

FileMaker HipChat Sender

Author: Donald L. Merand

Sends messages to a HipChat room from FileMaker, either via a FileMaker API script, or manually via a layout.

Setup

1. Make sure you've got a HipChat account and API Token.
2. Open up the HipChat.fp7 file.
3. Edit the custom function AUTHTOKEN and replace the return value with your API token obtained in step one.
4. *Note: I'm using the convention of database-wide constants being stored in custom functions. I find the syntax very clean in my calculation fields.*
5. You probably also want to edit the DEFAULTROOMID custom function to be your preferred room for posting automated / FileMaker messages in your HipChat account.

Sending Messages

You can send messages from the HIPCHAT layout, which is pretty obvious. You'll need to add at least one default room. See the section below on adding rooms.

To send a message, select a room, make sure to pick a "from" account (which you can make up), type in a message, and click "send". In the "result" box, you'll see whether the message went through, or whether HipChat's API returned any errors.

The "URL" tab shows you what is happening behind-the-scenes. We're creating a web page with a form full of hidden fields, then submitting it to HipChat's API when the document loads. The "result" tab is just a webviewer into which the HTML gets loaded.

Creating and Importing Rooms

You can create rooms by hand if you know the room IDs of your HipChat rooms. You can get a list of those for your account (once logged in to HipChat) at this link.

I've also included a ruby script which will give you a CSV list of all room_ids and room_names for your account. To use this script:

1. Open up your terminal-o'-choice
2. Type `cd path/to/hipchat`

3. Open up the `hipchat_rooms.rb` file in your text editor-o'-choice (make sure it's not something crazy like TextEdit that'll save the file as HTML. My recommendation? Use vim).
4. Find the line that says `auth_token = "YOUR AUTH TOKEN HERE"`. Change the part in quotes to your API Authorization token, obtained way up above.
5. Now save the file, go back to the command line, and type `ruby hipchat_rooms.rb`.
6. By default this command will output to STDOUT (the command line). If you wanted to save to a file, say `rooms.csv`, then type `ruby hipchat_rooms.rb > rooms.csv`.
7. Importing the rooms list is a standard FileMaker import. Match up based on `room_id` and you should be all set.

Known Bugs/Quirks

The *truly* cross-platform approach of using a web viewer to send the message has the distinct disadvantage that you have to wait for the webviewer to load before the message gets sent. If you're in an automated environment where you're sending API calls to the HipChat sender at a higher rate than the webviewer can load the pages, you're going to miss messages. In practice, this means that you can't send more than one message every 2–3 seconds if you don't want them to get swallowed.

If this is a major concern, I've put in a script that'll use CURL instead to post a message. The downside of this approach is that it requires OS X (and CURL), so it's not cross-platform. It's fast though.

Credits

- Interface icons come from The Noun Project