
vxmessenger Documentation

Release 1.2.5

Praekelt Foundation

November 29, 2016

1	Getting Started	3
2	Hook up an Application to your Messenger integration	5
3	Richer Templates	7
3.1	A Button Reply	7
3.2	A Generic Reply	8
3.3	A List Reply	8
3.4	A Quick Reply	9
4	Message format	11
4.1	Supported webhooks	11

All of Vumi's applications can be surfaced on Messenger with the Messenger Transport. It provides a great experience for interactive mobile conversations at scale.

Getting Started

Install **Junebug**, the standalone Vumi transport launcher and the Facebook Messenger Transport:

```
$ apt-get install redis-server rabbitmq-server
$ pip install junebug
$ pip install vxmessenger
```

Launch the Junebug service with the Vumi Messenger channel configured:

```
$ jb -p 8000 \
  --channels facebook:vxmessenger.transport.MessengerTransport \
  --logging-path .
```

Using the template, below and update your `FB_PAGE_ID`, `FB_ACCESS_TOKEN` and save it as a file called `config.json`:

```
{
  "type": "facebook",
  "amqp_queue": "messenger_transport",
  "public_http": {
    "enabled": true,
    "web_path": "/api",
    "web_port": 8051
  },
  "config": {
    "web_path": "/api",
    "web_port": 8051,
    "noisy": true,
    "page_id": "YOUR_FB_PAGE_ID",
    "retrieve_profile": false,
    "outbound_url": "https://graph.facebook.com/v2.6/me/messages",
    "access_token": "YOUR_FB_ACCESS_TOKEN"
  }
}
```

Post it to Junebug to start the channel:

```
$ curl -X POST -d@config.json http://localhost:8000/channels/
```

You're now able to communicate with Facebook's Messenger API and can offer bot access to your Vumi application. Facebook will want to verify your application, for that to work make sure it's served over SSL. The API URL is:

```
http://localhost:8051/api
```

If you've used a different `web_port` and `web_path` parameter you'll need to update the URL accordingly.

Note: There is also a Dockerfile available that you can customise to run Junebug in a Docker container: <http://github.com/praeeltfoundation/docker-junebug>

The Docker container includes Nginx and offers the [Junebug](#) API under the `/jib/` endpoint, all other transports are made available from the root path. For the example above the endpoint would be `/api` on port 80.

Hook up an Application to your Messenger integration

All Vumi applications can be surfaced on Facebook Messenger as bots, how about hooking up a simple game of hangman?:

```
$ twistd -n vumi_worker \  
  --worker-class=vumi.demos.hangman.HangmanWorker \  
  --set-option=random_word_url:http://randomword.setgetgo.com/get.php \  
  --set-option=transport_name:messenger_transport \  
  --set-option=worker_name:hangman
```

Javascript Sandbox applications are also available. Check out some of the examples below:

Sample FAQ browser <https://github.com/smn/faqbrowser-docker>

Sample Service rating application <https://github.com/smn/servicerating-docker>

Note: Do you want to expose multiple applications within a single Bot? The Vumi Application Router allows you to do exactly that, have a look at the [example router specifically for Facebook Messenger](#).

Richer Templates

The Vumi Messenger Transport allows one to use the richer templates available, including texts, images, hyperlinks and buttons.

To make use of these add the relevant `helper_metadata` to your outbound Vumi message:

3.1 A Button Reply

Please be aware of the [limitations](#) that Facebook applies to these messages. A call to action may only have a maximum of 3 buttons and character count limits apply.

```
self.publish_message(  
    helper_metadata={  
        'messenger': {  
            'template_type': 'button'  
            'text': 'The accompanying text with the button',  
            'buttons': [{ # Up to 3 buttons  
                'type': 'postback', # defaults to postback if not specified  
                'title': 'Button 1',  
                'payload': {  
                    # In here you can put whatever you want to  
                    # 'content' and 'in_reply_to' will go into the standard vumi message  
                    'content': 'The content expected when a button is pressed', # This can be left blank  
                    'in_reply_to': 'The ID of the previous message', # This can be left blank  
                    # Anything else will end up in transport_metadata.messenger and helper_metadata.  
                    'anything_extra': 'Bonus!' }  
                },  
            }, {  
                'type': 'web_url',  
                'title': 'Button 2',  
                'url': 'http://some.url'  
            }, {  
                'type': 'phone_number',  
                'title': 'Button 3',  
                'url': '+271234567'  
            }  
        ]  
    }  
})
```

