

PUBLIC (PUBLIQUE)

2024-04-04

Correction des erreurs

Contenu

1	Dépannage et assistance.	3
1.1	Desktop Agent.	3
	Activer le mode de support.	4
	Exporter les mesures relatives au fonctionnement.	5
	Messages d'erreur Desktop Agent.	6
	Desktop Agent Systray vide.	8
	Utilisation du PC pendant l'exécution d'un bot.	9
	Redémarrage du système en mode non surveillé.	10
	Limite d'affectation de mémoire atteinte.	11
1.2	Desktop Studio.	11
	Meilleures pratiques pour le verrouillage d'écran et l'infrastructure de bureau virtuel déconnectée.	12
	Meilleures pratiques pour la bibliothèque PDF.	20
	Meilleures pratiques pour les appels de service Web.	21
	Meilleures pratiques pour le traitement d'une quantité importante de données lors de l'exécution d'un scénario.	31
	Meilleures pratiques pour les captchas.	31
	Meilleures pratiques pour l'automatisation des écrans SAP GUI avec des hiérarchies horizontales.	32
	Meilleures pratiques pour la gestion des fichiers.	34
	Meilleures pratiques pour l'extension Chrome.	36
	Meilleures pratiques pour la gestion de la fenêtre temporaire Serveur occupé.	39
	Événements asynchrones dans les boucles For.	45
1.3	Cloud Factory.	49
	Périmètre insuffisant.	49
1.4	Cloud Studio.	52
	Mappage des fonctionnalités de Cloud Studio et Desktop Studio.	52
	Définition du contrôle Pour chaque pour des éléments sans rectangle englobant.	53
	Déverrouillage d'une session verrouillée pendant l'exécution de l'automatisation.	55
	Application d'IU en cours d'exécution avec des privilèges plus importants.	56
	Lancement du SAP Logon en mode non surveillé.	56
	Meilleures pratiques pour l'optimisation d'un élément.	63
	Fonctionnement des tests.	65
	Exécution d'une automatisation à partir d'une autre automatisation.	66

1 Dépannage et assistance


Bienvenue dans la rubrique Dépannage et assistance. Cette rubrique répertorie plusieurs notes de la base de connaissances disponibles à l'adresse <https://support.sap.com/> ou [directement en définissant le composant](#).

Si vous rencontrez des problèmes lors de l'utilisation de SAP Intelligent RPA, nous vous suggérons de déclencher une demande via le SAP Support Portal. Pour en savoir plus sur la création d'un incident, voir [Obtenir de l'aide](#).

Par composant

- [Notes liées à Desktop Agent \[page 3\]](#)
- [Notes liées à Desktop Studio \[page 11\]](#)
- [Notes liées à Cloud Factory \[page 49\]](#)


Problèmes inter-composants et autres problèmes

- Pourquoi ne puis-je pas voir le produit SAP Intelligent RPA sur SAP Marketplace ? [2824501](#)
- Contrôles de configuration des navigateurs Web - [2796986](#)
- Où puis-je trouver du contenu prédéfini pour S/4HANA ? - Le contenu est disponible sur rapid.sap.com. Consultez la note [2788986](#)  pour la stratégie de lancement.

Installation et configuration

- [Conditions préalables techniques et configuration requise](#)
- [Configuration de Factory for SAP Intelligent Robotic Process Automation](#)
- [Configuration des composantes On-Premise](#)
- [Mise à jour de votre infrastructure SAP Intelligent RPA 1.0 \(Desktop Agent et Studio\)](#)
- [Configuration des navigateurs Internet](#)

1.1 Desktop Agent

- Voici comment éviter les problèmes de connexion : [3155610](#) 
- Erreur "Agent XXX déjà connecté" lors de la tentative de connexion à Desktop Agent - [2796483](#)
- Aucune action lors du lancement d'un scénario non surveillé avec Desktop Agent pour SAP Intelligent RPA - [2796491](#)
- Aucune icône Desktop Agent dans la zone des icônes système Windows [2796377](#)
- Lors du téléchargement d'un Package à partir de Factory, erreur "CCtxtRun2DIg::OnGoToSocket : échec de SetIpaSocket" - [2798996](#)
- Aucune clé de registre ou valeur erronée & Aucun fichier JSON trouvé - [2796396](#)
- Problème lors du redémarrage de l'Agent (nouvelle session Windows, changement de projet ou nouveau projet depuis Factory). Pour permettre le redémarrage de l'Agent, le service "SAP Intelligent RPA Service" (CxAgent.exe) doit être opérationnel.

