

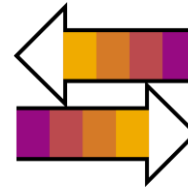


Week 4: Further Topics

Unit 4a: Latest Updates

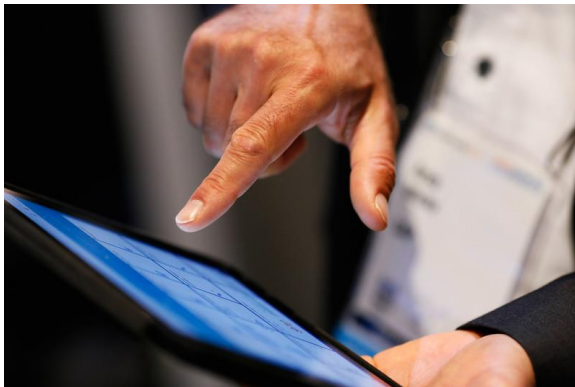
API triggers and notifiers

Triggers provide the means of starting a bot execution from different channels

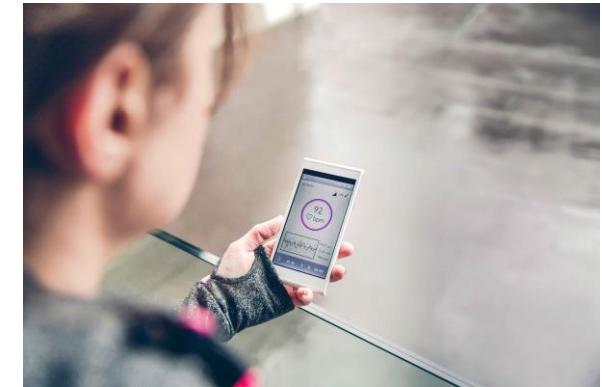


Notifiers provide the means of communicating the bot execution state through different channels

API triggers and notifiers provide the connection to SAP Intelligent RPA via Web API channel



Start RPA bot from your application



Receive notifications to your application

API triggers and notifiers – API trigger

- Are attached to a particular scenario in a deployment
 - Can be added, modified, or removed at any time
- Provide a dedicated API endpoint protected by API key
- Provides input for a bot execution
 - Defined by scenario input schema
- Can provide invocation context for call correlation
 - Invocation context is available in notifiers

Edit API trigger

Activation

☒

*Name

Trigger PO creation

Description

e.g. called from service X

*Execute

scPurchaseOrder

Trigger owner

Assign to me

② Use credentials of

☒ Trigger owner

☐ API trigger caller

URL

https://

HTTP Method

POST

X-Token header

1oQ4Oda-hNaGLWIXJ-PEkadMkOLqqa80

Generate new token

② Payload

Example

Input Schema

```
{  "invocationContext": "${invocation_context}",  "input": {    "ProductName": "",    "EquipmentNumber": ""  }}
```

Save

OK

API triggers and notifiers – API notifier

- Are attached to a trigger
 - Can be added, modified, or removed at any time
- React on an activation event
 - Start/Success/Failure/Cancel
 - Different variables are available for different events
- Contain a REST call configuration
 - Callback URL
 - HTTP method
 - Authorization
 - Headers
 - Payload
 - Custom JSON structure
 - Use `${invocation_context}` sent with API trigger
 - Use other variables available for the notifier event

The screenshot shows the 'Create API Notifier' form. It includes fields for 'Notify on' (set to 'Success'), 'Name' (set to 'New Notifier'), 'Description' (set to 'e.g. notified via API X'), 'Callback URL' (set to 'https://call.my.service'), 'HTTP Method' (set to 'POST'), and 'Authorization' (set to 'Basic'). There are also fields for 'Username' (set to 'user') and 'Password' (masked with dots). A 'Headers' section with an 'Add Header' button is present. The 'Payload' section shows a JSON structure with 'invocationContext' and 'output' fields. The form has 'Create' and 'Cancel' buttons at the bottom right.

