



Workplace Developers Workshop

Updated: March 31, 2022

Table of Contents

Table of Contents	2
Read Me First	4
Setup	4
Creating a custom Integration	4
Postman Setup	8
LAB 1 - Chat Bots: Thread Management	11
Creating new named threads	11
Sending follow-up messages	13
Renaming a thread	14
Add a user to a thread	15
Remove a user from a thread	16
LAB 2 - Chat Bots: Sending Attachments	18
Message with image attachment	18
Message with audio attachment	19
Message with video attachment	20
Message with file attachment	21
LAB - 3 Chat Bots: Quick Replies	22
Text quick reply	22
LAB 4 - Chat Bots: Templates	24
Generic	24
LAB 5 - Group Bots: Post Management	25
Preparation	25
Simple post	26
Post with link and markdown	27
Get comments	30
Post comments	31
Post Likes	32
Delete Post	33
LAB - 6 Group Bots: Post with images and videos	35
Post with an image	35
Post with multi images	36
Instruction 1: Upload 1st photo unpublished	37
Instruction 2: Upload 2st photo unpublished	38

	Developer Workshop Workplace from Meta
Instruction 3: Post with multi images	39
Post with a video	40
Instruction 1: Upload a video	41
Instruction 2: Post with a video	42
LAB - 7 Use Webhooks	43
Edit a sample PHP code	43
Heroku deployment	44
Webhooks Setup	47
Run Chatbot conversation	48
Appendix	50
Workplace Developer Document	50
Third Party Apps	50
Sample Apps for Workplace	51

Read Me First

This training includes the following materials. Please review them first.

- Step-by-step guide (this document)
- Sample programs for Postman Workshop (Workplace Bots Collection v 1.1.json)
- Sample video for Postman Workshop (DeveloperWorkshop.mov)
- Program folder for Webhook (my-project folder)
 - Program file (index.php)
 - Image files (7 image files under imgs folder)

Setup

Creating a custom Integration

1. Go to the Admin Panel
2. Click Integrations, then click Create Custom Integration:

The screenshot shows a modal dialog titled "Create custom integration". It has a "Name" field (A) containing "Developer Workshop", a "Description" field (B) containing "Custom Integration for Developer Workshop", and a note at the bottom stating "Use of the API is subject to the terms of the [Workplace Platform Policy](#)". At the bottom right are "Cancel" and "Create" buttons, with "Create" being highlighted in blue.

- a. Name: Developer Workshop

- b. Description: (Option)
- c. Click Create
3. Click  Edit icon
4. Click Update Logo and upload a photo for your Integration

Integration details
You can update details about this integration

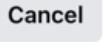
Name, logo & description 

Integration logo 
4

Upload Logo

Integration name
Developer Workshop

Description
Custom Integration for Developer Workshop

5  

5. Click Create Access Token

New token created

To protect your security, only share this token with app developers you trust.

DQVJzTFhNWXFVU19rT0hoY1kxMTU4a3c4R2R4OE5
xUEIGNVRKQzJXWXNreWgwUkpLNUxWOUkwX1Vka
mFjUEJya0FTRGdzYIixNF8wa0pHcDZAVV1FCbWtXc
WRCWEILYktLOExqTzRaZAFZAMSUI3dF9BQk9QNEZ
AxcUJYaHRyYWc3bTgyMINJQVExM21JZAEpvdmZA3
S05BU2w2UDVfZATIIx2tkWG9PcGR3NmZAXRjd6elZ
AHOTFmeIVmbU5IWGJCLUw5Nk9UeHBtMXBB

A Copy

We can only show this token once, so keep it safe. If you lose it, you'll need to create a new one. Anyone can use this token to access information about your Workplace community, depending on the privacy settings of this app.

I understand

B

Use of the API is subject to the [Workplace Platform Policy](#). If you want to share your access token with any third-party developer, please contact us first via [support](#).

C Done

- a. Copy your access token and store ***in a secure place***.
 - b. If you accept the terms, click I understand.
 - c. Click Done.
6. Select the following permissions.

Developer Workshop Workplace from Meta

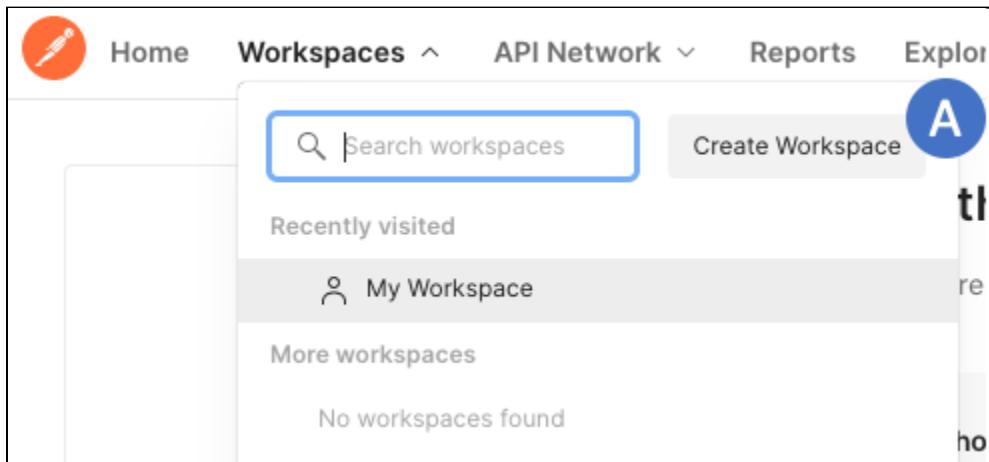
The screenshot shows the 'Integration details' page for 'Developer Workshop'. On the left, a sidebar lists sections: Details, NLP settings, **Permissions** (selected), Webhooks, Security, and Link preview. The main area displays a list of permissions categorized by icons:

- A** Group chat bot (checked), Mention bot (checked)
- B** Delete chat messages, Export insights data, Read group membership, Manage accounts, Manage Frontline access, Manage groups, Manage Knowledge Library content, Manage work profiles, Manage surveys, Provision user accounts, Read and action reported content, Read Knowledge Library content, Read surveys results, Read user email, Read org chart, Read security logs.
- C** Export employee data, Create link previews, Logout, Manage badges, Manage Frontline Areas, Manage important posts, Manage people sets, Manage shift data, Message any member (checked), Read all messages.
- D** Read group content (checked), Read People Sets, Read surveys, Read user timeline, Read work profile, Manage group content (checked).
- E** None

- a. Group chat bot
 - b. Mention bot
 - c. Message any member
 - d. Read group content
 - e. Manage group content
7. Click Save at the end of the web page.

Postman Setup

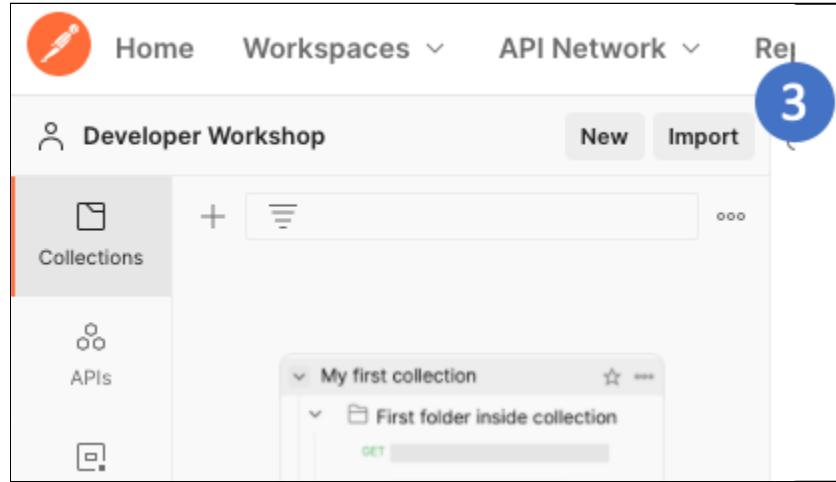
1. Start Postman and access Workspaces.



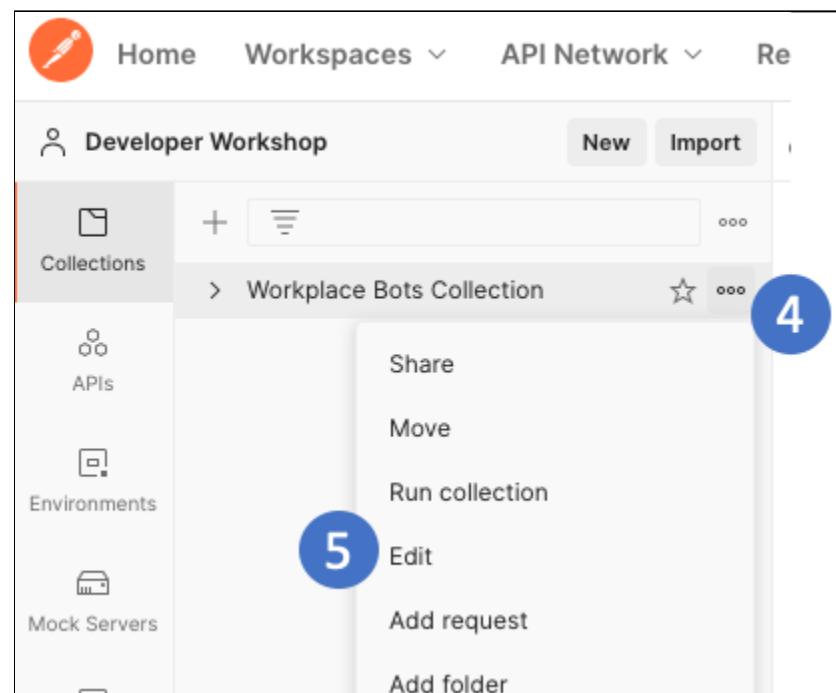
- a. Click Create Workspace
2. Create new Workspace

A screenshot of the 'Create workspace' dialog in Postman. The dialog has a title 'Create workspace'. It contains fields for 'Name' (labeled 'A') and 'Summary' (labeled 'B'). Below these is a 'Visibility' section (labeled 'C') with radio buttons for 'Personal' (selected), 'Private', 'Team', and 'Public'. At the bottom are 'Create Workspace' and 'Cancel' buttons (labeled 'D').

- a. Name: Developer Workshop
 - b. Description: (Option)
 - c. Visibility: Personal
 - d. Click Create Workspace
3. Import “Workplace Bots Collection”.



4. Mouse over this collection, then click “...”, then click Edit.



Developer Workshop Workplace from Meta

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Reports, Explore, Search Postman, Invite, Settings, Notifications, Upgrade.
- Collection Sidebar:** Developer Workshop, Collections, APIs, Environments, Mock Servers, Monitors, Flows.
- Collection Overview:** Workplace Bots Collection (highlighted with a blue circle labeled 5). Sub-folders: Chat Bots, Group Bots.
- Authorization Tab:** Selected (highlighted with a blue circle labeled 6). Description: "This authorization method will be used for every request in this collection. You can override this by specifying one in the request." Type: Bearer Token (highlighted with a blue circle labeled 7). Token input field: DQVJ1bERPZA1M5MUVvR3QyLWV3 ... (highlighted with a blue circle labeled 8).
- Right Panel:** Share, Fork, Environment dropdown, and other icons.

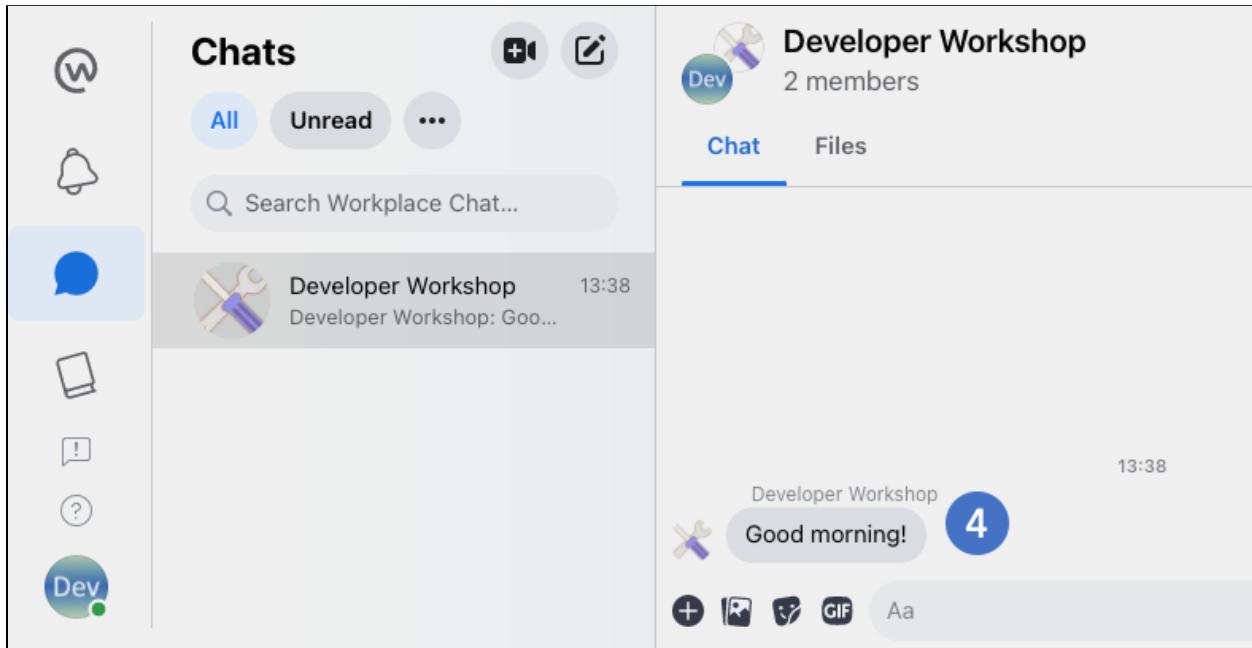
5. Click Authorization.
6. Make sure you have selected Bearer Token.
7. Token: Your Access Token you created in the previous section.
8. Click Save.

