

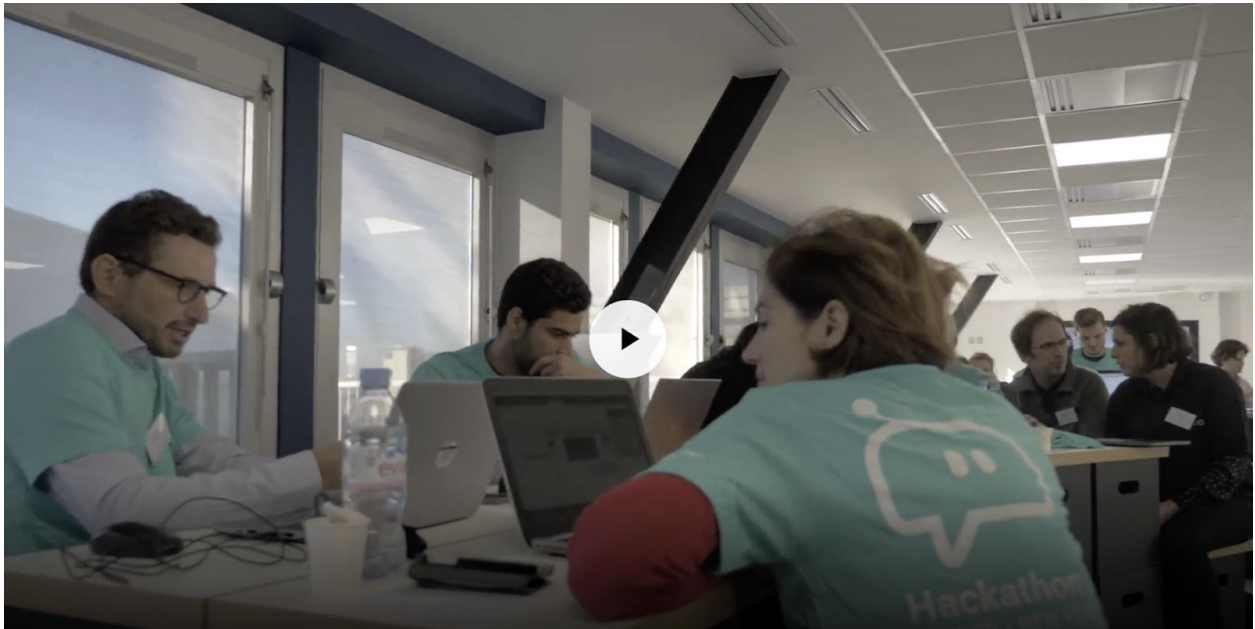
SYMPHONY

DEVELOPER NEWSLETTER

This newsletter covers winning teams from the Symphony Innovate Hackathons, workflow-driven chat bots and the importance of state management, updates to REST API reference documentation, and upcoming developer events.

Symphony Innovate Hackathon Winning Teams

On September 27, over 100 developers participated in our Symphony Innovate Hackathons in London, New York, and Paris. It was thrilling to see how much could be built in just one day!



Watch the Symphony Paris Hackathon video to learn more

The winning developments for each category were:

Most Cutting-Edge Technical Development

1. The Collaborator (BNP Paribas, London)
2. Elnek Killer Bot (BNP Paribas, Paris)
3. Watchover Bot (Barclays, New York)

Most Impactful Business Automation Development

1. Rachel Client Onboarding Bot (Credit Suisse, London)
2. Smart RFQ Bot (CA-CIB/Invivo, Paris)
3. SCORE (BNP Paribas, New York)

Machine Learning, Natural Language Processing (NLP) and AI technologies were leveraged in the winning developments. Check out our [hackathon recap blog](#) for more details on how the teams leverage these technologies in their projects!