Create API Notifier

*Notify on: Success

Available Variables: `${invocation_context}`
`${output}`

*Name: New Notifier

Description: e.g. notified via API X

Callback URL: https://call.my.service

HTTP Method: POST

Authorization: Basic

*Username: user

*Password:

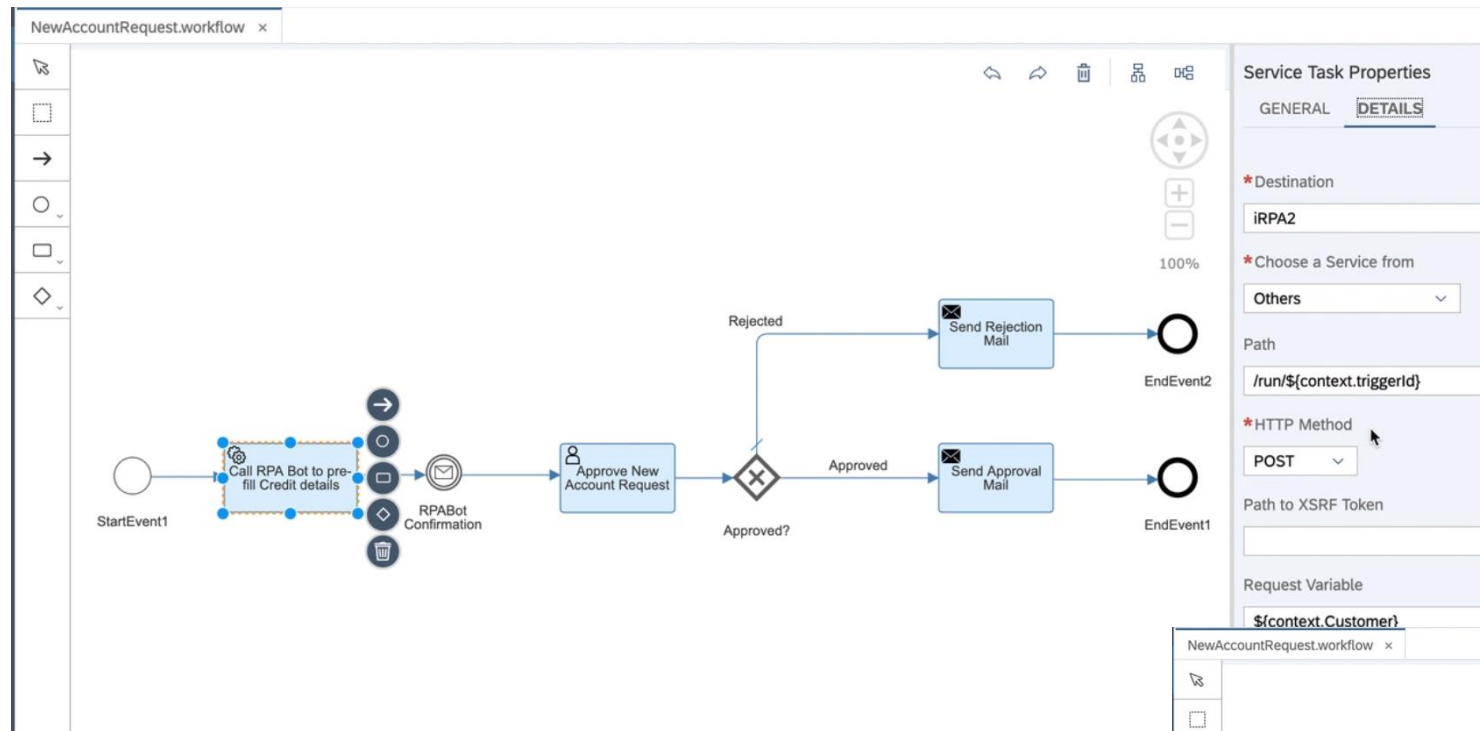
Headers: Add Header

Payload:

```
1 {  
2   "invocationContext": "${invocation_context}"  
3   "output": "${output}"  
4 }
```