LAB 1 - Chat Bots: Thread Management

Expand the “Thread Management” section in Postman.

Creating new named threads

The first step we are going to take is to initiate a new thread with our user. For this and the following labs, please expand “Chat Bots”, then “Thread Management” in the Postman collection.



Instructions:

1. Open Creating new named threads.

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, Monitors, and Flows. Under 'Collections', 'Workplace Bots Collection' is expanded, showing 'Chat Bots' and 'Thread Management'. 'Thread Management' is also expanded, showing several POST requests: 'Creating new named threads', 'Sending follow-up messages', 'Renaming a thread', 'Add a user to a thread', 'Remove a user from a thread', 'Sending attachments', 'Quick replies', and 'Templates'. The main workspace shows a POST request titled 'Creating new named threads' with the URL <https://graph.workplace.com/me/messages>. The 'Body' tab is selected, showing JSON code:

```

1  {
2    "recipient": {
3      "ids": ["{{your_user_id}}"]
4    },
5    "message": {
6      "text": "Good morning!"
7    }
8  }

```

2. Body: Replace {{your_user_id}} with your Workplace user id.
3. Click Send.
4. Go to your Workplace Chat.

The screenshot shows the 'Body' tab of the Postman response. It displays the JSON response from the previous step:

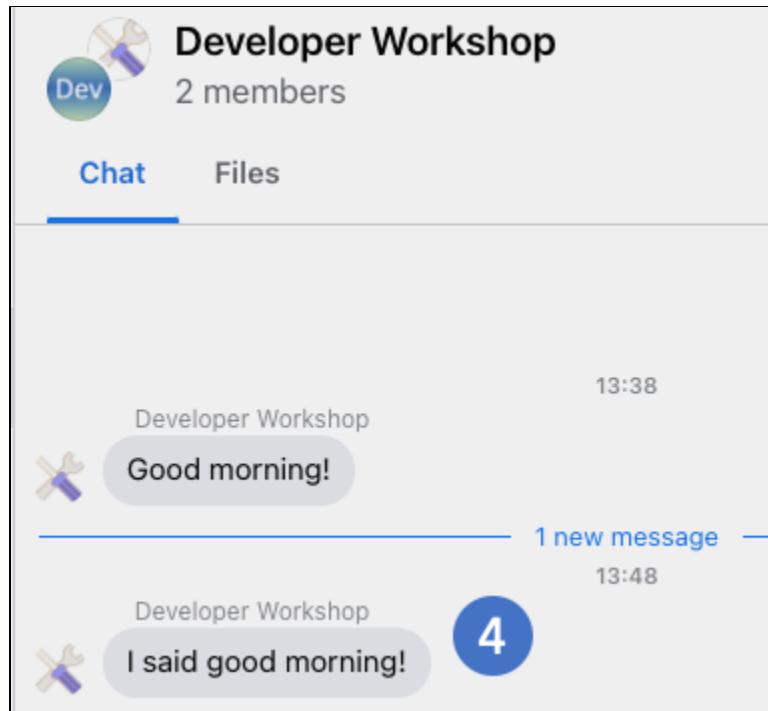
```

1  {
2    "message_id": "m_BvCdoryB1T6dy0bfKzRGt7c5zDqLF6535r40EyCkeuDnNQSz",
3    "thread_key": "t_4746450152149178"
4  }

```

5. Write down the thread key (thread_key) you received as we will need for the following steps.

Sending follow-up messages



Instructions:

1. Open Sending follow-up messages.

The screenshot shows the Postman API client interface. The left sidebar lists collections, APIs, environments, mock servers, monitors, and flows. Under 'Collections', 'Developer Workshop' is selected. In the main workspace, a POST request is shown for the endpoint `https://graph.workplace.com/me/messages`. The 'Body' tab is selected, displaying the following JSON payload:

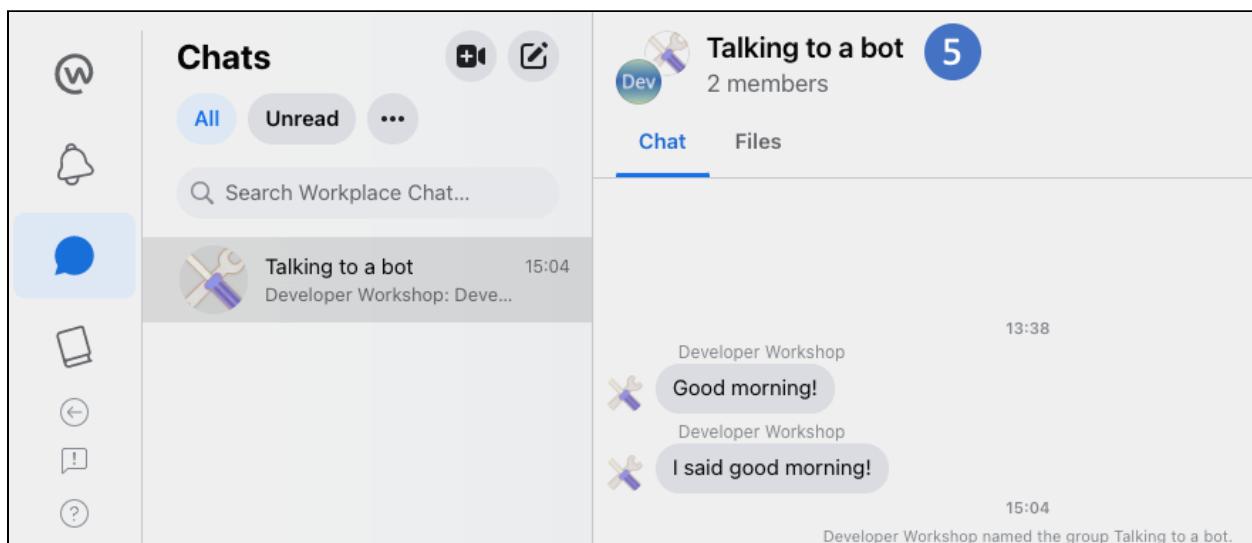
```
1 {  
2   "recipient": {},  
3   "thread_key": "t_4746450152149178"  
4 },  
5   "message": {  
6     "text": "I said good morning!"  
7   }  
8 }
```

A blue circle with the number '2' is visible in the bottom right corner of the code editor.

2. Body: Replace {{thread_key}} with your thread key.
3. Click Send.
4. Check your Workplace Chat.

Rename a thread

We are going to rename the thread to “Talking to a bot”. We specify the thread key in the URL and the new name in the body of the POST request.



Instructions:

1. Open Renaming a thread.

The screenshot shows the Postman application interface. The top navigation bar includes Home, Workspaces, API Network, Reports, and Explore. The search bar says 'Search Postman'. The left sidebar shows collections, environments, and mock servers. Under 'Collections', there is a 'Developer Workshop' collection with a 'Workplace Bots Collection' folder containing 'Chat Bots' and 'Thread Management' subfolders. Under 'Thread Management', there are three items: 'Creating new named threads', 'Sending follow-up messages', and 'Renaming a thread'. The 'Renaming a thread' item is highlighted with a blue circle containing the number '2'. The main workspace shows a POST request for 'Renaming a thread' with the URL https://graph.workplace.com/t_4746450152149178/threadname. The 'Body' tab is selected, showing the following JSON payload:

```
1 "name": "Talking to a bot"
```

2. URL: Replace {{thread_key}} with your thread key.
3. Check the Body.
4. Click Send.
5. Check your Workplace Chat to see the new thread name.

Add a user to a thread

Sometimes we need to add a user to a thread. One user case is to add someone to support a user in case the bot can't help.

The screenshot shows a Workplace chat window titled "Talking to a bot" with 2 members. The "Chat" tab is selected. The conversation log includes messages from the bot at 13:38 and 15:04, and a message from the user at 15:04. The right sidebar shows the group's "About" section with a note about naming and adding people, and a list of members: DEMO USER (TKCORP12), Dev User (TKCORP12), and Developer Workshop.

Instructions:

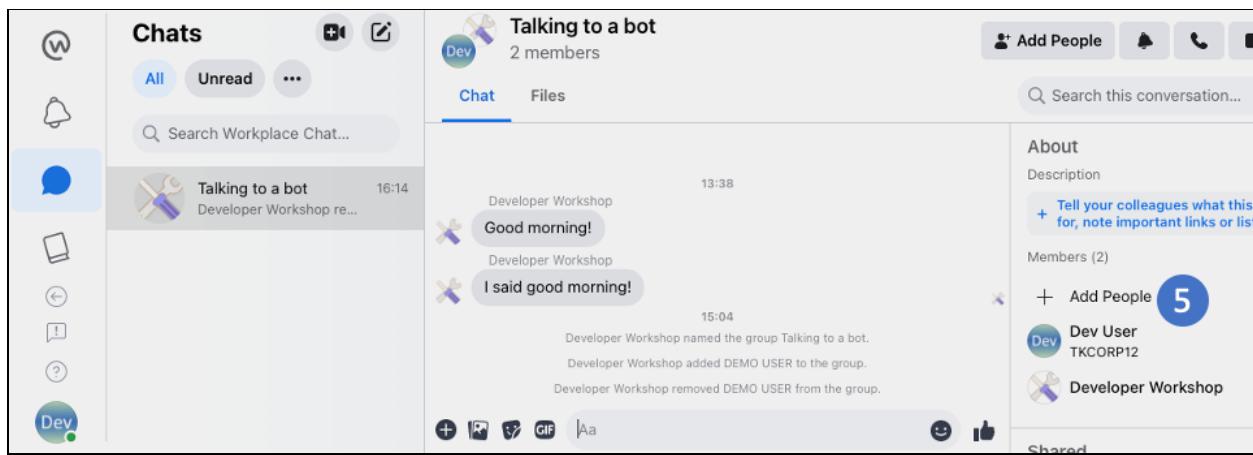
1. Open Add a user to a thread.

The screenshot shows the Postman interface with the "Developer Workshop" workspace selected. The "Thread Management" collection is open, and the "Add a user to a thread" endpoint is selected. The "Body" tab is active, showing a JSON payload with a single key "to": [100075975894963].

2. URL: Replace {{thread_key}} with your thread key.
3. Check the Body and replace {{another_userid}} with the user id you want to add to your thread.
4. Click Send.
5. Check your Workplace Chat. Is there a new member?

Remove a user from a thread

Removing a user is just as easy as adding one.



Instructions:

1. Open Remove a user from a thread.

2. URL: Replace {{thread_key}} with your thread key and check DELETE as a method.
3. Check the Body and replace {{another_userid}} with the user id you want to remove from your thread.

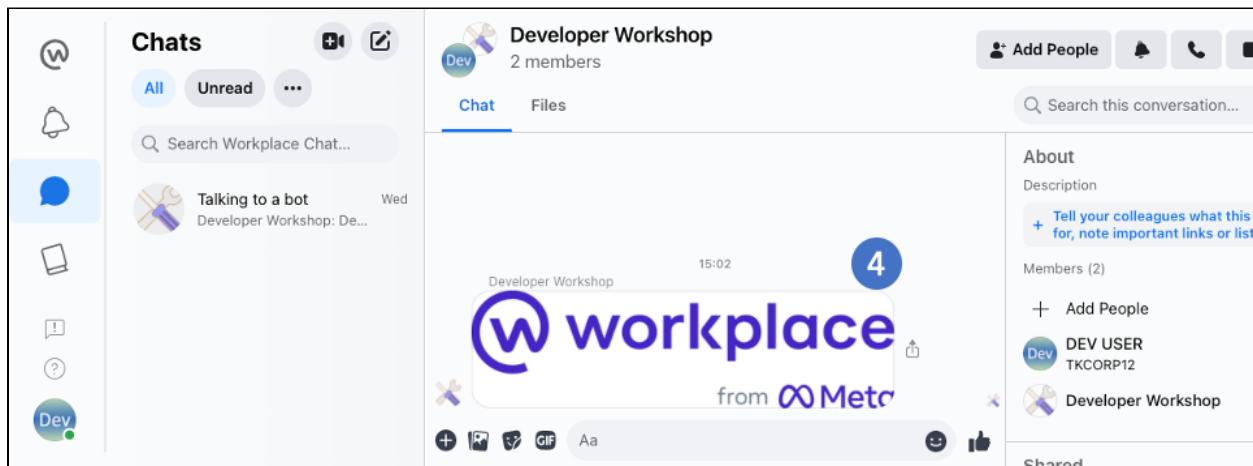
4. Click Send.
5. Check your Workplace Chat. How many members does the thread have?

