

DEVELOPER NEWSLETTER

In this newsletter, we cover the recent Python client launch in the Symphony Generator, RSA authentication with REST clients, sneak peek on the upcoming v1.52 platform release, and exciting developer events.

Python Client Now Available in the Symphony Generator

You can now choose Python as your preferred language in our Symphony Generator. Python has the same configuration as other available clients, including Java, .Net, and Node.JS.

Follow the steps below to get it installed in Symphony Generator, in your choice of IDE:

1. **Install and set up** your Symphony Generator
2. Run Symphony Generator **\$ yo symphony**, select Python when prompted, and follow the steps to configure the client
3. Select the desired authentication method, **using RSA or certificates**. Follow the steps accordingly to generate a config file.
4. Once the bot code is generated, **install** the Python client by running **\$ pip install -r requirements.txt** to bring all the necessary dependencies for your bot to run
5. Navigate to your config.json file and ensure the path to your public/private key (for RSA authentication) or .p12 certificate (for certificates) is correct

```
2 from sym_api_client_python.auth.rsa_auth import SymBotRSAAuth
3 from sym_api_client_python.clients.SymBotClient import SymBotClient
4 from sym_api_client_python.listeners.imListenerTestImp import IMListenerTestImp
5 from sym_api_client_python.listeners.roomListenerTestImp import RoomListenerTestImp
6 #debug logging --> set to debug --> check logs/example.log
7 import logging
8 logging.basicConfig(filename='sym_api_client_python/logs/example.log', format='%(asctime)s - %(name)s - %(levelname)s', level=logging.INFO)
9 logging.getLogger("urllib3").setLevel(logging.WARNING)
10 #main() acts as executable script --> run python3 hello.py to start Bot...
11
12 #adjust global variable below to auth either using RSA or certificates
13
14 def main():
15     print('hi')
16     #RSA Auth flow: pass path to rsa_config.json file
17     configure = SymConfig('sym_api_client_python/resources/rsa_config.json')
18     configure.loadFromRSA()
19     auth = SymBotRSAAuth(configure)
20     auth.authenticate()
21     #initialize SymBotClient with auth and configure objects
22     botClient = SymBotClient(auth, configure)
23     #initialize datafeed service
24     DataFeedEventService = botClient.getDataFeedEventService()
25     #initialize listener classes and append them to DataFeedEventService class
26     #these listener classes sit in DataFeedEventService class as a way to easily handle events
27     #coming back from the DataFeed
28     imListenerTest = IMListenerTestImp(botClient)
29     DataFeedEventService.addIMListener(imListenerTest)
30     roomListenerTest = RoomListenerTestImp(botClient)
31     DataFeedEventService.addRoomListener(roomListenerTest)
32     #create data feed and read datafeed recursively
33     print('starting datafeed')
34     DataFeedEventService.startDataFeed()
35
36 if __name__ == "__main__":
37     main()
```

Simple bot written in Python using our new Python client

6. Run the generated bot `$ python3 main.py`

More information on Python Client, including setup examples [here](#).

Using RSA Authentication with REST Clients

Symphony recommends all developers use [RSA Authentication](#) as the authentication method moving forward. RSA authentication allows you to quickly authenticate with the system, without the need to involve the PKI Infrastructure team.

The authentication process requires the following steps:

1. The user creates a public/private RSA key pair
2. The admin imports the public key into the pod using the Admin Console or public APIs
3. The user creates a short-lived JWT token and signs it with their private key
4. On a call to an authentication endpoint, the server checks the signature of the JWT against the public key and returns an authentication token

For full description of each step, visit our [RSA Bot Authentication Workflow](#) documentation.

Sneak Peak! Upcoming Symphony v1.52 Platform Release

There are a number of new platform features that will be released in the upcoming Symphony v1.52. Three key updates:

1. We have released APIs to create allowed [URI protocols](#) in your pod
2. MessageML now supports complex table formatting and [styling](#), providing you increased control over how your tables look. Preview this on messageML [rendering tool](#)
3. Additional parameters have been added to message search enabling you to more easily and accurately search for messages via [APIs](#)

View the [full list of updates](#) and the [swagger](#) for this upcoming release.

SYMPHONY

DEVELOPER NEWSLETTER

Developer Events

You are invited to one of the biggest innovation days for Symphony - our 2018 Hackathons! On September 27, we will host three regional hackathons as part of our [Innovate 2018 Hackathon](#) series. At these hackathons, you'll get to develop new workflows and applications while learning best practices from other talented developers who are innovating on Symphony. If you haven't already, [register](#) to participate in New York, London, or Paris.

Join one of our [Symphony Developer Meetup Groups](#) to receive updates on future developer events near you! We'll also be hosting developer meetups in Hong Kong and Singapore in October. [Register now](#) and event details will follow. We look forward to seeing you there!

The Developer Documentation, the instructions provided in this Symphony Developer Newsletter and the Symphony Generator (collectively, the "bot materials") are each provided "as is" without warranty of any kind (including without limitation, any warranty of merchantability or fitness for a particular purpose or non-infringement), and as such shall not be considered a "Symphony Service," as such term is used and defined in the services agreement between you or firm and Symphony Communication Services, LLC ("Symphony"). This means, among other things, that (I) Symphony makes no representations or warranties, express or implied, with respect to any matter relating to the bot materials; (II) Symphony is under no obligation to provide support or maintenance for the bot materials; and (III) Symphony disclaims all liability for or with respect to your or your firm's access to or use of the bot materials, and under no circumstances and under no legal theory, whether in tort, contract, or otherwise, will Symphony be liable to you or your firm (i) for any indirect, special, incidental, or consequential damages, (ii) for punitive damages, (iii) for damages for lost profits, lost sales, or business interruption of any character, in each case even if you have been advised, knew or should have known of the possibility of such damages. The Bot Materials are subject to change without notice and are for information and illustrative purposes only. None of the Bot Materials is, and should not be regarded as "investment advice" or as a "recommendation" regarding a course of action, including without limitation as those terms are used in any applicable law or regulation. The Bot Materials are provided with the understanding that with respect to the Bot Materials you will make your own independent decision with respect to any course of action in connection herewith, as to whether such course of action is appropriate or proper based on your own judgment and your specific circumstances and objectives.

September 2018