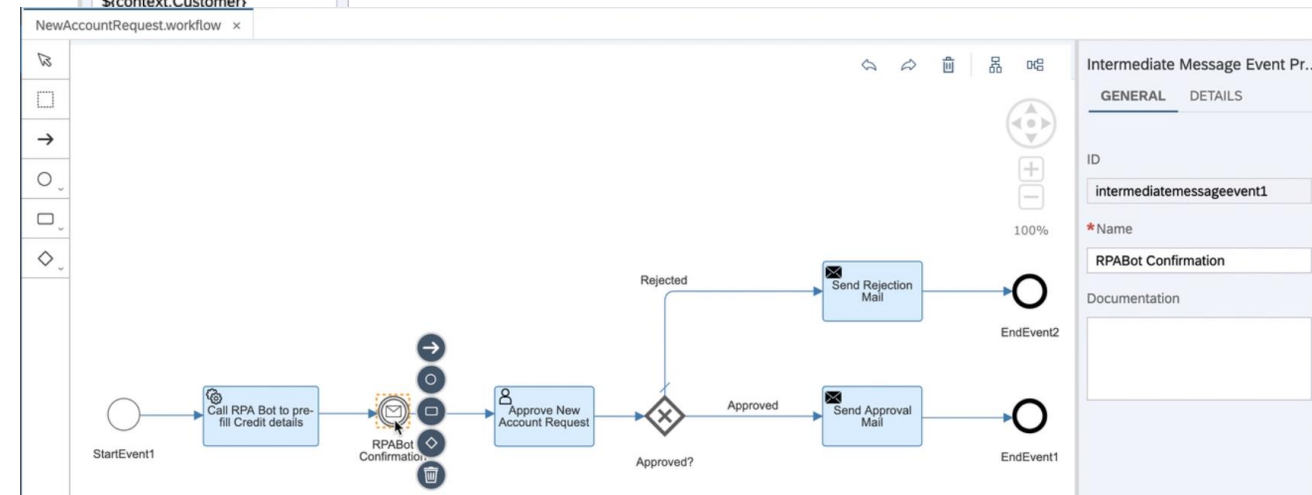
Create Cancel

API triggers and notifiers – BPM integration: Workflow definition



← Call API trigger to start the bot

Wait for bot execution →



API triggers and notifiers – BPM integration

1. Define an API trigger to open an endpoint
2. Define an API notifier with the required BPM configuration
 - Callback URL of your workflow service
 - Payload structure expected by BPM references invocation context
 - Authentication

Create API Notifier

*Notify on Available Variables `${invocation_context}`
`${output}`

*Name

Description

Callback URL

HTTP Method

Authorization

*Auth URL

*Client ID

*Client secret

*Grant type

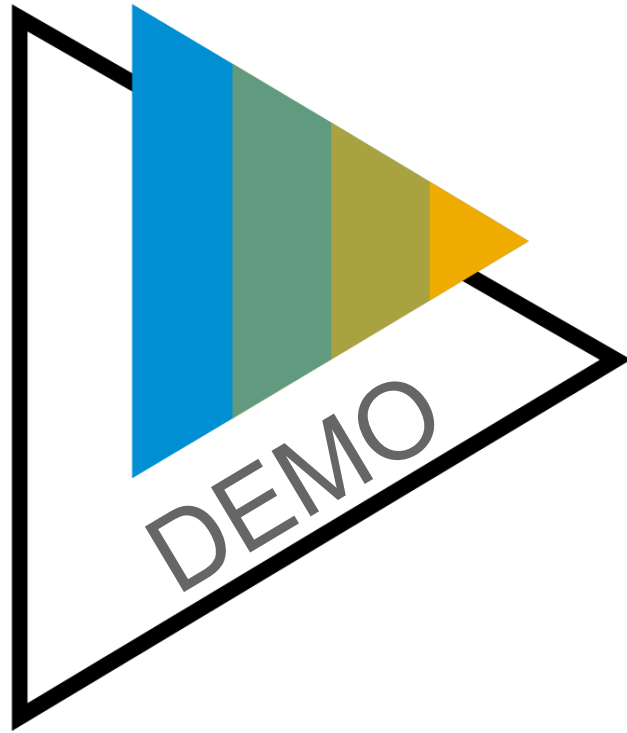
Headers [Add Header](#)