LAB 2 - Chat Bots: Sending Attachments

Expand the “Sending attachments” section in Postman.

Message with image attachment

Sending an image is one of the most common features used by a bot. This can be used to send receipts, pictures of products, etc. Images can also make your bot look nice and more interactive.



Instructions:

1. Open Message with image attachment.

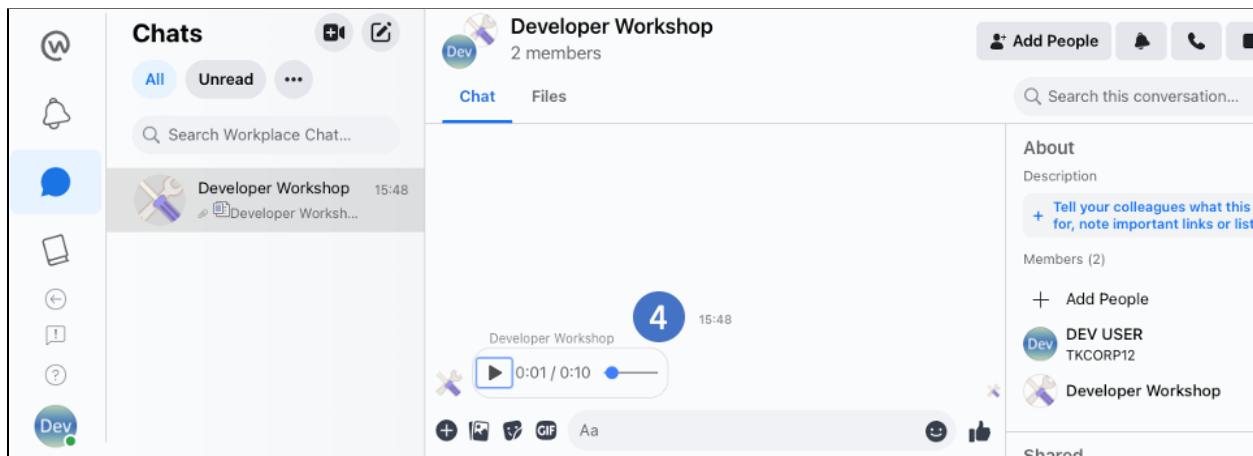
The screenshot shows the Postman interface for the 'Developer Workshop' collection. The left sidebar lists collections, environments, mock servers, monitors, and flows. Under 'Workplace Chat Bots Collection', the 'Sending attachments' section is selected, showing three POST requests: 'Message with image attach...', 'Message with audio attach...', and 'Message with video attach...'. The 'Message with image attach...' request is currently selected. The request details show a POST method to 'https://graph.workplace.com/me/messages'. The 'Body' tab is active, showing a JSON payload with a recipient ID, a message object containing an attachment (image type, URL, and reusable flag), and other fields. A blue circle with the number '2' is over the JSON code, and another blue circle with the letter 'B' is over the URL field. A third blue circle with the letter 'C' is over the 'is_reusable' field in the JSON code.

```
1 {
2   "recipient": {
3     "id": "100078149773806"
4   },
5   "message": {
6     "attachment": {
7       "type": "image",
8       "payload": {
9         "url": "https://wpdeveloperworkshop.herokuapp.com/imgs/Workplace_logo.png",
10        "is_reusable": true
11      }
12    }
13 }
```

2. Body: Replace {{your_userid}} with your user id.
 - a. We are using a URL to reference the image.
 - b. Have you noticed the "is_reusable" inside the payload? When you set it to true, you can reuse the image, receiving an "attachment_id" in the response.
3. Click Send.
4. Check your Workplace Chat. Have you received an image?

Message with audio attachment

You can also send audio attachments. Look that the content is embedded in Workplace Chat. Developing bots for Workplace is so easy since you don't need to think too much about the front-end.



Instructions:

1. Open Message with audio attachments.

Developer Workshop Workplace from Meta

The screenshot shows the Postman interface. On the left, there's a sidebar with collections like 'Workplace Bots Collection' and various API endpoints. The main area shows a POST request to 'https://graph.workplace.com/me/messages...'. The 'Body' tab is selected, displaying a JSON payload. A blue circle labeled '2' highlights the 'ids' field where '100078149773806' is entered. A blue circle labeled 'A' highlights the 'url' field containing 'https://wpdeveloperworkshop.herokuapp.com/audios/Workplace_audio.mp4'. The 'Headers' tab shows '(11)'.

```
1 {
2   "recipient": {
3     "id": "100078149773806"
4   },
5   "message": {
6     "attachment": {
7       "type": "audio",
8       "payload": {
9         "url": "https://wpdeveloperworkshop.herokuapp.com/audios/Workplace_audio.mp4",
10        "is_reusable": true
11      }
12    }
13 }
```

2. Body: Replace {{your_userid}} with your user id.
 - a. We are using a URL to reference the image.
3. Click Send.
4. Check your Workplace Chat. Listen to the audio.

Message with video attachment

Video is also very important today as we don't look for how-tos or manuals anymore, we look for videos to explain what we need to do.

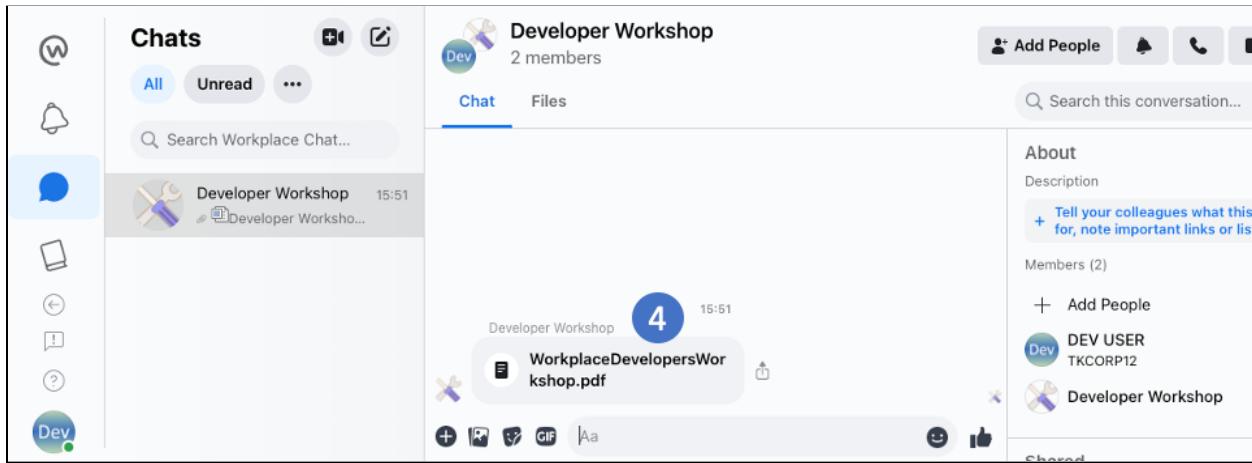
The screenshot shows the Workplace Chat interface. On the left, there's a sidebar with icons for Chats, All, Unread, and ... The main area shows a chat with 'Developer Workshop' (2 members). The 'Chat' tab is selected, showing a message from 'Developer Workshop' at 15:49 with the text 'Workplace Integration with Microsoft Teams'. A blue circle labeled '4' highlights the message text. On the right, there's a sidebar with 'About' (Description: 'Tell your colleagues what this is for, note important links or list'), 'Members (2)' (DEV USER TKCORP12), and 'Add People'.

Instructions:

1. Open Message with video attachment.
2. Body: Replace {{your_userid}} with your user id.
3. Click Send.
4. Check your Workplace Chat. Watch the movie.

Message with file attachment

Sometimes you need to send files to the user. For example, a payslip, receipts, logs, etc. Let's send a file through Workplace Chat.



Instructions:

1. Open Message with file attachment.
2. Body: Replace {{your_userid}} with your user id.
3. Click Send.
4. Check your Workplace Chat. Download the file.

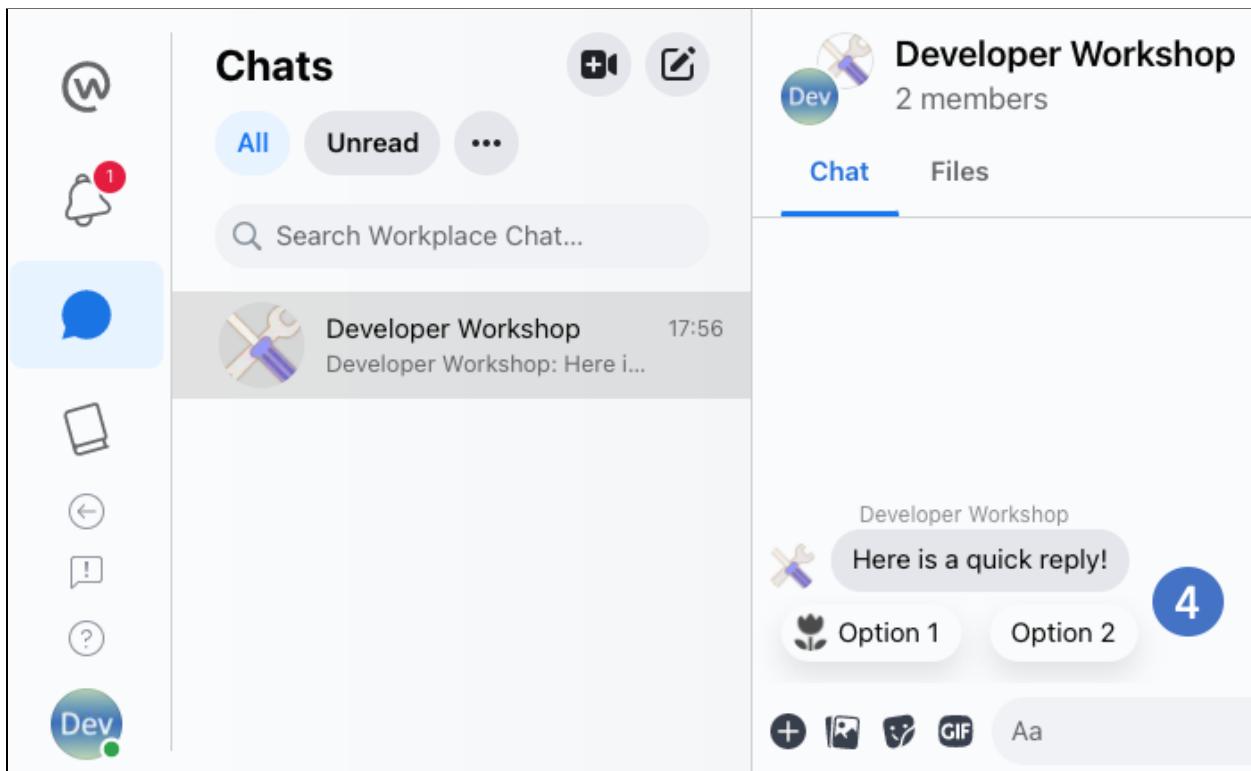
LAB - 3 Chat Bots: Quick Replies

Expand the "Quick replies" section in Postman.

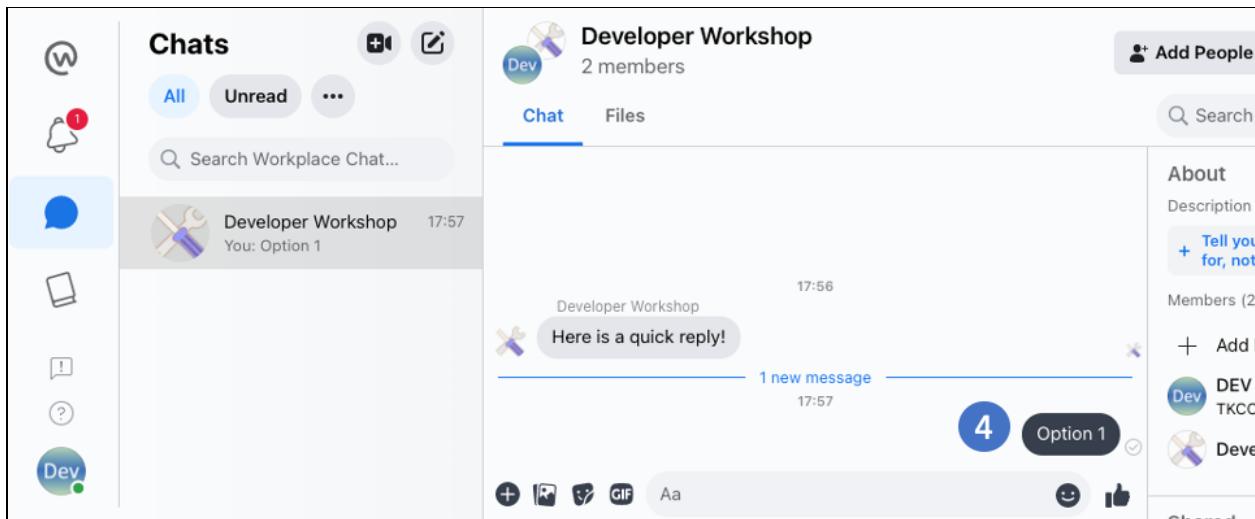
Text quick reply

Some bots rely on Natural Language to understand what the users want. Even though this is a great solution, it is at the same time more complex, and prone to errors. Providing the users some options can easily direct users toward the options available.

