

Setting up your bot

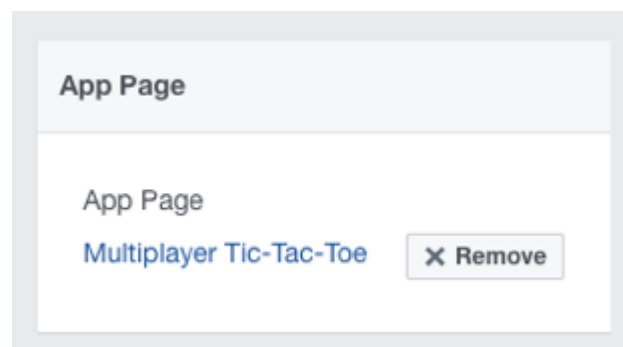
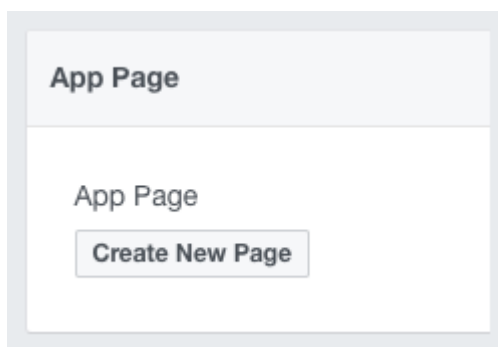
An interesting feature of Instant Games is the ability to have a Messenger Platform Bot attached to it. Although this is optional, it gives your game a powerful channel for re-engagement. The guide below shows you how to create and set up your Game Bot.

Step 1: Create a Page

In order to create a Game bot, you'll first need to create a Facebook page. In order for the page to work correctly with your Instant Game it needs some special properties:

1. The page's category needs to be **App Page**
2. The page's name needs to **contain the name** of the app.
3. The page **cannot be associated** with another app.

You can create a page with these special conditions by going to the **App Page** section of the **Instant Games** product in your App dashboard. Before moving on to the next step, make sure that your **App Page** section looks like the step on the right:



Note: If your Instant Game is not correctly associated with a page as explained above, your bot will not be able to receive `messaging_game_plays` events

Step 2: Activate your Bot

After creating your page, you'll need to make sure to respond to its messaging webhooks. [Webhooks](#) are HTTP calls that we send to your backend when a messaging event is sent to your page. Your server's logic will then decide how to properly respond to each event, if a response is appropriate. To associate your server's endpoints with your page events, follow the instructions on the [Messenger Platform Quickstart Tutorial](#) to enable the bot for your page. The table below contains information about the webhooks and permissions you will need to make your bot work with Instant Games:

Section	Values
Page events	<code>messages</code> and <code>messaging_game_plays</code>
Permissions	<code>pages_messaging</code>

Instant Games bots are only permitted to use [standard messaging](#) and the `GAME_EVENT_MESSAGE_TAG` but not `PAGES_MESSAGING_SUBSCRIPTIONS`.

If your bot has other functionality that requires [subscription messaging](#) or [customer matching](#) you should create a separate app and apply for Messenger platform permissions again.

Step 3: Respond to `messaging_game_plays` webhooks

Once your bot is correctly configured, your server application will start receiving `messaging_game_plays` webhooks every time a player closes the Instant Game. Below is an example of a server application detecting and responding to one of these webhooks.

```
if (event.game_play) {
  var senderId = event.sender.id; // Messenger sender id
  var playerId = event.game_play.player_id; // Instant Games player id
  var contextId = event.game_play.context_id;
  var payload = event.game_play.payload;
  var playerWon = payload['playerWon'];
  if (playerWon) {
    sendMessage(
      senderId,
      contextId,
      'Congratulations on your victory!',
      'Play Again'
    );
  } else {
    sendMessage(
```

```

        senderId,
        contextId,
        'Better luck next time!',
        'Rematch!'
    );
}
}

```

You can refer to the Messenger Platform documentation for more information on this webhook: [Game Play Webhook Documentation](#).