Workflow-Driven Chat Bots and State Management

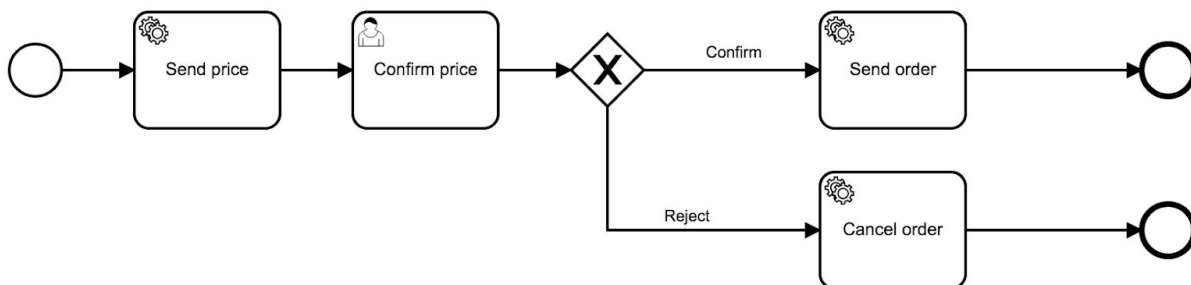
Digitizing workflows starts with a process that is well-defined and repeatable. Technologies can then help automate. The Symphony Generator includes an example template to help automate a workflow:

1. A user would like to know the price of a product
2. Based on the price, the user will action a decision

This is a simple workflow example. However, users may put in multiple requests simultaneously into a chat room, hence it's crucial to track the state of each request independently. Workflow Engines allow us to track the state of each request and define processes using a right drag and drop UI.

To build this process, we've chosen [Camunda](#), an open source workflow and decision engine.

We create the process mentioned above using a modeler:



There are a number of tasks and decisions required in the process above. The variable [StreamID](#) tracks instances and is the unique identifier of a process. The decisions made by the users are tied to a [MessageID](#). The user information can be used in conjunction with either one of these data points to control the instances of processes even further.

Please review an example of [a simple human-bot use case](#) and [generate your own java-based bot using the Camunda engine](#)! You can reach out to our [Platform Solutions](#) team for any implementation questions or best practices.

Updates to REST API Reference Documentation

We have made substantial updates on our REST API reference documentation to make it simple and more readable. Check out [our new reference documentation](#) now!

The responses and sample cURLs are now displayed on the right panel. Additionally, you can toggle between versions of Symphony to see older versions of documentation.

DEVELOPER NEWSLETTER

The screenshot shows the documentation for the **Message Status** API endpoint. It includes a sidebar with navigation links, a main content area with a description and parameters, and a dark-themed code block showing a cURL request and its JSON response.

Message Status
Released in 1.47. Get the status of a particular message: 'sent', 'delivered', and 'read'.

PATH PARAMS

- messageId*** string: Message ID as a URLSafe Base64 encoded string. See [Message ID](#) for full reference.

HEADERS

- sessionToken*** string: Session token obtained from [Session Authenticate](#) endpoint.

The response indicates the status of the message for internal and external users:

- sent**: All users to whom the message has been sent and received by the Symphony system, but not yet delivered to any user's Symphony client.
- delivered**: All users who have at least one Symphony client to which the message has been delivered, and not read yet.
- read**: All users who have read that message, in any Symphony client.

Note:

- For security reasons, the response excludes `passwords` for external users.

cURL

```
curl https://acme.symphony.com/pod/v1/message/_Gj13WjR_SlRvTmPNl6Fn__qWYCYL_dV/status -H "sessionToken: SESSION_TOKEN"
```

200 OK **400 Bad Request** **401 Unauthorized** **403 Forbidden** **500 Internal Server Error**

```
{
  "author": {
    "userId": "7078106103901",
    "firstName": "Gustav",
    "lastName": "Mahler",
    "displayName": "Gustav Mahler",
    "email": "gustav.mahler@music.org",
    "userName": "gmahler",
    "timestamp": "1531968487845"
  },
  "read": [
    {
      "userId": "7078106103901",
      "firstName": "Gustav",
      "lastName": "Mahler",
      "displayName": "Gustav Mahler",
      "email": "gustav.mahler@music.org",
      "userName": "gmahler",
      "timestamp": "1489769156271"
    },
    {
      "userId": "7078106103902",
      "firstName": "Hildegard",
      "lastName": "Bingen"
    }
  ]
}
```

We appreciate any feedback on our APIs and documentation. Please feel free to provide feedback to our [Platform](#) team.

Developer Events

Register to attend one of our upcoming developer meetups in [Singapore on November 1](#) and [Hong Kong on November 6](#) at [Amazon Web Services](#)! At these meetups, attendees will learn about the Symphony Developer Program, building intelligent bots, and Machine Learning 101 from Amazon. We will also do a hands-on session building a bot using the [Symphony Generator](#) and [Amazon Comprehend](#)! If you plan to participate during the hands-on session, please bring your own device.

If you haven't already, join one of our [Symphony Developer Meetup Groups](#) to receive updates on future 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 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.