One important component is "payload": "<POSTBACK_PAYLOAD>". You can inform Workplace API what to send on webhooks when people select a specific option.



Developer Workshop Workplace from Meta



Instructions:

1. Open Text quick reply.

The screenshot shows the Postman application interface. The left sidebar lists collections, environments, mock servers, monitors, flows, and history. The main area shows a collection named 'Developer Workshop' with a 'Text quick reply' endpoint selected. The 'Body' tab is active, showing a JSON payload for a quick reply message. The payload includes a recipient ID, a message with text 'Here is a quick reply!', and a quick reply object with a title 'Option 1' and a URL 'https://wpdeveloperworkshop.herokuapp.com/imgs/icon04.png'. A blue circle labeled '2' points to the quick reply object in the JSON code.

```
1 {
2   "recipient": {
3     "ids": ["100078149773806"]
4   },
5   "message": {
6     "text": "Here is a quick reply!",
7     "quick_replies": [
8       {
9         "content_type": "text",
10        "title": "Option 1",
11        "payload": "<POSTBACK_PAYLOAD>",
12        "image_url": "https://wpdeveloperworkshop.herokuapp.com/imgs/icon04.png"
13      }
,
```

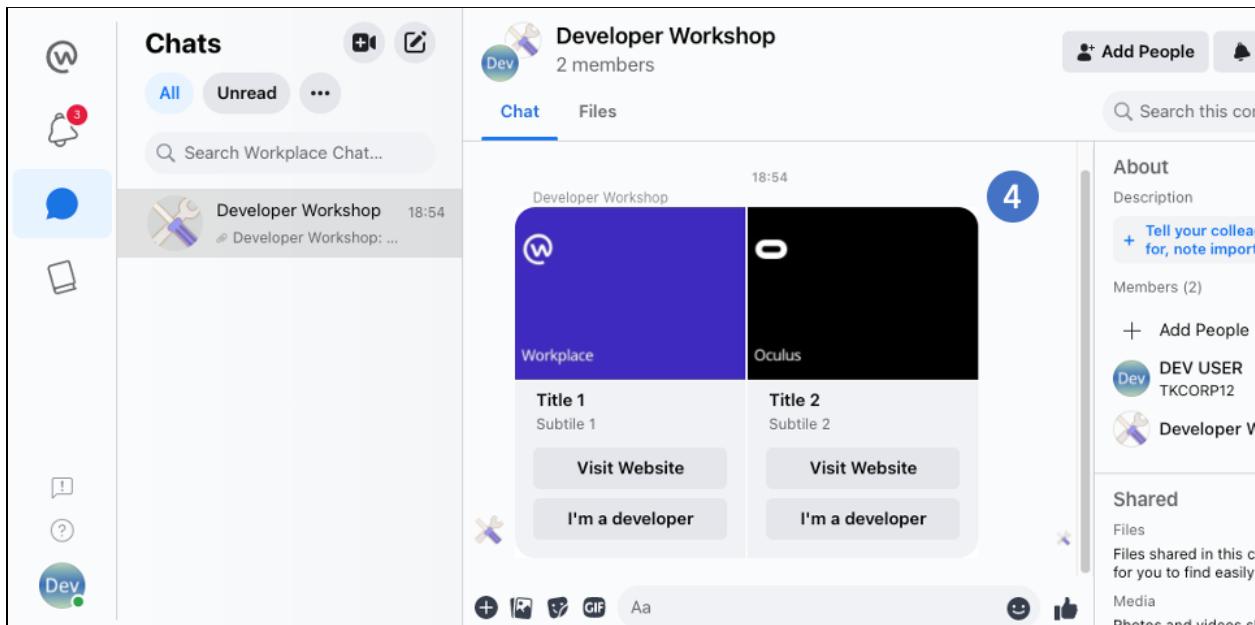
2. Body: Replace {{your_userid}} with your user id.
3. Click Send.
4. Check your Workplace Chat. Is there a difference between button 1 and 2?

LAB 4 - Chat Bots: Templates

Expand the "Templates" section in Postman.

Generic

Use templates as a navigation tool to your bot. Each card can have an image, title, subtitle, and buttons. Each button can be an external URL, or a postback just like we saw in the quick reply lab.



Instructions:

1. Open Generic.
2. Body: Replace {{your_userid}} with your user id.
3. Click Send.
4. Check your Workplace Chat. Explore the options.

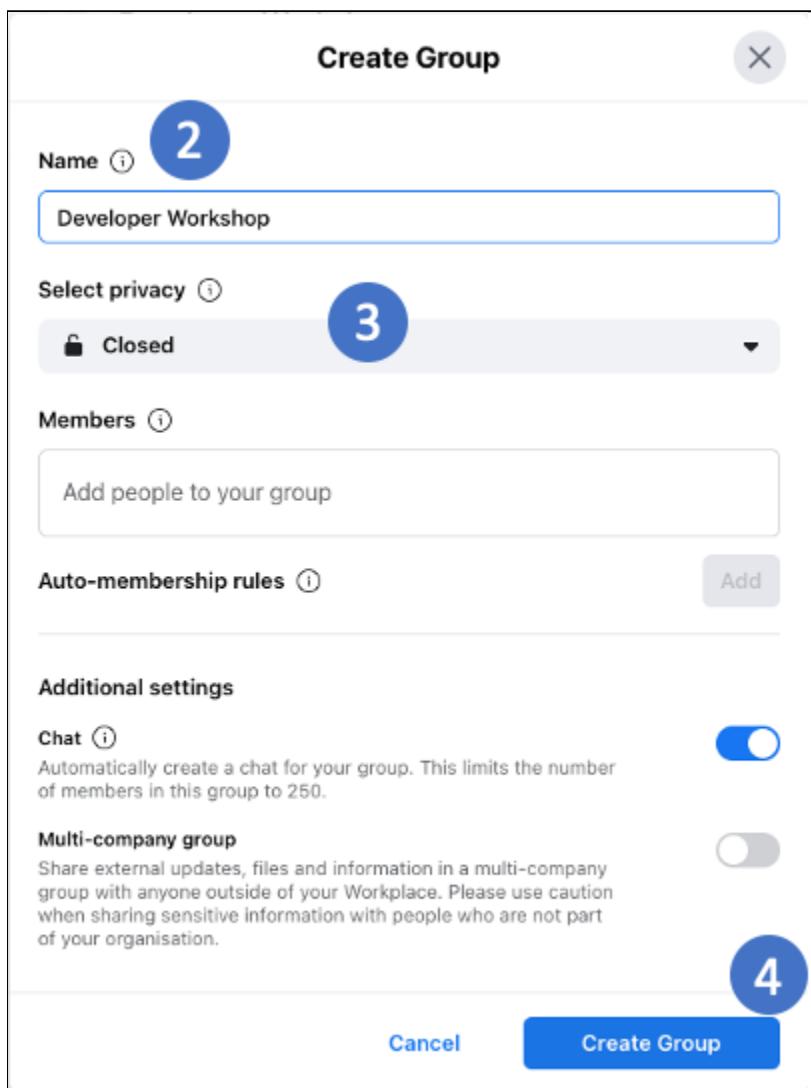
LAB 5 - Group Bots: Post Management

Expand "Post Management" in the Postman collection.

Preparation

In this lab we are going to create a new group. Go to Workplace and follow these steps:

1. In the left column, click "Create Group"



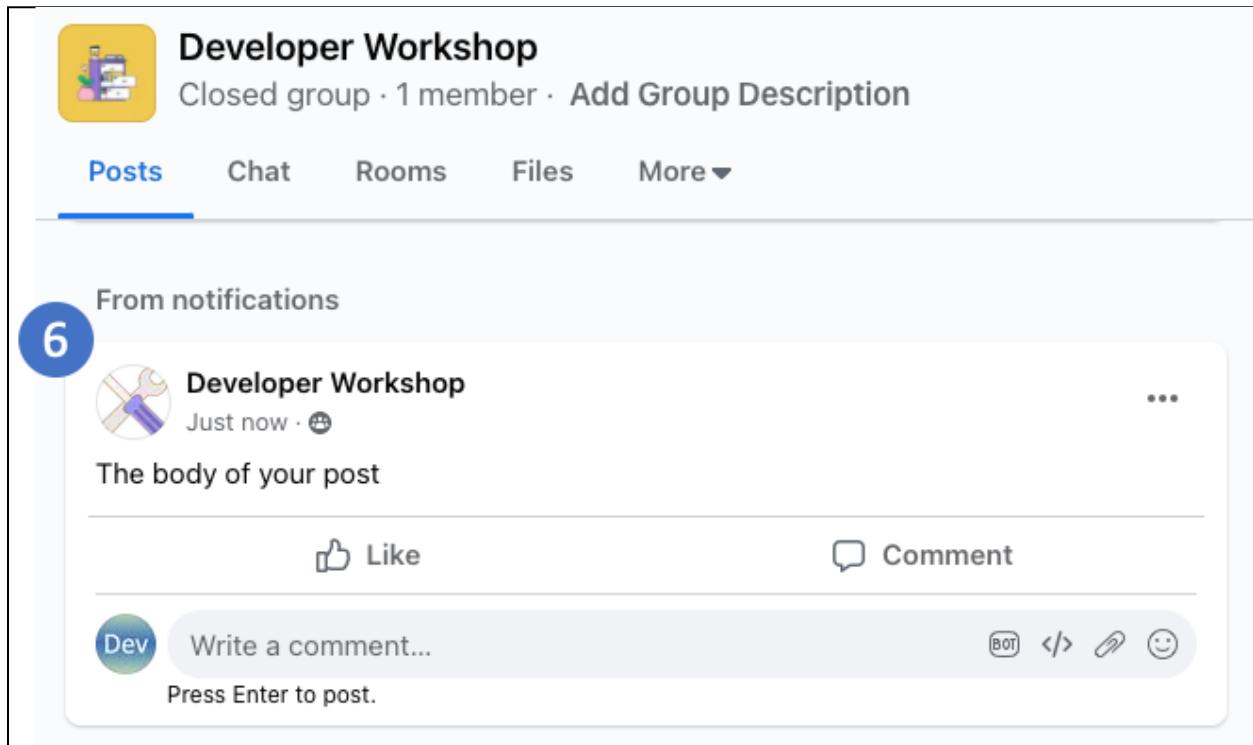
2. Group name: Developer Workshop
3. Select Secret as the privacy option (Anytime is fine.)
4. Click Create Group
5. In the address bar of your browser, you should see something like <https://{{your subdomain}}.workplace.com/groups/XXXXXXXXXXXXXX/>.

The XXXXXXXXXXXX is the ID of the group you just created. Save this ID as we are going to need in this lab.

Tip: Check the Graph API collection to see how you could have created this group using Graph API.

Simple post

The first thing we are going to do is a simple post from our bot in this new group. Bots do group posts to get the attention from all members of the group. Some interesting use cases include notification bots, making a post about the sales quota quarter to date achievement, and recognition bots, when an employee received a thanks, or just completed another year in the company.



Instructions:

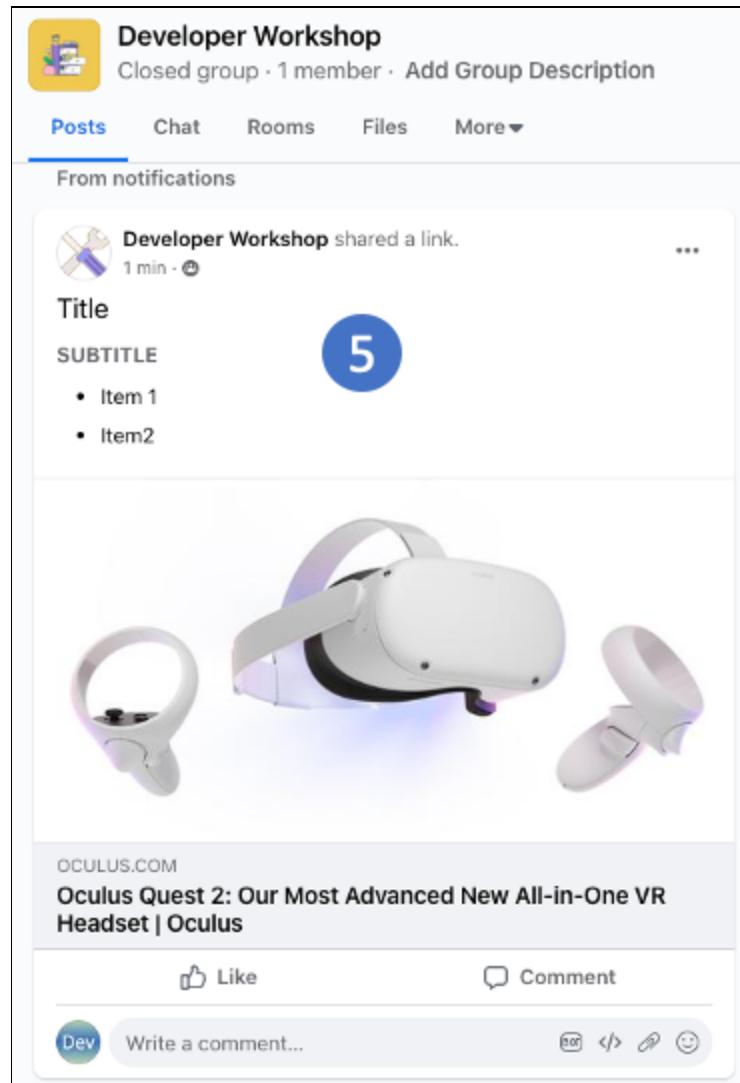
1. Open Simple post.