Step 4: Bringing your players back into the game

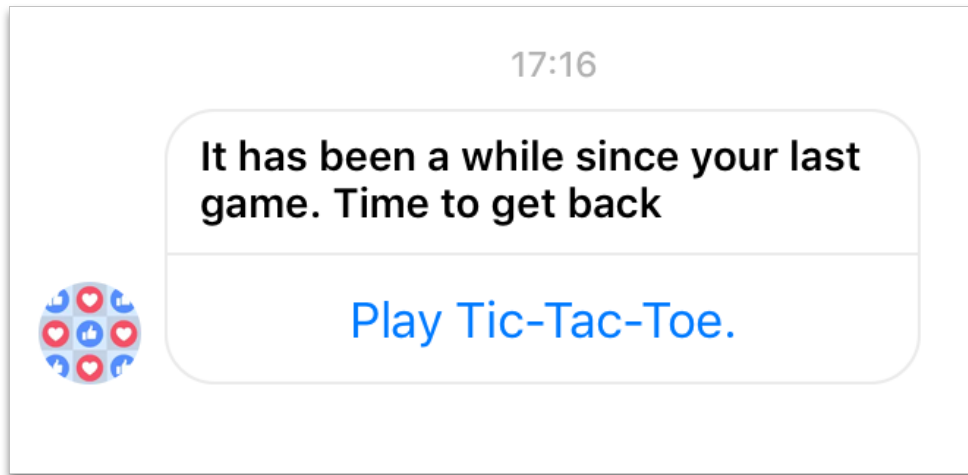
Below is an example of how to use the Graph API to send a game_play button to your players.

```

curl "https://graph.facebook.com/v2.6/me/messages?access_token=<PAGE_ACCESS_TOKEN>"
-X POST
-H "Content-Type: application/json"
-d '{
  "messaging_type": "UPDATE",
  "recipient": {
    "id": "<RECIPIENT_ID>"
  },
  "message": {
    "attachment": {
      "type": "template",
      "payload": {
        "template_type": "generic",
        "elements": [
          {
            "title": "It has been a while since your last game. Time to get back",
            "buttons": [
              {
                "type": "game_play",
                "title": "Play Tic-Tac-Toe.",
                "payload": "{}",
                "game_metadata": {
                  "context_id": "<CONTEXT_ID>"
                }
              }
            ]
          }
        ]
      }
    }
  }
}

```

```
}  
}'
```



You can refer to the Messenger Platform documentation for more information on this button: [Game Play Button Documentation](#).

Step 5: Follow our guidelines and policies

Before it is launched to production, your game bot should go through [Messenger Platform submission process](#). Make sure to follow the Best Practices below before submitting your bot:

Do:

Provide **relevant**, **timely** and **valuable** updates to the players. For more information, visit our [Best Practices](#) section.

Give the user control (for example, by confirming they would like to be notified, and with what frequency).

Use entry point data on play buttons to load the game in contextually relevant ways.

Name the bot the same as the game.

Make use of social updates like turn reminders, tournaments results, timed rewards and challenges.

Make sure your players have the right incentives open the game via a bot message by using the message payload to reward them in-game with something valuable. A bot message is usually not valuable if it opens your game to the start screen.

Use a persistent menu to provide common actions, such as launching the game.

Set default action to use `game_play` on custom updates, so that the entire image takes you into the game.

Use bots to announce new features or content.

Optimize time of day for message sends per user, being sensitive to timezones.

Follow the general [best practices for Messenger Bots](#).

Do not:

- ✗ Send a message immediately after the player closes the game.
- ✗ Send messages to re-engage the player with no context (e.g.: "Come back to the game now!"). Instead prefer re-engagement messages with rich context (e.g.: "Your scout has come back with more info")
- ✗ Adopt the voice of other Facebook users or mislead players to believe their friends are communicating with them.
- ✗ Continue to send a user bot messages when they repeatedly do not engage. Policy limits will apply and block your message from being sent. **Current limits are 5 messages over 10 days of last game play session.** [More information in section 9.4 of our Platform Policy Docs](#)
- ✗ Link to any app store.

References

Please refer to the documents below for more information on how to build and optimize your Game bot.

- ["Game Play" webhook documentation](#)
- ["Game Play" button documentation](#)
- [Best practices with game bots](#)

Next steps

Now that you know how to build an Instant Game with an associated Game bot, it's time to test it, and prepare for launching: [Testing, publishing and sharing your Instant Game](#)

Games

Game Services

Mobile Games

Instant Games

[Getting Started](#)

[Quick Start](#)

[Game Setup](#)

[Game Bot Setup](#)

[Test, Publish, Share](#)

[Bundle Config](#)

[Game Launch](#)

[Guides](#)

[FAQ](#)

[SDK](#)

[PC Games SDK](#)

[Games on Facebook](#)

[App Center](#)

[Facebook Gameroom](#)

[Gaming Community](#)

LANGUAGES

[English \(US\)](#)

[Deutsch](#)

[Bahasa Indonesia](#)

[Español](#)

[Français \(France\)](#)

[العربية](#)

[中文\(简体\)](#)

[Português \(Brasil\)](#)

[Italiano](#)

[日本語](#)

Products

[Facebook Login](#)

[Sharing on Facebook](#)

[Games](#)

[Facebook App Ads](#)

SDKs

[iOS SDK](#)

[Android SDK](#)

[JavaScript SDK](#)

[PHP SDK](#)

[Unity SDK](#)

Tools

[Graph API Explorer](#)

[Open Graph Debugger](#)

[Object Browser](#)

[JavaScript Test Console](#)

[API Upgrade Tool](#)

[Facebook Analytics](#)

Support

[Platform Status](#)

[Developers Group](#)

[Marketing Partners](#)

[Bugs](#)

News

[Blog](#)

[Success Stories](#)

[Videos](#)

[About](#) [Create Ad](#) [Careers](#) [Platform Policy](#) [Privacy Policy](#) [Cookies](#) [Terms](#)

Facebook © 2018