- Examen nécessaire de la connexion Agent/Factory : téléchargez l'historique des événements d'un agent dans Factory pour SAP Intelligent RPA. Pour en savoir plus, voir [Synthèse des agents](#).
- Comment résoudre les problèmes de connexion de l'agent survenus lors de la mise à jour de Cloud Factory : [2981018](#)
- Comment masquer les informations de machine et de connexion : [3056143](#)

1.1.1 Activer le mode de support

Desktop Agent 3 dispose d'un mode de support. Vous pouvez utiliser le mode de support pour enregistrer des traces ou installer une version spécifique de l'agent.

Prérequis

- Vous devez avoir le rôle IRPAOfficer.
- Si vous voulez installer une version spécifique de l'agent après avoir activé le mode de support, l'agent doit être en mode surveillé. Pour en savoir plus, voir [Modification du mode de l'agent](#).

Procédure

1. Dans SAP Intelligent RPA, sélectionnez l'onglet [Agents](#).
2. Recherchez les agents pertinents et sélectionnez **•••** [Autres actions](#) et [Activer le support](#).
3. Sélectionnez une durée comprise entre 1 heure et 5 jours et sélectionnez [Activer](#).
4. Le cas échéant, vous pouvez configurer davantage les options du traceur, y compris les composants du traceur dans l'agent sous [Options](#) et [Traceur](#).

Résultats

- Le mode de support est activé pour la durée indiquée.
Vous pouvez [Désactiver le mode de support](#) manuellement en suivant l'étape 2.
- Les mises à jour automatiques sont activées.
Vous pouvez désactiver les mises à jour automatiques dans Desktop Agent en sélectionnant [SystèmeOptions](#) et en décochant [Activer les mises à jour automatiques](#).

ⓘ Remarque

Lorsque le mode de support est désactivé, l'option [Activer les mises à jour automatiques](#) n'est pas visible.

- Les traces fonctionnelles sont enregistrées par tâche, les traces techniques sont enregistrées au démarrage et les traces sont chiffrées et chargées dans le locataire.

Dans Desktop Agent, vous pouvez accéder aux options du *Traceur* sous *Options* lorsque le mode de support est activé.

1.1.2 Exporter les mesures relatives au fonctionnement

Exportez des mesures relatives au fonctionnement pour votre agent.

Prérequis

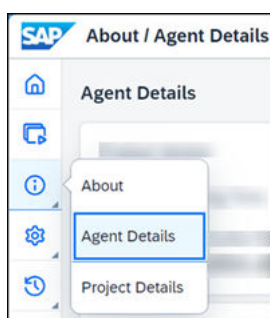
Vous devez avoir activé le mode de support pour pouvoir exporter des mesures relatives au fonctionnement. Pour en savoir plus, voir [Activer le mode de support \[page 4\]](#).

Contexte

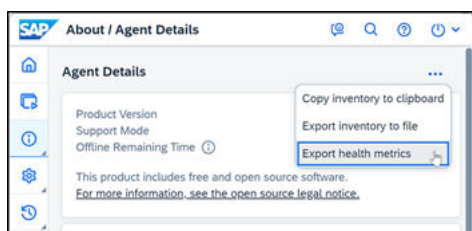
Vous pouvez exporter des mesures relatives à votre agent et les envoyer à l'équipe de support SAP afin qu'elle les examine. Une fois le mode de support activé, les mesures relatives au fonctionnement sont stockées pendant un maximum de 7 jours.