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Home, Workspaces, API Network, Reports, and Explore. Below these are sections for Collections, APIs, Environments, Mock Servers, and Monitors. A 'Developer Workshop' workspace is selected. In the main area, there's a 'Collections' section with a '+' button and a search bar. To the right, a 'Workplace Bots Collection' is expanded, showing 'Chat Bots' and 'Group Bots' sections. Under 'Group Bots', there are two items: 'POST Simple post' and 'POST Post with images and markdo...'. Above this, a breadcrumb navigation shows 'Workplace Bots Collection / Group Bots / Simple post'. The main panel shows a 'POST' request to 'https://graph.facebook.com/380194830120200/feed'. The 'Body' tab is selected, showing the JSON payload: "message": "The body of your post". There are tabs for Params, Authorization, Headers (10), Body (selected), Pre-request Script, Tests, and Set. A blue circle with the number '2' is overlaid on the top right of the main panel.

2. Replace {{group_id}} in the URL with your group id.
3. Check Body contents.
4. Click Send.
5. If the request is successful, you should get the post id in the HTTP response.
6. Check if the group has a new post made by your bot.

Post with link and markdown

Sometimes we want to make a more professional post, with text formatting, images, links, and so on. Let's see how we can do this.



Instructions:

1. Open Post with images and markdown

Postman interface showing a POST request to `https://graph.facebook.com/380194830120200/feed`. The Body tab displays the following JSON payload:

```

1 "formatting": "MARKDOWN",
2 "message": "# Title\n## Subtitle\n* Item 1\n* Item2",
3 "link": "https://www.oculus.com/quest-2/"
4
5
    
```

2. Replace {{group_id}} in the URL with your group id.
3. Check Body content. We are sending a link and also using markdown to format the message text.
4. Click Send.
5. Check a new well-formatted post.

Postman response body showing a JSON object with an id field:

```

1
2 "id": "380194830120200_380419526764397"
3
    
```

6. Write down the Post ID you get in the HTTP Response. We'll use it in the next request.

Facebook post showing a developer comment:

1 comment Seen by everyone

Like **Comment**

DEV USER
This is a developer comment!

Like Reply 1m

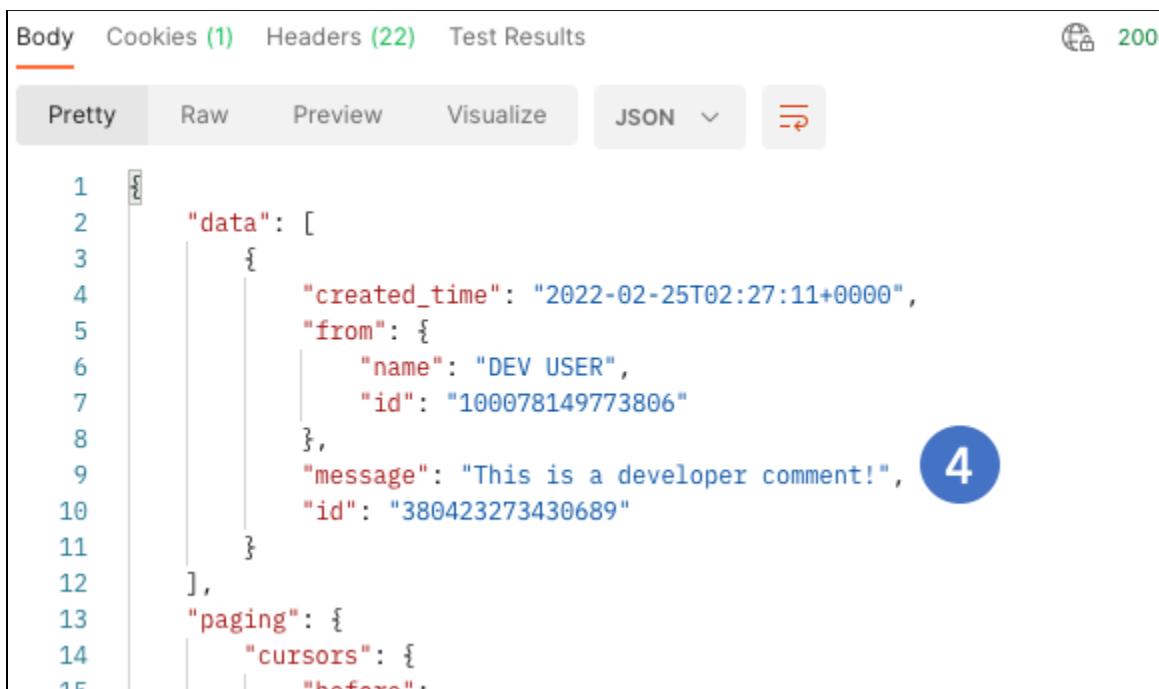
Write a comment... BOT </> 🎁 😊

7. Go to the post and make a comment.

Get comments

The best way to get notified about a comment in a post is through group webhooks, because we can get the comments automatically without querying the API all the time.

But you can use the Graph API to get post comments, especially in cases where the bot is mentioned and you need to look at the post history.



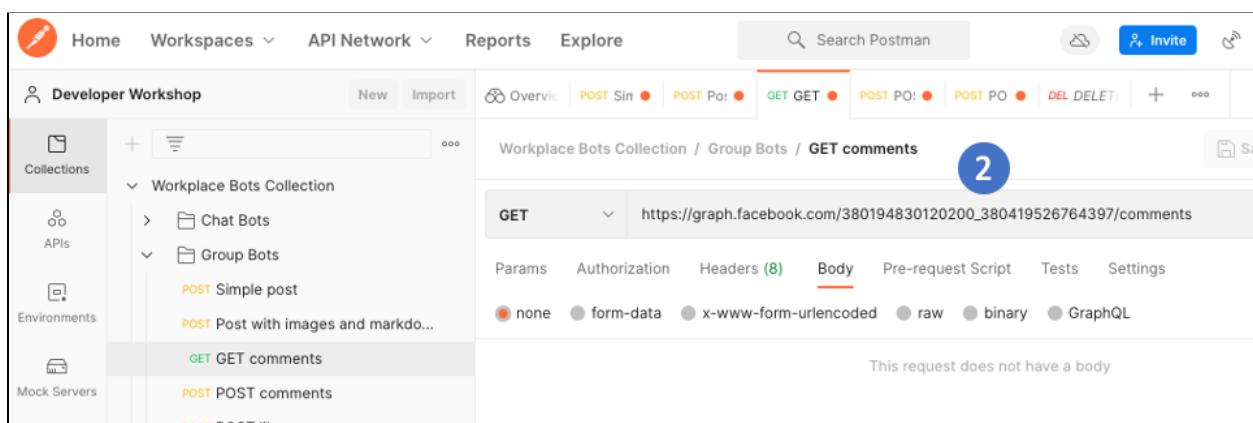
A screenshot of the Postman application interface. At the top, there are tabs for 'Body', 'Cookies (1)', 'Headers (22)', and 'Test Results'. On the right, it shows a status of '200' with a globe icon. Below the tabs, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. The 'JSON' button is highlighted with a dropdown arrow pointing down. The main area displays a JSON response with line numbers on the left:

```
1
2   "data": [
3     {
4       "created_time": "2022-02-25T02:27:11+0000",
5       "from": {
6         "name": "DEV USER",
7         "id": "100078149773806"
8       },
9       "message": "This is a developer comment!",
10      "id": "380423273430689"
11    }
12  ],
13  "paging": {
14    "cursors": {
15      "before": "
```

A blue circle with the number '4' is overlaid on the right side of the JSON response.

Instructions:

1. Open Get comments.

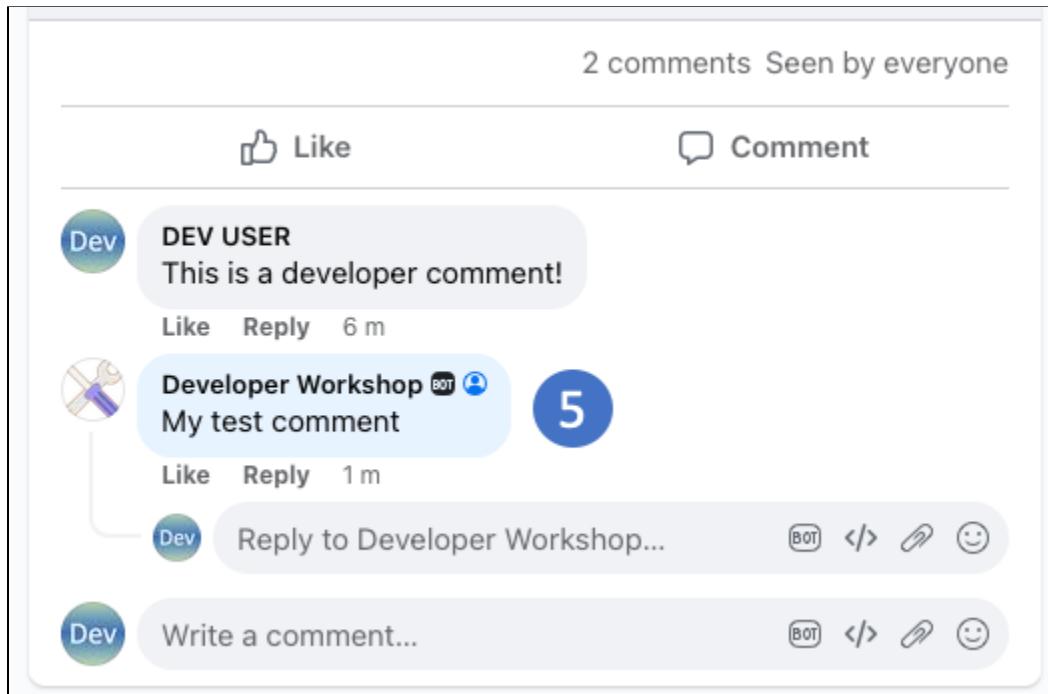


A screenshot of the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', 'Reports', and 'Explore'. A search bar says 'Search Postman' and there's a 'Invite' button. The left sidebar shows 'Developer Workshop' with sections for 'Collections', 'APIs', 'Environments', and 'Mock Servers'. Under 'Collections', 'Workplace Bots Collection' is expanded, showing 'Chat Bots' and 'Group Bots'. 'Group Bots' contains a 'Simple post' and a 'GET comments' request. The 'GET comments' request is selected. The main panel shows the request details:
Method: GET
URL: https://graph.facebook.com/380194830120200_380419526764397/comments
Headers (8):
Body: none
Params, Authorization, Pre-request Script, Tests, Settings are also visible.
A blue circle with the number '2' is overlaid on the right side of the request details.

2. In the URL, replace {{post_id}} with your post id.
3. Click Send.
4. Check if you get all the comments.

Post comments

The bot can post a comment as well. Some bots use this feature to inform the user about a completed action, or the status of the request.