Payload

```
1 {
2   "context": {
3     "status": "SUCCESS",
4     "output": "${output}"
5   },
6   "definitionId": "${invocation_context.messageDefId}",
7   "workflowDefinitionId": "${invocation_context.workflowDefId}",
8   "businessKey": "${invocation_context.key}"
9 }
10 |
```

[Create](#) [Cancel](#)

Demo API trigger and notifier definition



Integration with SAP Conversational AI

Interact

Conversational AI (CAI)

Chatbots to interface and hand-over to execution bot



Interfacing



Execute

Intelligent RPA

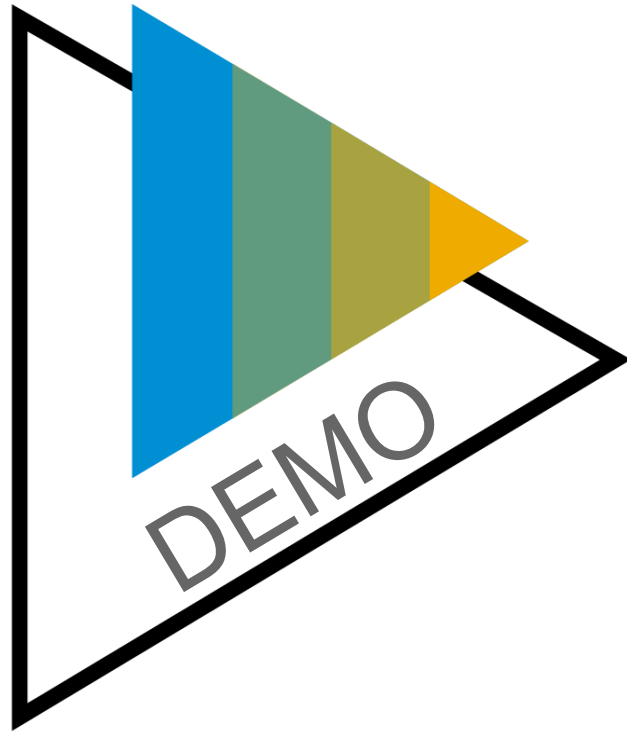
Multiple bot workflows for execution (attended + unattended)