Procédure

1. Dans Desktop Agent, sélectionnez ⓘ *À propos* et *Détails de l'agent*.



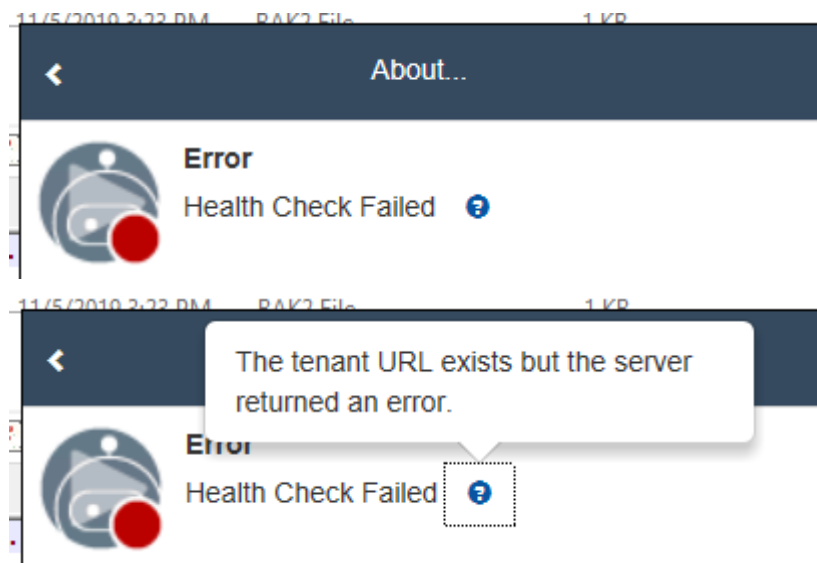
2. Sélectionnez ⋮ *Plus* et *Exporter les mesures relatives au fonctionnement*.



Résultats

Un fichier zip contenant des fichiers CSV pour les différentes mesures est exporté sur votre PC. Vous pouvez envoyer ce fichier au support SAP.

1.1.3 Messages d'erreur Desktop Agent



Au sein du systray de Desktop Agent, le statut, l'erreur et l'avertissement sont signalés dans l'écran À propos. L'icône d'avertissement (orange) ne nécessite aucune interaction humaine car l'agent fera un autre essai. L'icône d'erreur (rouge) requiert souvent une interaction humaine.

Veuillez trouver ci-dessous des erreurs courantes qui peuvent se produire :

- Échec du contrôle de fonctionnement (HEALTHCHECK_FAILURE) - L'URL du locataire existe, mais une erreur a été renvoyée par le serveur.
 - Cause : l'URL existe, mais l'agent n'est pas en mesure de se connecter.
 - Solution :

- Vérifiez l'URL via un navigateur Internet.
- Contactez votre administrateur pour obtenir l'URL correct.
- Contrôlez la connectivité.

Si vous utilisez un réseau privé virtuel (VPN), ce problème peut provenir du fait que votre VPN n'autorise pas les WebSockets. Pour en savoir plus, voir la section **À propos du protocole WebSockets** en bas de cette page : [Conditions préalables techniques et configuration requise](#).

- L'URL du locataire est inaccessible (RETRY_HEALTHCHECK_ON_FAILURE).
 - Cause : l'URL n'est plus disponible.
 - Solution :
 - La connectivité est interrompue : relancez l'interface réseau, vérifiez l'IP et le DNS.
 - URL incorrect : contactez votre administrateur pour obtenir l'URL correct.
- Authentification non terminée (AUTHENTICATION_NOT_COMPLETED).