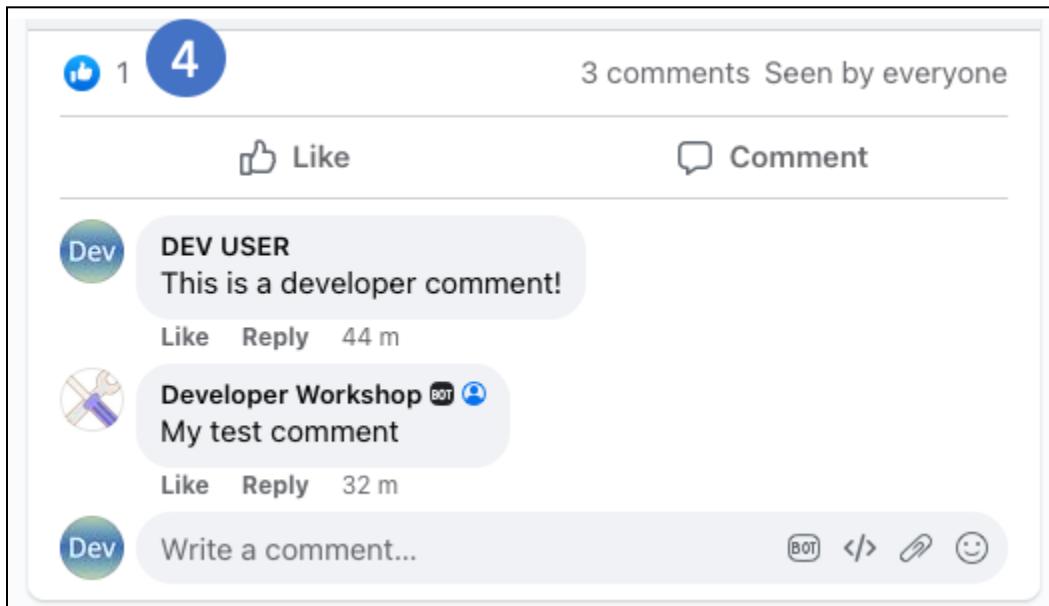
Instructions:

1. Open Post comments.

2. In the URL, replace {{post_id}} with the post id you received in the previous step.
3. Check Body content.
4. Click Send.
5. Is there a new comment on this post?

Post Likes

Your bot can also react to posts. This can make the bot look like a person.



Instructions:

1. Open Post likes.

The screenshot shows the Postman application interface. On the left, there's a sidebar with various sections: Home, Workspaces, API Network, Reports, Explore, and a search bar. Below these are collections, APIs, environments, mock servers, monitors, flows, and history. The main area displays a 'Workplace Bots Collection' under 'Group Bots'. A specific POST request for 'POST likes' is selected. The request details show a POST method to the URL https://graph.facebook.com/380194830120200_380419526764397/likes. The 'Body' tab is active, showing a JSON payload with a single key 'success': true. The response section shows a 200 OK status with a size of 654 bytes.

2. In the URL, replace {{post_id}} with your post id.
3. Click Send.
4. Check if the bot sent a Like to the post.

Delete Post

Deleting a post is not a very common use case for a bot, but you can do it. Some use cases include content organization and use terms enforcement.

Instructions:

1. Open DELETE post.

Developer Workshop Workplace from Meta

The screenshot shows the Postman application interface. The left sidebar is titled 'Developer Workshop' and contains sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. A blue circle with the number '2' highlights the 'Workplace Bots Collection' under 'Collections'. The main workspace shows a 'DELETE' request for the URL `https://graph.facebook.com/380194830120200_380419526764397`. The 'Headers' tab is selected, showing 8 headers. The 'Body' tab is selected, displaying a JSON response with three items: 1, 2, and 3. Item 2 contains the key-value pair `"success": true`. The status bar at the bottom right indicates a 200 OK response.

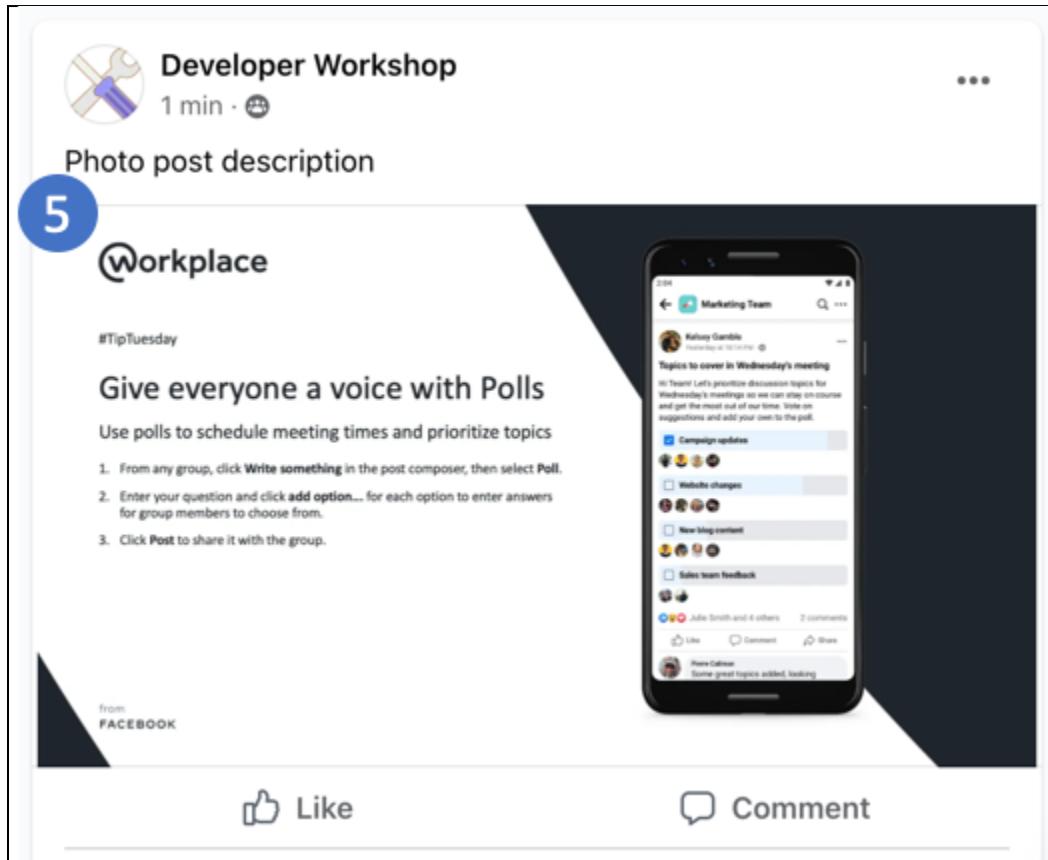
2. In the URL, replace {{post_id}} with your post id and check DELETE as a method.
3. Click Send.
4. Check if the post was deleted.

LAB - 6 Group Bots: Post with images and videos

Expand "Post with images and videos" in the Postman collection.

Post with an image

Group bots support adding an image on the post. Images can also make your bot look nice and more interactive.



Instructions:

1. Open Post with an image.

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, and Monitors. The main area displays a collection named 'Developer Workshop'. Under this collection, there's a folder 'Workplace Bots Collection' which contains 'Chat Bots', 'Group Bots', 'Post Management', and 'Post with images and videos'. The 'Post with images and videos' folder is expanded, showing four POST requests: 'Post with an image', 'Upload 1st photo unpublish...', 'Upload 2nd photo unpublish...', and 'Post with 2 images'. The 'Post with an image' request is selected. The request details show a POST method and a URL starting with https://graph.workplace.com/{{group_id}}/photos The 'Body' tab is selected, displaying raw JSON code:

```

1
2 →   "url": "{{photo_url}}",
3 →   "message": "Photo post description"
4

```

2. Replace {{group_id}} in the URL with your group id.
3. Check Body contents and replace {{photo_url}} to "http://wpdeveloperworkshop.herokuapp.com/imgs/Polis.png"
4. Click Send.
5. If the request is successful, check a new post with an image.

Post with multi images

Previous method supports a single image on the post. When you want to add multi images. You need 2 steps. The 1st step is uploading images which you want to use and the 2nd is making a post.

The screenshot shows the Facebook Workplace interface. At the top, there's a post from the 'Developer Workshop' group titled 'Signpost information with pinned posts'. A blue circle with the number '13' is overlaid on the post. Below this, another post from '#TipTuesday' is visible with the title 'Set reminders'. At the bottom of the screen, there's a comment input field with the placeholder 'Write a comment...' and a toolbar with icons for like, comment, and reply.

Developer Workshop
Just now ·

Post with multi images. **13**

Workplace

#TipTuesday

Signpost information with pinned posts

To pin a post to a group:

- From the group, find the post you'd like to pin.
- Click and select **Pin Post**.

The most recent pinned post will be shown at the top of the group feed.

To choose one post to be at the top of the pinned posts:

- Go to the pinned post you want to put on the top.
- Click and select **Move to Top**.

Workplace

#TipTuesday

Set reminders

For chat messages:

- Click from the message and then select **Remind Me**.
- Choose the time you want to be reminded about the message.

For posts:

- Click from the right-hand side of the post.
- Select **Remind me about this post**.
- Choose the time you want to be reminded about the post.

Like Comment

Dev Write a comment...

Instruction 1: Upload 1st photo unpublished

To prepare, upload the file you want to embed in your post.

1. Open Upload 1st photo unpublished

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Top Bar:** Home, Workspaces, API Network, Reports, Explore, Search Postman, Invite.
- Request Details:**
 - Method: POST
 - URL: https://graph.workplace.com/me/photos
 - Body tab is selected.
 - JSON content type is selected.
 - Body content (highlighted by a blue circle labeled '2'):

```

1
2 "url": "https://wpdeveloperworkshop.herokuapp.com/imgs/pinned_posts.png",
3 "published": "false"
4
      
```
- Response Headers:** 200 OK
- Bottom Tools:** Pretty, Raw, Preview, Visualize, JSON.

2. Check Body contents and replace {{photo_url}} to "https://wpdeveloperworkshop.herokuapp.com/imgs/pinned_posts.png"
 - a. "published" parameter should be "false". It means the image is reserved for posting.
If it is "true", it makes a post directly and can't reuse the image for post.
3. Click Send.
4. If the request is successful, Write down the id (photo id) you get in the HTTP Response. We'll use it in the last request.

Instruction 2: Upload 2st photo unpublished

5. Open Upload 2st photo unpublished

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Top Bar:** Home, Workspaces, API Network, Reports, Explore, Search Postman, Invite.
- Request Details:**
 - Method: POST
 - URL: https://graph.workplace.com/me/photos
 - Body tab is selected.
 - Body content (Pretty):


```
{
  "url": "https://wpdeveloperworkshop.herokuapp.com/imgs/Reminders.png",
  "published": "false"
}
```
 - Response status: 200 OK

6. Check Body contents and replace {{photo_url}} to [“https://wpdeveloperworkshop.herokuapp.com/imgs/Reminders.png”](https://wpdeveloperworkshop.herokuapp.com/imgs/Reminders.png)
7. Click Send.
8. If the request is successful, Write down the id (photo id) you get in the HTTP Response. We'll use it in the last request.

Instruction 3: Post with multi images

9. Open Post with multi images.

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections like Home, Workspaces, API Network, Reports, and Explore. Below these are collections for 'Workplace Bots Collection' containing 'Chat Bots', 'Group Bots' (with 'Post Management' and 'Post with images and videos' sub-sections), and 'Mock Servers'. Other sections include APIs, Environments, Mock Servers, Monitors, Flows, and History.

The main workspace shows a 'Post with 2 images' request. The method is 'POST' and the URL is 'https://graph.facebook.com/380194830120200/feed'. The 'Body' tab is selected, showing a JSON payload:

```

1  {
2   "message": "Post with multi images.",
3   "attached_media": [
4     {
5       "media_fbid": "103803918920657"
6     },
7     {
8       "media_fbid": "103802068920842"
9     }
10 }