3.2 A Generic Reply

Please be aware of the [limitations](#) that Facebook applies to these messages. A call to action may only have a maximum of 3 buttons, 10 elements, and character count limits apply.

```
self.publish_message(  
    helper_metadata={  
        'messenger': {  
            'template_type': 'generic'  
            'elements': [{ # Up to 10 elements  
                'title': 'The title',  
                'subtitle': 'The subtitle', # This can be left blank  
                'image_url': 'The image_url to use', # This can be left blank  
                'item_url': 'The url if you click on the element', # Optional, Cannot be used with default_action  
                'default_action': { # The action if you click on element, optional, Cannot be user w  
                    'url': 'http://test',  
                    'webview_height_ratio': 'compact|tall|full', # Leave out for regular link  
                    'messenger_extensions': True, # If you are using the js extensions api, optiona  
                    'fallback_url': 'http://moo' # Fallback URL if target doesn't support js extensi  
                                                # will use url if not found, optional  
                },  
                'buttons': [{ # Up to 3 buttons  
                    'type': 'postback', # defaults to postback if not specified  
                    'title': 'Button 1',  
                    'payload': {  
                        # In here you can put whatever you want to  
                        # 'content' and 'in_reply_to' will go into the standard vumi message  
                        'content': 'The content expected when a button is pressed', # This can be le  
                        'in_reply_to': 'The ID of the previous message', # This can be left blank  
                        # Anything else will end up in transport_metadata.messenger and helper_metada  
                        'anything_extra': 'Bonus!'   
                    }  
                }, {  
                    'type': 'web_url',  
                    'title': 'Button 2',  
                    'url': 'http://some.url'  
                }, {  
                    'type': 'phone_number',  
                    'title': 'Button 3',  
                    'url': '+271234567'  
                }, {  
                    # Share this post with friends. Only available in Generic Reply  
                    'type': 'element_share'  
                }  
            ]  
        }  
    }  
)
```

3.3 A List Reply

Please be aware of the [limitations](#) that Facebook applies to these messages. A call to action may only have a maximum of 1 button per element, 2-4 elements, 1 extra button, and character count limits apply.

```
self.publish_message(  
    helper_metadata={  
        'messenger': {
```

```

'template_type': 'list'
'top_element_style': 'compact|large', # Defines top element style
                                     # Defaults to compact if not specified
'elements': [{ # 2 - 4 elements
    'title': 'The title',
    'subtitle': 'The subtitle', # This can be left blank
    'image_url': 'The image_url to use', # This can be left blank
    'default_action': { # The action if you click on element, optional
        'url': 'http://test',
        'webview_height_ratio': 'compact|tall|full', # Leave out for regular link, optional
        'messenger_extensions': True, # If you are using the js extensions api, optional
        'fallback_url': 'http://moo' # Fallback URL if target doesn't support js extensions
                                     # will use url if not found, optional
    },
    'buttons': [{ # Up to 1 button
        'type': 'postback', # defaults to postback if not specified
        'title': 'Button 1',
        'payload': {
            # In here you can put whatever you want to
            # 'content' and 'in_reply_to' will go into the standard vumi message
            'content': 'The content expected when a button is pressed', # This can be left blank
            'in_reply_to': 'The ID of the previous message', # This can be left blank
            # Anything else will end up in transport_metadata.messenger and helper_metadata
            'anything_extra': 'Bonus!'
        }
    }, {
        'type': 'web_url',
        'title': 'Button 2',
        'url': 'http://some.url'
    }, {
        'type': 'phone_number',
        'title': 'Button 3',
        'url': '+271234567'
    }
    ]
}],
'buttons': [{ # Up to 1 button at bottom of list
    ...
}
}
})

```

3.4 A Quick Reply

Please be aware of the [limitations](#) that Facebook applies to these messages. A call to action may only have a maximum of 10 buttons and character count limits apply.

A Quick Reply is similar to a button reply, but Messenger shows the buttons temporarily and they disappear after being pressed. You can also have up to 10 buttons. Only postback buttons is allowed in this mode.

```

self.publish_message(
    helper_metadata={
        'messenger': {
            'template_type': 'quick'
            'text': 'The accompanying text with the button',
            'quick_replies': [{ # Up to 10 quick replies
                'type': 'text', # defaults to text if not specified
            }
        ]
    }
)

```

```
'title': 'Button 1',
'image_url': 'The image_url to use', # This can be left blank
'payload': {
    # In here you can put whatever you want to
    # 'content' and 'in_reply_to' will go into the standard vumi message
    'content': 'The content expected when a button is pressed', # This can be left blank
    'in_reply_to': 'The ID of the previous message', # This can be left blank
    # Anything else will end up in transport_metadata.messenger and helper_metadata.
    'anything_extra': 'Bonus!'
}
}, {
    'type': 'location',
}]
}
})
```

Message format

Due to some extra features of the messenger platform, there is some extra data that you may need to pay attention to:

transport_metadata:

Contains a dict `messenger` with the following keys:

mid: Messenger message id.

attachments: List containing dictionaries as such:

```
{
  "type": "image",
  "payload": {
    "url": "IMAGE_URL"
  }
}
```

optin: Dict containing a `ref` key, which is the `PASS_THROUGH_PARAM` as defined by:

https://developers.facebook.com/docs/messenger-platform/plugin-reference#send_to_messenger

Other items defined in payload: e.g. `"anything_extra": "Bonus"`

helper_metadata:

Contains a dict `messenger` with the same data as `transport_metadata` and the user profile as such:

Note: only if `retrieve_profile` is configured as `true`

```
{
  "first_name": "Firstname",
  "last_name": "Lastname",
  "profile_pic": "IMAGE_URL"
}
```

4.1 Supported webhooks

messages: Standard conversational messages & attachments.

messaging_postbacks: Postback buttons.

messaging_optins: Send-to-Messenger / authentication callback.