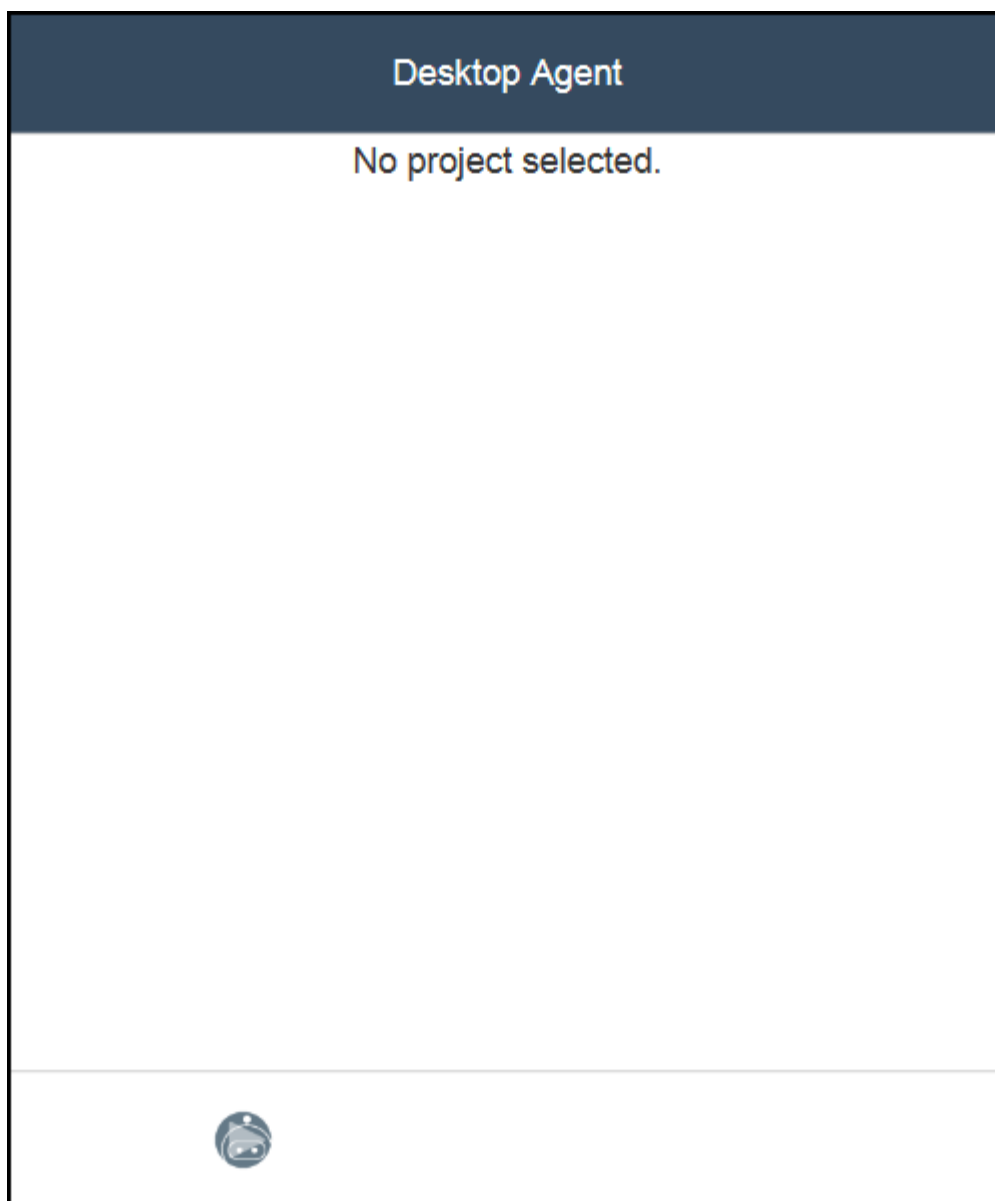
- Cause : l'authentification n'a pas été complètement terminée.
- Solution : relancez l'agent afin de lancer à nouveau l'authentification.
- La connexion a été perdue (NEED_SENDING_CLIENT_INFO).
 - Cause : débranchement d'un câble, déconnexion du Wi-Fi...
 - Solution : vérifiez votre connexion. Lorsqu'elle sera à nouveau opérationnelle, l'agent se reconnectera automatiquement.
- Agent introuvable dans le groupe d'agents (INVALID_SEND_CLIENT_INFO_ANSWER).
 - Cause : votre agent n'est pas déclaré dans Cloud Factory.
 - Solution : contactez l'administrateur pour ajouter votre agent dans un groupe d'agents (Agents > Groupes d'agents).
Pour en savoir plus, voir [Création d'un groupe d'agents](#).
- La connexion est fermée définitivement.
 - Cause : erreur de communication entre Desktop Agent et Factory
 - Solution : Desktop Agent doit être mis à jour. La version minimale requise pour résoudre ce problème est la version 2011 2.0.6. Pour en savoir plus, voir la note SAP [2997097](#).
- Package introuvable dans le systray
 - Cause : votre package n'apparaît pas dans le systray.
 - Solution : accédez à [Scripts](#) et recherchez : `GLOBAL.events.START.on(function (ev) {`. Si vous voyez un extrait de code comme celui affiché ci-dessous, commentez la condition "if" :

Exemple de code

```
GLOBAL.events.START.on(function (ev) {
  if(ctx.options.isDebugEnabled) {
    //Add item in systray menu.
    ....
  }
});
```

- En attente de la saisie des identifiants par l'utilisateur
 - Cause : Desktop Agent attend que l'utilisateur saisisse ses identifiants
 - Solution : Arrêtez Desktop Agent et relancez-le. Il va vous être demandé de vous connecter. Si vous rencontrez encore des problèmes :
 1. Ouvrez l'URL du locataire dans Internet Explorer, connectez-vous et sauvegardez l'ID utilisateur ainsi que le mot de passe.
 2. Sélectionnez *Outils > Options Internet > Sécurité*.
 3. Cliquez sur *Internet > Niveau personnalisé...*
 4. Sous *Options > Scripting*, définissez *Scripting actif* sur *Activer*.
 5. Revenez à *Options Internet > Sécurité*. Cliquez sur *Sites de confiance* puis *sur Sites*.
 6. Saisissez l'URL du locataire et cliquez sur *Ajouter* pour l'inclure dans la liste des sites de confiance. Vous devriez maintenant être en mesure de visualiser la fenêtre temporaire indiquant que Desktop Agent est en cours de journalisation en utilisant les identifiants stockés. Il peut maintenant être mis en ligne.
- Échec de l'enregistrement
 - Cause : échec de l'enregistrement
 - Solution : assurez-vous d'utiliser l'URL correct en vérifiant les étapes d'enregistrement du locataire. Pour en savoir plus, voir [Enregistrement du locataire Desktop Agent](#).