```

Below the JSON, there are tabs for Body, Cookies (1), Headers (19), and Test Results. Under Body, there are Pretty, Raw, Preview, Visualize, and JSON dropdowns. The JSON dropdown is open, showing the raw JSON code:

```

1  {
2   "id": "380194830120200_382252299914453",
3   "post_supports_client_mutation_id": true
4 }

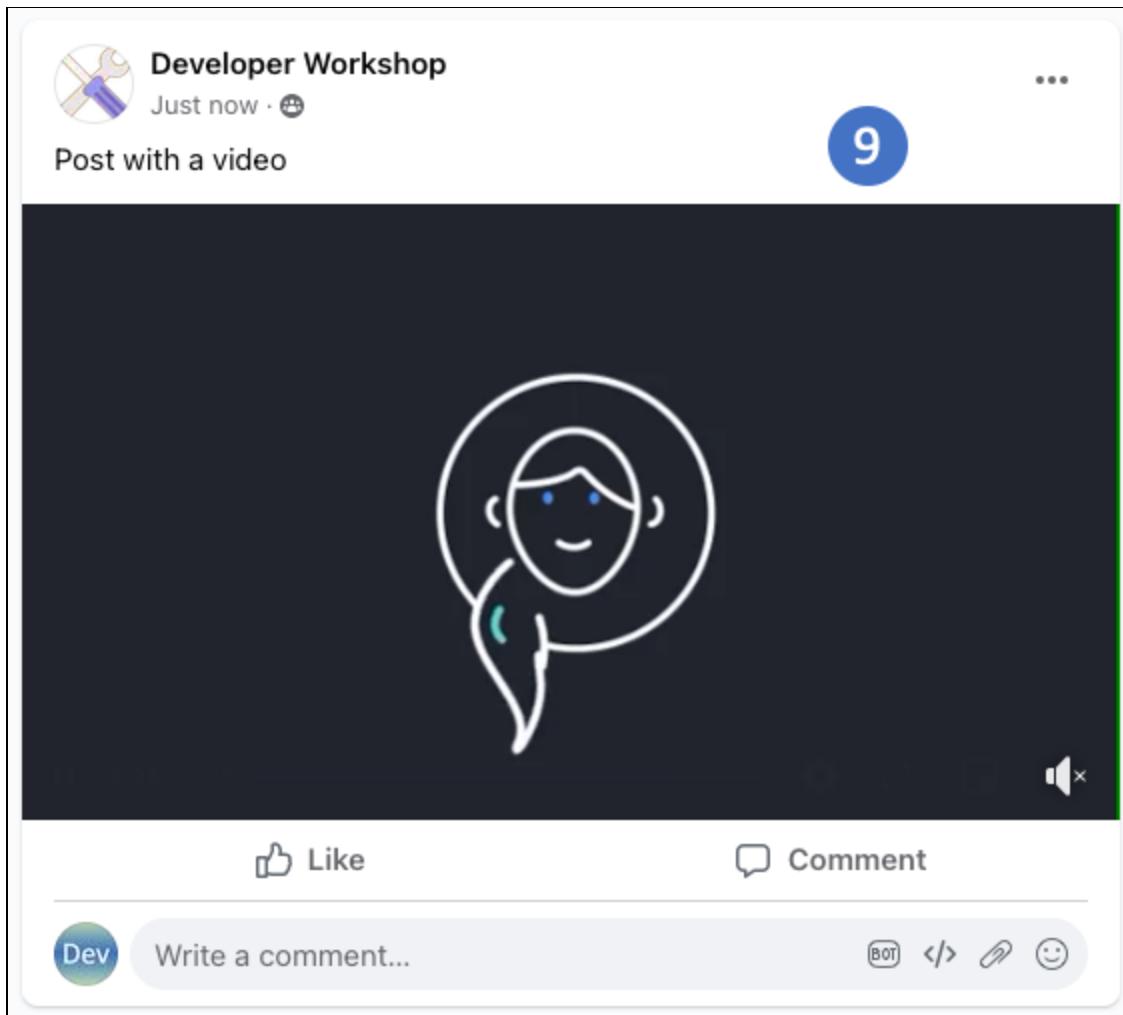
```

A blue circle with the number '10' is positioned above the URL, and another blue circle with the number '11' is positioned above the JSON body area.

10. Replace {{group_id}} in the URL with your group id.
11. Check Body content and replace {{photo_id}} with the photo id you received in the previous 2 steps.
12. Click Send.
13. If the request is successful, check a new post with 2 images.

Post with a video

Group bots also support adding a video on the post. Videos can also make your bot look nice and more interactive.



Instruction 1: Upload a video

To prepare, upload the video you want to embed in your post.

1. Open Upload a video.

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections like Home, Workspaces, API Network, Reports, and Explore. Below these are collections for Developer Workshop, APIs, Environments, Mock Servers, Monitors, Flows, and History. Under the Developer Workshop collection, there's a 'Workplace Bots Collection' which contains 'Chat Bots', 'Group Bots' (which is expanded), 'Post Management', and 'Post with images and videos'. Under 'Post with images and videos', there are four items: 'POST Post with an image', 'POST Upload 1st photo unpublish...', 'POST Upload 2nd photo unpublish...', and 'POST Post with multi images'. The last item is selected and highlighted in grey. To its right is a detailed view of a POST request for 'Upload video'.

Request Details:

- Method: POST
- URL: https://graph.facebook.com/me/videos
- Body tab is selected.
- Body type: form-data
- Params: none
- Authorization: (green dot)
- Headers: (10) (green dot)
- Body (green dot): Selected tab.
- Pre-request Script: none
- Tests: none
- Settings: none

Body Content:

KEY	VALUE	DESCR
<input checked="" type="checkbox"/> source	DeveloperWorkshop.mov	2
<input checked="" type="checkbox"/> no_story	true	
Key	Value	Description

Response:

200 OK

Pretty Raw Preview Visualize JSON

```

1   "id": "517763269702101"
2
3

```

4

2. Check Body contents and select a video file which includes demo material.
File name : DeveloperWorkshop.mov
 - a. "no_story" parameter should be "true" to be reserved for posting.
3. Click Send.
4. If the request is successful, Write down the id (video id) you get in the HTTP Response.
We'll use it in the last request.

Instruction 2: Post with a video

5. Open Post with a video.

The screenshot shows the Postman interface. On the left sidebar, under 'Collections', there is a 'Workplace Bots Collection' which includes 'Chat Bots', 'Group Bots' (which further includes 'Post Management' and 'Post with images and videos'), and other methods like 'Post with an image', 'Upload 1st photo unpublish...', 'Upload 2nd photo unpublish...', 'Post with multi images', 'Upload video', and 'Post with video'. The 'Post with video' method is selected. In the main panel, a POST request is being made to the URL <https://graph.facebook.com/380194830120200/feed>. The 'Body' tab is active, showing the following JSON payload:

```

1  {
2   "message": "Post with a video",
3   "attached_media": [
4     {
5       "media_fbid": "517763269702101"
6     }
7   ]
8 }
9

```

The 'JSON' tab at the bottom shows the resulting response body:

```

1  {
2   "id": "380194830120200_382826086523741",
3   "post_supports_client_mutation_id": true
4 }

```

6. Replace {{group_id}} in the URL with your group id.
7. Check Body content and replace {{video_id}} with the photo id you received in the previous 2 steps.
8. Click Send.
9. If the request is successful, check a new post with a video.

Notes: As in the previous chapter, you can include multiple videos in a post.

LAB - 7 Use Webhooks

In this LBA, you will experience how chatbots work using the methods you have learned so far. You will also learn how to set up a webhook for real-time conversations with chatbots.

The sample program will perform with a simple response, but you can customize the program to automate more complex processes.

Edit a sample PHP code

1. Open sample PHP code with text editor.

```
1 <?php
2 // Parameters
3 $access_token = '{{ACCESS_TOKEN}}';
4 $token = '{{VERIFY_TOKEN}}';
5 $url = "https://graph.facebook.com/me/messages?access_token=".$access_token;
6
7 if($_GET['hub_challenge']){
8     if(isset($_GET['hub_verify_token'])){
9         if($token == $_GET['hub_verify_token']){
10             echo $_GET['hub_challenge'];
11             exit;
12         }else{
13             error_log("Mismatch Token.");
14         }
15     }
16 }
17
18 $req = file_get_contents('php://input');
```

2. Copy {{ACCESS_CODE}} which you created in the Setup section.
3. Create {{VERIFY_TOKEN}}. Any string is fine. This sample uses “DeveloperWorkshop”.
4. Save the changes.

Heroku deployment

Heroku is a cloud platform that lets companies build, deliver, monitor and scale apps. It's the fastest way to go from idea to URL, bypassing all those infrastructure headaches.

HEROKU Products Marketplace Pricing Documentation Support COVID-19 More

Log in or Sign up

NEW

Salesforce Functions with Heroku Data

Access data from Heroku Postgres and leverage Salesforce data through Heroku Connect. Publish events with Apache Kafka for Heroku to integrate back office actions with connected data stores. Share and optimize data access overheads with Heroku Redis.

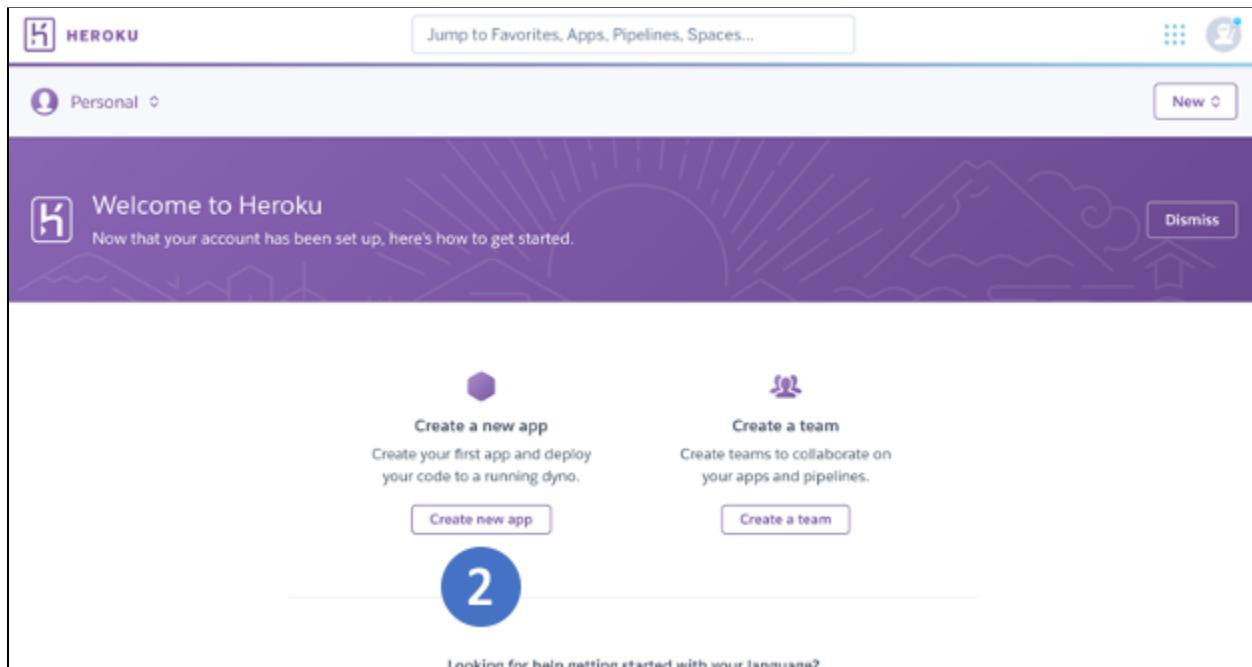
Learn More

Sign Up for Free

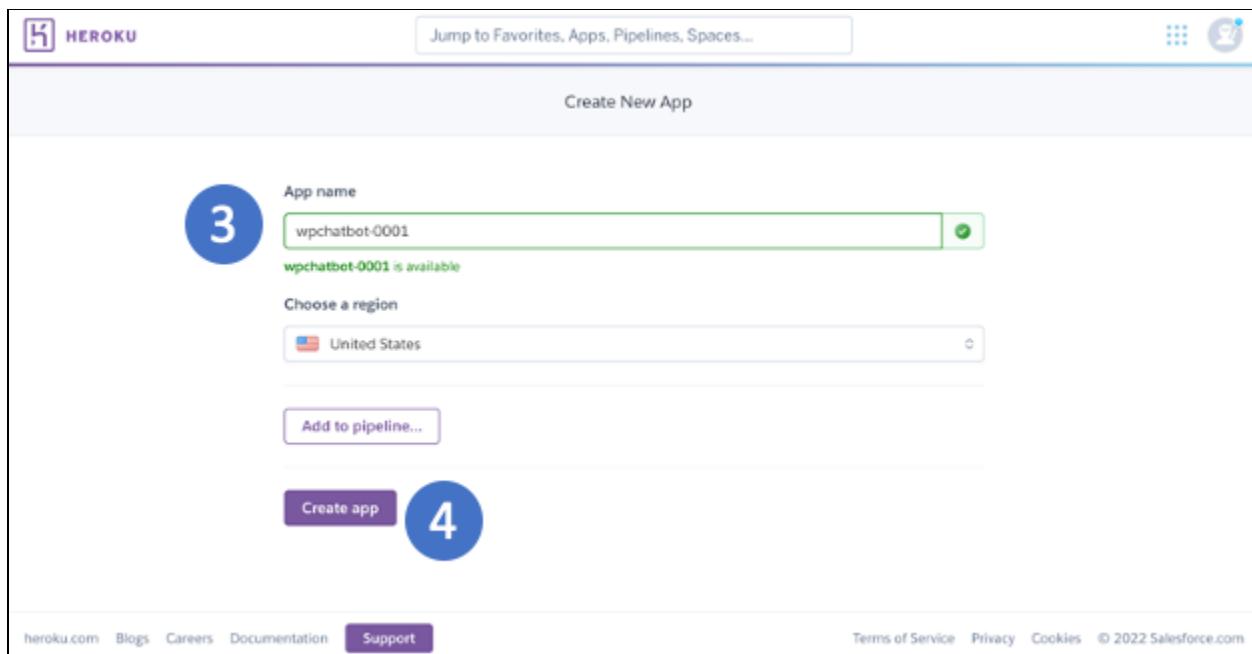
1

Developer Workshop Workplace from Meta

1. Access to <https://www.heroku.com/home> and create a new Heroku account if you do not have it.



2. Login to Heroku and click Create new app.



3. Type App name. This sample is “wpchatbot-0001”.

Notes : The app name must be unique. Please give it a unique name and substitute your app name for the following chapters and proceed.

4. click Create app.

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Deploy using Heroku Git

Use git in the command line or a GUI tool to deploy this app.

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Create a new Git repository

Initialize a git repository in a new or existing directory

```
$ cd my-project/  
$ git init  
$ heroku git:remote -a wpchatbot-0001
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys. Please follow the instructions here.

5. Scroll down and you will see the instructions for implementation.

- a. Install the Heroku CLI.