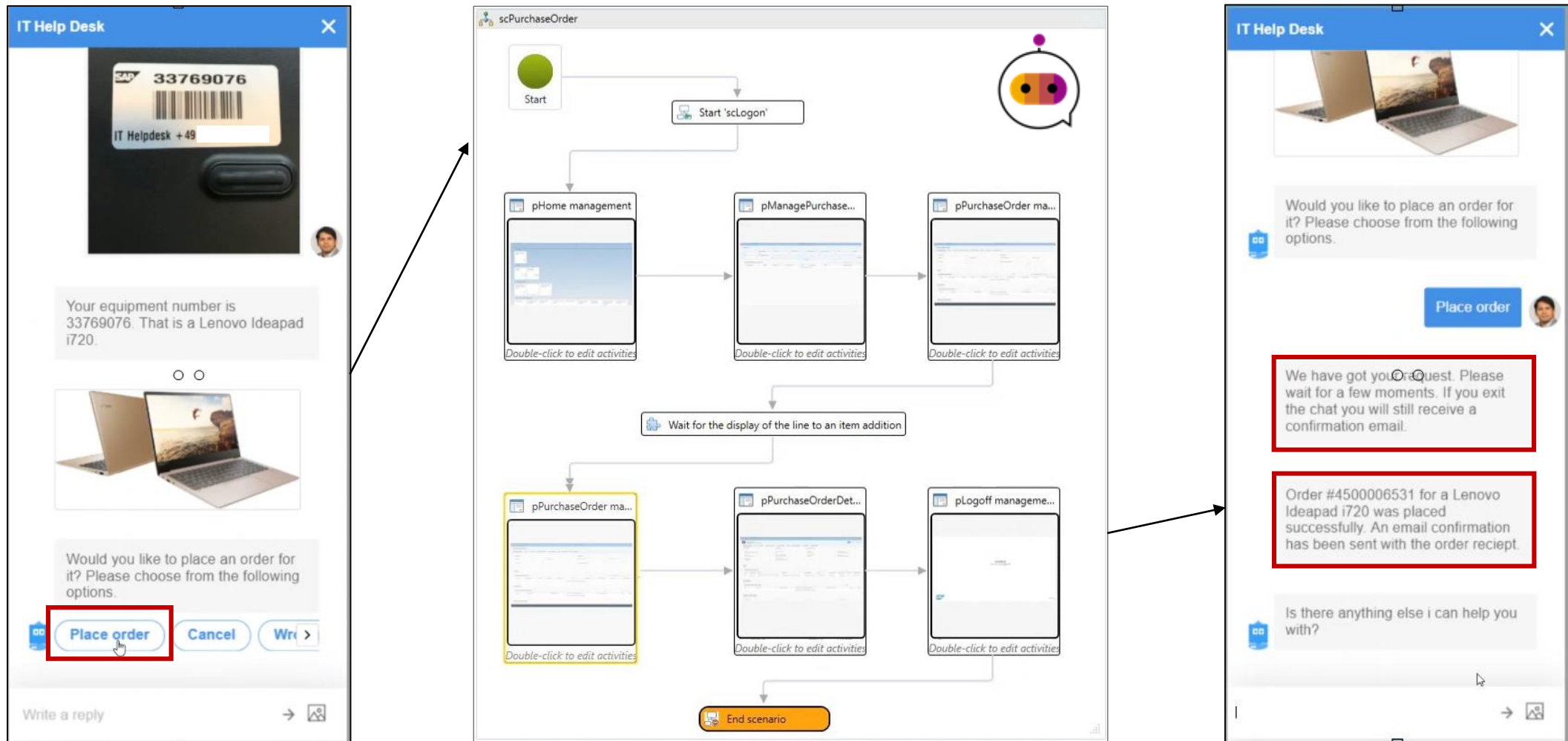
Performing Tasks

Latest Updates

Demo ChatBot in action



Conversational AI integration: Trigger an Intelligent RPA bot and receive feedback



Conversational AI integration: API trigger – Open an Intelligent RPA bot endpoint

Create API trigger

Activation ☒

*Name

Description

*Execute

Trigger owner

Use credentials of ☒ Trigger owner ☐ API trigger caller

Example Input Schema

Payload

```
{
  "invocationContext": "${invocation_context}",
  "input": {
    "ProductName": "",
    "EquipmentNumber": ""
  }
}
```

Show required fields only ☐

Create Cancel

POST

SAVE

Authentication Headers Body

HOW DOES THIS WORK?

