SYMPHONY

DEVELOPER NEWSLETTER

This newsletter covers our Innovate Asia Hackathons recap, click to call integration, agent server load balancing, and recent developer events.

Recap from Innovate Asia Hackathons 2019

Last month, over 140 developers across Hong Kong, Singapore and Tokyo faced off to build the Most Cutting Edge Technical Development or the Most Impactful Business Automation. Our hosts, **HKEX** in Hong Kong and **Deutsche Bank** in Singapore, also sponsored additional prizes for Best Collaboration Workflow Between Two or More Companies and Highest Impact on End-to-End Coordination Across Banks & Clients, respectively. Teams ranging from global banks to software startups to universities gathered to build innovations on and for Symphony, leveraging Symphony's open APIs, unique chat interface, and unique security model.

Hackathon 2019
HONG KONG
Hackathon 2019
HONG KONG • SINGAPORE • TOKYO
Hackathon 2019
HONG KONG • JAPAN • SINGAPORE
Most Impactful Business Automation Development

The winning developments for each category were:

Most Cutting-Edge Technical Development

- Question and Answer bot by Diamond team from JP Morgan (Hong Kong)
- Swift bot by SS Hackers team from Standard Chartered (Singapore)
- Question and Answer bot by Tendon team from Goldman Sachs (Tokyo)

Most Impactful Business Automation Development

- Consent Bot by Synergy team from Goldman Sachs (Hong Kong)
- RFQ distributor and tracker bot by DEEP WATeRS team from Deutsche Bank (Singapore)
- RegBot by NKM^2 team from Nomura (Tokyo)

June 2019

Executive Host Award: *Best Collaboration Workflow Between Two or More Companies*

- Margin call alert bot by H2O team from HKEX & HSBC (Hong Kong)

Executive Host Award: *Highest Impact on end-to-end coordination across banks & clients using Symphony*

- KYC onboarding, trade status, and knowledge collation bot by Bitter Sweet Symphony team from Deutsche Bank (Singapore)

The workflow solutions presented covered a wide range of use cases on Trading, Sales, Research, Compliance, Technology and Operations Functions.

Check out the **detailed summary** in our **2019 Hackathon blog** to see how these bots work!

Develop: Click to Call Integration

Symphony provides audio/video calling and screen-sharing via its rich Real-Time Communication (RTC) capabilities included in the core product. If your organization has chosen to use an existing soft-phone provider rather than Symphony's RTC solution, you can opt to integrate it with Symphony via a sample Extension App. This sample app adds a call button into 1-to-1 IMs, sends the user to the soft-phone app and injects the phone number for the other user that's listed in Symphony. The sample is Java-based and is available at this **GitHub repository link**.

To get started, clone the repository and open `src/main/resources/static/js/controller.js`. The only edit required is changing the protocol on line 37 from `dialpad:` to the one used by your soft-phone provider. For example, this could be `sip:` for Cisco phones.

```
window.location.assign(`dialpad:${payload.user.phone}`);
```

If you are comfortable with the Java + Spring Boot implementation, you can deploy this as-is and modify `src/main/resources/static/bundle.json` with the details of the deployed location. Otherwise, feel free to switch this to any other server-side implementation and retain the extension app bits in the `src/main/resources/static` directory.

Develop: Agent Server Load Balancing using Java SDK Client

When many bots are deployed in your environment or if high availability of bots is critical to your workflows, you may have multiple API Agents installed on-premise. The release version 1.0.23 of the

Java SDK Client provides a built-in support for Agent Server Load Balancing. Three modes are supported: round-robin, random and external. Both round-robin and random modes are handled by the SDK itself and use a predefined list of agent servers. External mode assumes the use of a hardware, cloud provider or DNS-based solution. There is also support for sticky sessions so bots that are reliant on reading datafeed APIs are bound to a specific agent server.

To use agent server load balancing in the Java SDK Client, a separate configuration file is created, specifying the following required parameters:

```
{
   "loadBalancing": {
       "method": "random", // or roundrobin or external
       "stickySessions": true
   },
   "agentServers": [
       "sym-agent-01.my-company.com",
       "sym-agent-02.my-company.com",
       "sym-agent-03.my-company.com"
   ]
}
```

Note that any bot that creates and reads the datafeed APIs require enabling sticky sessions. Setting this to false will result in a bot potentially requesting a datafeed from a server different from the one it was created on, returning an error.

If external is chosen as the method, only a single frontend address needs to be specified in the `agentServers` array - all subsequent entries are ignored. There also needs to be an additional parameter in each agent server's startup script to expose an additional REST endpoint providing the actual fully-qualified hostname for each server. This is not required for random or round-robin.

```
exec java $JAVA_OPTS -Dhost.name=sym-agent-01.my-company.com ...
```

The load balancing configuration file then gets loaded in your main class and added as an extra parameter in the `initBot()` method to get a `SymBotClient`.

```
// Option 1
URL lbUrl = getClass().getResource("lb-config.json");
SymLoadBalancedConfig lbConfig =
    SymConfigLoader.loadLoadBalancerFromFile(lbUrl.getPath());

// Option 2
InputStream lbConfigStream = getClass().getResourceAsStream("/lb-config.json");
SymLoadBalancedConfig lbConfig = SymConfigLoader.loadLoadBalancer(lbConfigStream);
```

```
// Inject lbConfig into initBot
SymBotClient botClient = SymBotClient.initBot(config, botAuth, lbConfig);
```

The SDK then handles agent server connection failures on startup or during any API call by rotating the current agent server. This would be the next server on the list for the round-robin method, a random server on the list for the random method or the next server assigned by the external load balancer for the external method.

Developer Events

Symphony and Main Incubator co-hosted a developer meetup in Frankfurt for the first time in May! Frankfurt-based developers learned about Symphony's developer program, how to automate RPA using a combination of Symphony and BPM toolkits, and how to leverage NLP services to build intelligent bots. Take a look at the **presentation deck** to learn more.

Make sure to join one of our **Symphony Developer Meetup Groups** to receive updates on upcoming developer events near you!

Share with a Colleague

Know a colleague that will find the developer newsletter useful? Help them **subscribe to the newsletter** now.

The Developer Documentation, the instructions provided in this Symphony Developer Newsletter (collectively, the "Symphony 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 Symphony materials; (II) Symphony is under no obligation to provide support or maintenance for the Symphony materials; and (III) Symphony disclaims all liability for or with respect to your or your firm's access to or use of the Symphony 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 Symphony Materials are subject to change without notice and are for information and illustrative purposes only. None of the Symphony 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 Symphony Materials are provided with the understanding that with respect to the Symphony 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.