1.1.4 Desktop Agent Systray vide



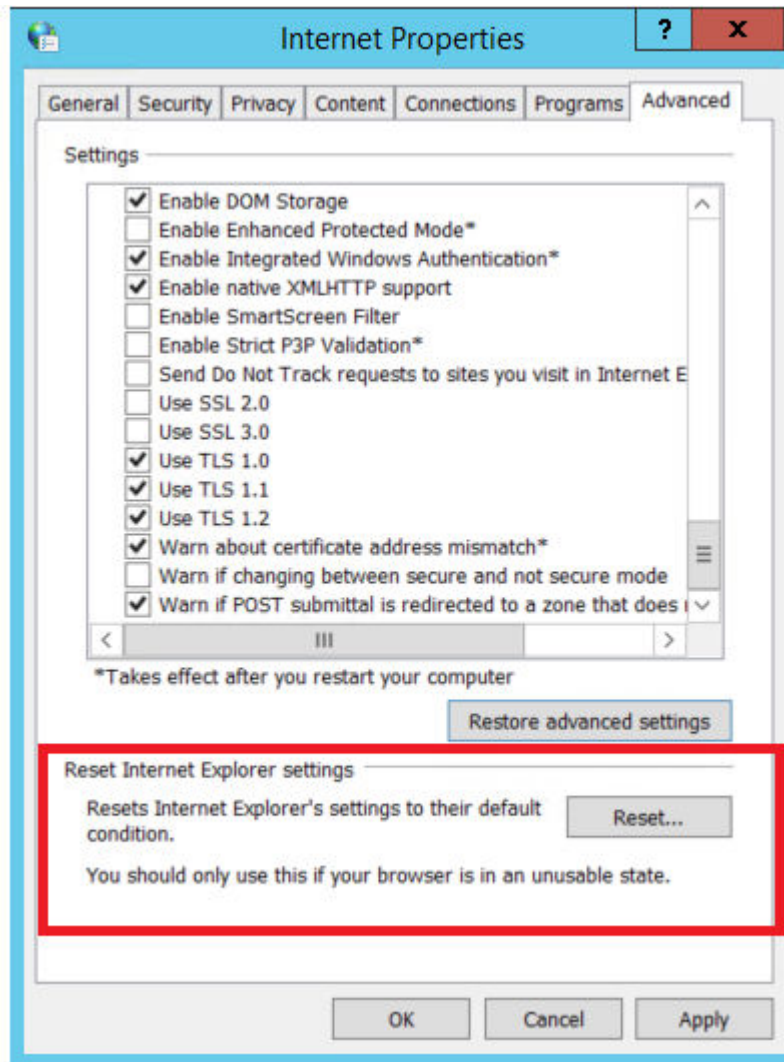
Vous avez vérifié toutes les conditions techniques requises et configuré votre navigateur, mais l'agent Systray est vide.

Cause

Internet Explorer est instable.

Résolution

1. Réinitialiser les paramètres d'Internet Explorer.



2. Configurez à nouveau les paramètres d'Internet Explorer. Pour plus de détails, voir [Configuration des navigateurs Internet](#).

1.1.5 Utilisation du PC pendant l'exécution d'un bot

Il se peut que vous souhaitiez utiliser votre PC pendant l'exécution d'un bot.

ⓘ Remarque

Si votre bot ne nécessite pas d'intervention humaine, vous ne devez pas utiliser votre PC pendant son exécution.

L'exécution d'un bot en arrière-plan est possible, mais vous devez prendre en compte les informations suivantes concernant la conception de votre bot :

- Il ne doit pas contenir d'**activités de clic de souris**, d'**activités de double-clic** et d'autres fonctions nécessitant des interactions avec la souris (l'**activité de clic normale** fonctionne).

