognizer you are able to map an incoming event to an intent.

Sounds fancy, right? Let's get more technical. Facebook uses referrals, optins and postbacks to send a custom payload to your chatbot. We can use this payload to trigger a specific dialog or just to process the data, e.g. authenticating a user.



When you tap Get Started, Demo Webshop will see your public information.

Thanks for pressing the Get Started button! 😉

Hey! You found the menu! 😀

Thanks for pressing the Send to Messenger button! ❤️

You opened this conversation using m.me/WooCommerceBot. Demo Webshop can see that you used their link.

Thanks for starting this chat through the referral link!

This provides countless possibilities for creating a better user interaction. One of the possible use cases is to register the user for a 'back in stock message' by pressing the Send to Messenger button on the website, but there are many more use cases.

## How to use the CallbackRecognizer?

1. If you don't have the BotBuilder-FacebookExtension package yet, get it via npm. `npm install botbuilder-facebookextension --save`

2. Load the BotBuilder-FacebookExtension module and register the CallbackRecognizer using the following code snippet.

```
const facebook = require('botbuilder-facebookextension');

bot.recognizer(new facebook.CallbackRecognizer());
```

3. In our example we are going to use callbacks to trigger a specific coupon dialog. Later on I will show you how to process the callback payload, instead of opening a specific dialog.

```
bot.dialog('/coupon', [
    (session, args, next) => {
        session.endDialog(`Happy to see that you like our newest
    }
]).triggerAction({
    matches: 'coupon' // callback payload
});
```

4. Check if the callbacks are enabled for your application in the Facebook Developers Console. Go to your application and choose webhook > fields. Make sure `messaging_postbacks`, `messaging_optins` and `messaging_referrals` are enabled!

We are all set for now. Let's have a look at all the possible callback options Facebook provides.

## Referral Callback

Referral callbacks are the easiest to setup. You can use the m.me link with a referral parameter, a parametric Messenger Code or a Messenger Conversation ad. Have a look that documentation that Facebook provides for the implementation instructions. The value of the `ref` parameter will be used as an intent, for example http://m.me/mybot?ref=coupon will fire the `/coupon` dialog in the example above.

## Get Started & Persistent Menu

The Get Started & Persistent Menu functions can bring a user directly into the top-level features and flows of your bot. You can't enable those features directly from your chatbot code (yet) and you only have to configure it once.

Paste the following curl request into your terminal and add your Page Access Token. The value of the `payload` parameter will be used as intent to trigger a dialog. Have a look in the Messenger Platform docs for all the options the Persistent Menu offers.

```
curl -X POST -H "Content-Type: application/json" -d '{
  "get_started":{
    "payload":"GET_STARTED"
  }
}' "https://graph.facebook.com/v2.6/me/messenger_profile?access_

curl -X POST -H "Content-Type: application/json" -d '{
  "persistent_menu":[
    {
      "locale":"default",
      "call_to_actions":[
        {
          "type":"postback",
          "title":"Get Coupon",
          "payload":"coupon"
        }
      ]
    }
  ]
}' "https://graph.facebook.com/v2.6/me/messenger_profile?access_
```

## Send to Messenger Plugin

My favourite callback option is the Send to Messenger plugin which you can integrate in your website. When you book a flight with KLM you can link your Facebook profile in the order process and you will receive your boarding pass on Facebook Messenger.

Integrating the Send to Messenger plugin is easy and requires you to implement a Javascript and HTML snippet on your website. Read more about the implementation in the Messenger Platform Docs. The value `data-ref` will be used as intent.

```
<div class="fb-send-to-messenger"
  messenger_app_id="APP_ID"
  page_id="PAGE_ID"
  data-ref="coupon"
  color="blue"
  size="standard">
</div>
```

*Beware that the Checkbox Plugin isn't fully supported out of the box (yet) since you receive a `user_ref` and not a user id.*

---

## Advanced options

### Use callbacks for data instead of intents

The CallbackRecognizer is easy to use for triggering a specific dialog, but it can also be used to retrieve and parse data. For example, when you want to use the callback payload to link a user from a website to a Facebook profile. In order to implement this we have to pass a settings object to the recognizer. This makes sure we don't use the payload as intent anymore by changing it into a general type intent like `optin`.

Make sure you **never** send an unencrypted value for account linking, since tech savvy users can change the payload they send! Use it wisely.

```
bot.recognizer(
    new facebook.CallbackRecognizer({
        referralValue: false, // Optional - Stop using referral
        postbackValue: false, // Optional - Stop using postback
        optinValue: false     // Optional - Stop using optin pay
    })
);
```

```
bot.dialog('/link-user', [
    (session, args, next) => {
        if (args.intent !== undefined && args.intent.entities !=
            const code = builder.EntityRecognizer.findEntity(arg

            // Do something with the payload (ref)
            console.log(entity.ref);
        }
    }
]).triggerAction({
    matches: 'optin' // referral, optin or postback
});
```

## Retrieving entities

Besides the intent(ion) that is passed through the dialog, you can also extract the full callback payload. The BotBuilder SDK includes an EntityRecognizer class to simplify working with these entities. Possible entity types are `optin`, `referral` and `postback`.

```
if (args.intent !== undefined && args.intent.entities !== undefi
    const code = builder.EntityRecognizer.findEntity(args.intent
    console.log(entity.facebook); // Original Facebook response
}
```

## Disable specific callback types

When you don't want to handle callbacks of a specific type, you can pass the following settings to the CallbackRecognizer.

```
bot.recognizer(
    new facebook.CallbackRecognizer({
        referral: false, // Optional - Disables the referral rec
```

```
            postback: false, // Optional - Disables the postback rec
            optin: false,    // Optional - Disables the optin recogn
        })
    );
```

## Can I send JSON?

Yes! But make sure you send it as a string. `JSON.stringify()`. When your payload is a JSON string, make sure you disable the referralValue, postbackValue or optinValue!

---

**Any questions?** Leave them in the comments or tweet me at @imicknl. In the coming weeks I am going to extend this package with more native Messenger Platform functions. Let me know what you would like to see integrated into this BotBuilder FacebookExtension!

View project on Github: BotBuilder-FacebookExtension

EMAIL TWITTER FACEBOOK LINKEDIN REDDIT

**Mick Vleeshouwer**

Partner Technology Strategist @ Microsoft | Software Engineer

*Amsterdam*

READ THIS NEXT

## Reverse engineering the PostNL consumer API

While working on automating my home using HomeAssistant (tip!), I thought it would be awesome to show incoming deliveries...

YOU MIGHT ENJOY

## How to retrieve User Data from Facebook using the Microsoft Bot Framework

The Microsoft Bot Builder SDK is one of three main components of the Microsoft Bot Framework. The Microsoft Bot...