Use a template

OR

Use the default body

```
{
  "invocationContext": {
    "conversationId": "{{conversation_id}}"
  },
  "input": {
    "EquipmentNumber": "{{memory.EquipmentNumber}}",
    "ProductName": "{{memory.ProductName}}"
  }
}
```

SAVE AS A NEW TEMPLATE

- Collect data from chat bot and pass it as RPA bot input
 - ProductName, EquipmentNumber
- Invocation context provides information for conversation correlation

Conversational AI integration: Conversational AI notifier

- Choose a notifier activation event
 - Success/Start/Failure/Cancel
- Configure notifier
 - Conversation ID
 - Chatbot authorization token
- Select an action to be performed
 - Send a text message to conversation
 - Modify conversation state
- Configure the action parameters
 - Message to be sent
 - Memory and language to be updated

The image shows two side-by-side screenshots. The left screenshot is the 'Create CAI notifier' configuration form, and the right screenshot is a chatbot conversation window titled 'IT Help Desk'.

Create CAI notifier form:

- *Notify on:** Start (dropdown menu)
- Available Variables:** \${invocation_context}
- *Name:** Confirm bot execution start
- Description:** Notify user on start execution
- *Conversation ID:** \${invocation_context.conversationId}
- *Token:** [Redacted]
- *What do you want to do?:** Send a message (dropdown menu)
- Message type:** Text
- Delay in seconds:** 0
- *Message content:** We have got your request. Please wait for a few moments. If you exit the chat you will receive a confirmation email.

IT Help Desk chatbot window:

- Header: IT Help Desk
- Image: Two laptops
- Message: Would you like to place an order for it? Please choose from the following options.
- Button: Place order
- Message: We have got your request. Please wait for a few moments. If you exit the chat you will still receive a confirmation email. (This message is highlighted with a red box and an arrow pointing from the configuration form's message content field.)
- Message: Order #4500006531 for a Lenovo Ideapad i720 was placed successfully. An email confirmation has been sent with the order receipt.
- Message: Is there anything else i can help you with?

Conversational AI integration: CAI notifier – Success notifier

Create CAI notifier

*Notify on

Available Variables `${invocation_context}`
`${output}`

*Name

Description

*Conversation ID

*Token

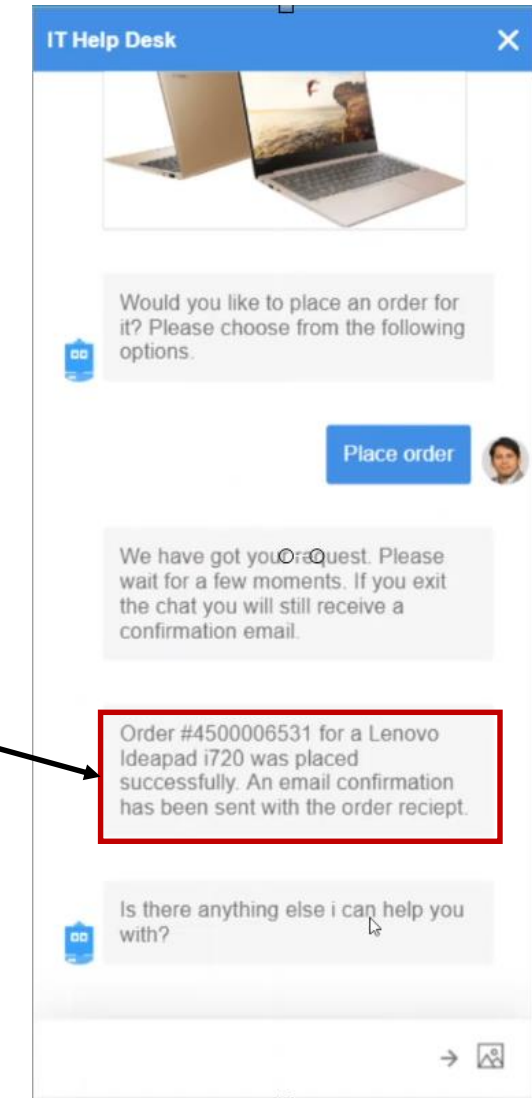
*What do you want to do?

Message type

Delay in seconds

*Message content

Create Cancel



Thank you.

Contact information:

open@sap.com

Follow all of SAP



www.sap.com/contactsap

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See www.sap.com/copyright for additional trademark information and notices.