```
[ec2-user@ip-172-31-9-193 my-project]$ ls -l  
total 8  
drwxr-xr-x 2 ec2-user ec2-user 142 Mar 30 05:55 imgs  
-rw-r--r-- 1 ec2-user ec2-user 4644 Mar 30 08:28 index.php  
[ec2-user@ip-172-31-9-193 my-project]$
```

- b. Open Command Line Interface.
- c. Copy my-project folder and check a sample PHP program and imgs folder under my-project.

```
remote: -----> Compressing...
remote:          Done: 15.5M
remote: -----> Launching...
remote:          Released v4
remote:          https://wpchatbot-0001.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/wpchatbot-0001.git
  2377c7c..ccd2e37  master -> master
[ec2-user@ip-172-31-9-193 my-project]$
```

- d. Deploy your code.

Webhooks Setup

Webhooks enables custom integration apps to subscribe to events in Workplace and receive updates in real time. When a change occurs in Workplace, an HTTPS POST request is sent to a callback URL for each custom integration app that's subscribed to the relevant webhook topic.

1. Go to the Admin Panel
2. Click Integrations, then Open Developer Workshop which you created in this workshop.
3. Move to the webhooks section and edit Page.
The Edit Custom Integration dialog provides tabs for each of the webhook topics available to apps on Workplace.

The screenshot shows the Facebook Admin Panel with the 'Integration details' sidebar on the left. The 'Webhooks' tab is selected. On the right, the 'Configure webhooks' dialog is open, showing a list of webhook triggers:

- mention (checkbox)
- messages (checkbox, selected)
- message_reads (checkbox)
- message_deliveries (checkbox)
- messaging_postbacks (checkbox)
- messaging_referrals (checkbox)

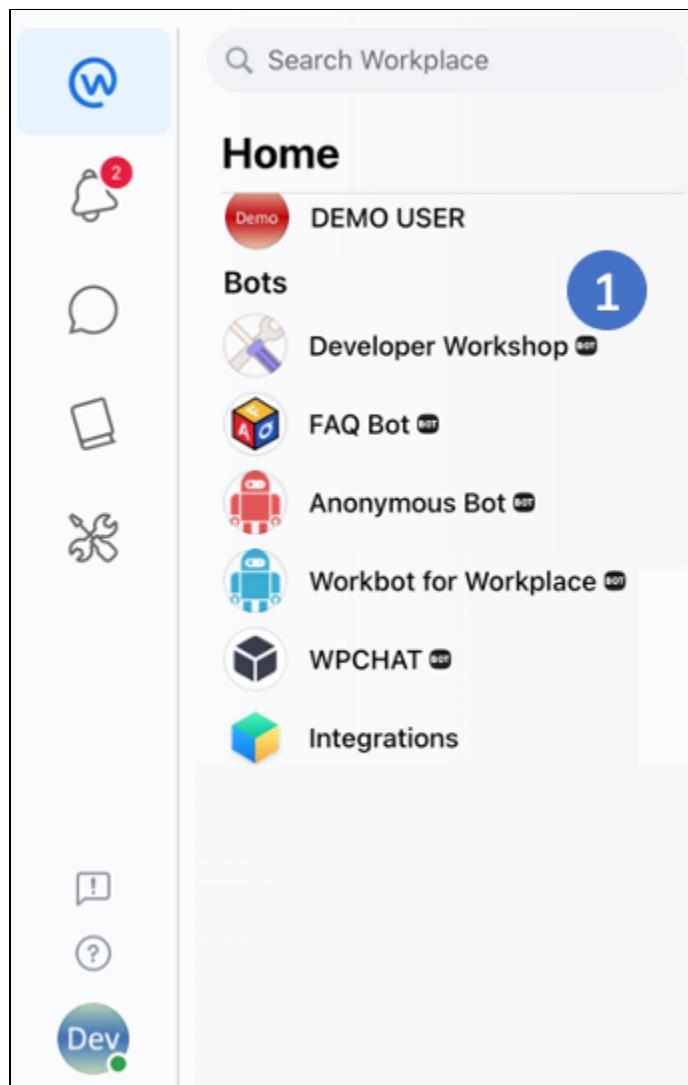
The 'Verify token' field contains 'DeveloperWorkshop'. Step numbers 4, 5, 6, and 7 are overlaid on the interface to indicate the sequence of actions:

- 4: Callback URL: https://wpchatbot-0001.herokuapp.cc
- 5: Selected webhook trigger: messages
- 6: Verify token: DeveloperWorkshop
- 7: Save button

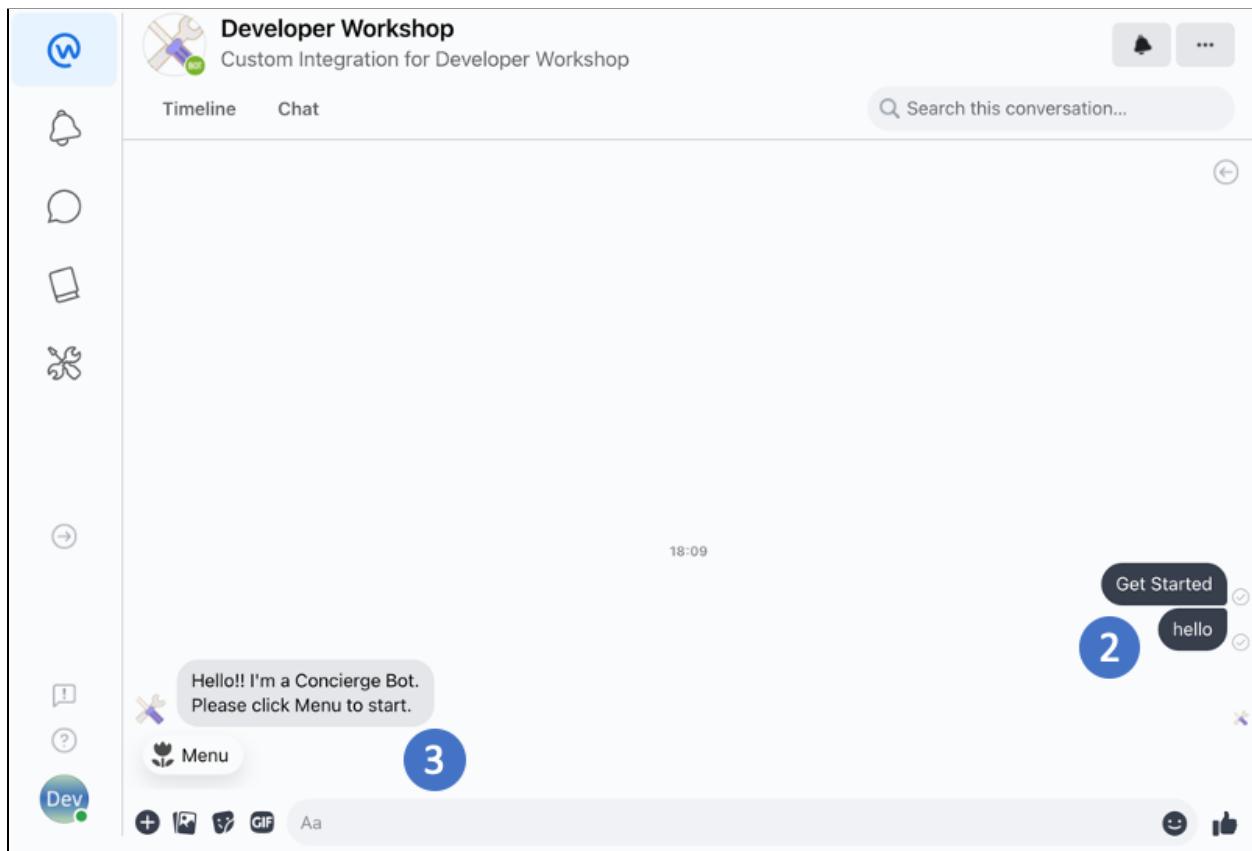
4. Set Callback URL : <https://{{Heroku Application Name}}.herokuapp.com/index.php>
{{Heroku Application Name}} is your application name.
Example : <https://wpchatbot-0001.herokuapp.com/index.php>
5. Check messages.
This is triggered when a custom integration Page (bot) is messaged in Workplace Chat.
6. Set Verify token which you set on PHP code in previous section.
Example : **DeveloperWorkshop**
7. Click Save and finish configuration.

Run Chatbot conversation

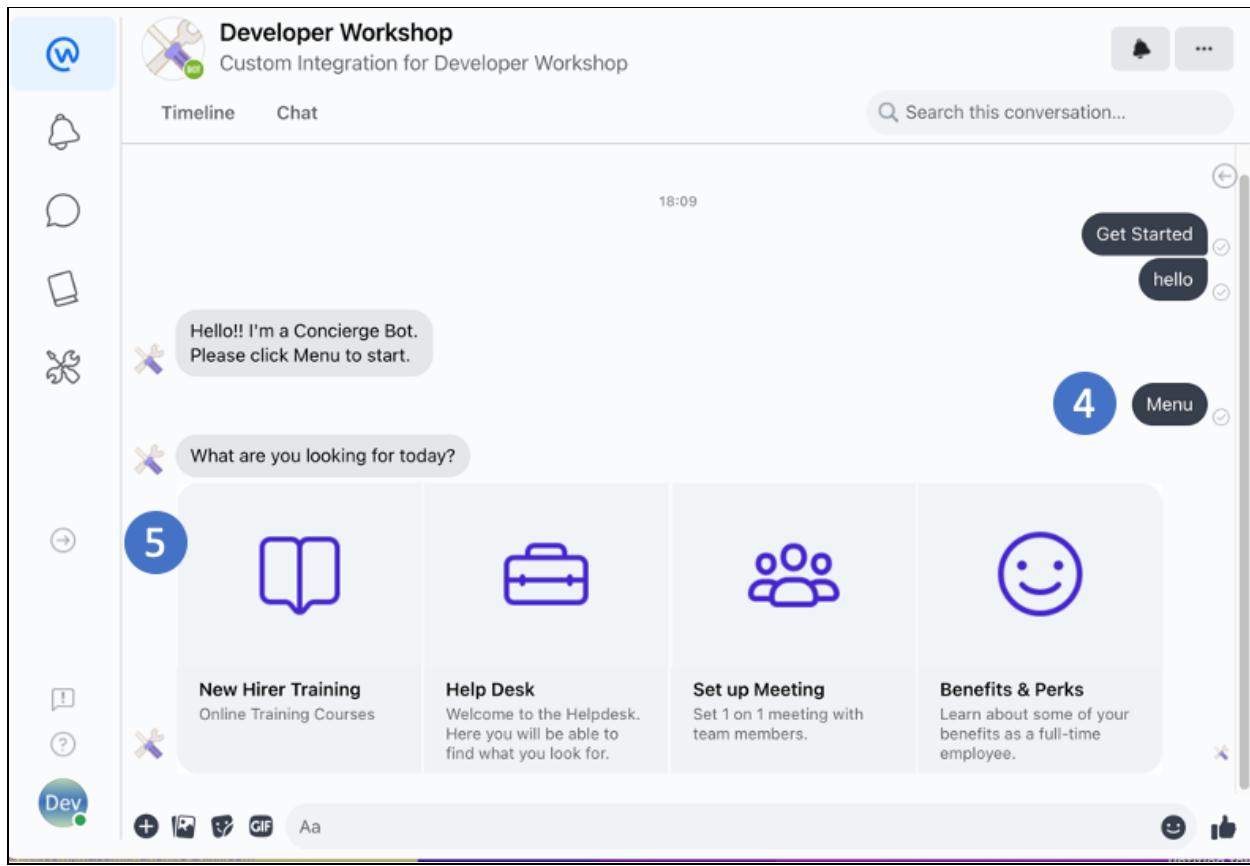
The application that receives the webhook applies Lab1 to Lab4 to establish a simple conversation in real time. say Hello and Hey and the conversation begins with a greeting. Requesting a menu, a list of services that can be provided is displayed.



1. You can find the Bots in the left panel navigation and open Developer Workshop Bot.



2. Type “Hello” or “hey”
3. The chat bot responds back in real time.



4. Chat bot asks you to click the "Menu" button so click Menu button.
5. Chat bot shows the list of services. Click the menu items, you jump to its page.

Appendix

Workplace Developer Document

Learn about building integrations for Workplace from Meta, whether you are part of a product company interested in offering a Third Party App or whether you are doing in-house development for a Workplace customer via Custom Integrations.

<https://developers.facebook.com/docs/workplace>

Third Party Apps

Third Party Apps allow Independent Software Vendors (ISVs) to integrate their SaaS and PaaS products with Workplace. Once reviewed and approved by the Workplace team, these apps can then be installed by any Workplace customer to deliver valuable automation.

<https://developers.facebook.com/docs/workplace/third-party-apps/introduction>

Sample Apps for Workplace

This is a collection of samples containing examples of Graph API doc, integrations and support scripts.

<https://github.com/fbsamples/workplace-platform-samples>