- Il ne doit pas contenir d'**activités de frappe de clavier**.
- Vous pouvez utiliser votre PC si votre flux d'opérations ne contient aucune activité nécessitant l'ouverture d'une application ou d'une fenêtre.
- Vous pouvez réduire l'application pour l'automatiser en fonction de sa technologie.
- Si vous interagissez avec la même application que le bot, il existe un risque important de conflits (même si l'application est réduite).

ⓘ Remarque

Si vous souhaitez utiliser votre PC pendant l'exécution d'un bot, il est préférable d'effectuer des tests. En effet, le comportement du bot dépend largement du cas d'utilisation.

1.1.6 Redémarrage du système en mode non surveillé

Le système peut redémarrer lorsque Desktop Agent est en mode non surveillé.

Contexte

Lorsque Desktop Agent est exécuté sur un système en mode non surveillé, il se peut que le système redémarre. Vous pouvez vouloir lancer automatiquement l'agent.

Comment lancer Desktop Agent immédiatement après le redémarrage du système

Pour lancer Desktop Agent immédiatement après le redémarrage, vous devez avoir stocké vos informations d'identification Windows dans Desktop Agent. Vous devez également avoir coché la case [Lancement automatique au démarrage de Windows](#) dans le Systray de Desktop Agent. Pour cocher cette option, cliquez sur le bouton [Autres actions](#), puis sur [Options](#).

La session s'ouvre en mode non interactif lorsque le système a redémarré. Ensuite, l'agent est lancé automatiquement.

Pour en savoir plus sur le stockage des informations d'identification Windows pour lancer Desktop Agent pour SAP Intelligent RPA en mode non surveillé, voir [Définition du mot de passe Windows dans Desktop Agent](#).

Comment supprimer les identifiants

Les identifiants peuvent être supprimés à l'aide de la commande suivante :

```
CxStoreCred.exe /forget
```

Avec cette commande, Desktop Agent arrête le lancement automatique après le redémarrage.

Pour en savoir plus sur le lancement de Desktop Agent lorsque le serveur redémarre, consultez la note SAP suivante : [3010368](#).

1.1.7 Limite d'affectation de mémoire atteinte

Les erreurs de mémoire insuffisante se produisent lorsque trop de données sont traitées en une seule fois ou lorsqu'une automatisation ne libère pas la mémoire après utilisation. Cela peut entraîner l'échec de l'automatisation. Suivez les recommandations pour éviter d'atteindre la limite d'allocation de mémoire.

Il est recommandé d'effectuer ce qui suit :

- Suivez [Meilleures pratiques pour les scripts personnalisés](#).
- Traitez ou extrayez des ensembles de données volumineux dans des sous-ensembles plus petits pour les traiter par lot. Ne chargez pas l'ensemble des données en une seule fois. Par exemple, au lieu de lire tout un fichier Excel en une seule fois, vous pouvez le boucler et traiter X lignes par X lignes.
- Libérez les données inutilisées dans la mémoire. Par exemple, une variable contenant une liste de centaines d'objets peut être supprimée lorsqu'elle n'est plus utilisée.
- Vérifiez régulièrement l'utilisation de la mémoire pour identifier une utilisation anormale. Sur un ordinateur Windows, vous pouvez vérifier l'utilisation de la mémoire à l'aide de l'application Gestionnaire des tâches.

Si l'erreur de mémoire insuffisante se produit encore après avoir suivi les recommandations, essayez d'exécuter la tâche qui nécessite un chargement de mémoire important à l'aide d'un outil externe, tel qu'un script Python.

1.2 Desktop Studio

Les notes de base de connaissances suivantes se rapportant à Desktop Studio sont disponibles :

- Temps d'attente long lors de la capture d'une page via l'automatisation de l'interface utilisateur - [2788781](#)
- Impossible de créer le composant du gestionnaire de débogage de processus - [2781601](#)
- Erreur "System.InvalidOperationException" lors du débogage d'un projet dans Desktop Studio - [2833621](#)
- Écran noir lors de la capture d'une page à l'aide de Intelligent RPA Desktop Studio avec Chrome - [2842523](#)
- Fichier [...] form.pscm introuvable - [2849305](#)

Astuces pour Desktop Studio

- Comment récupérer une variable (texte) à partir de Factory - [2822776](#) (voir également cette rubrique dans le guide du développeur)
- Comment récupérer une variable (identifiants) à partir de Factory - [2822500](#) (voir également cette rubrique dans le guide du développeur)
- Comment insérer une image dans un e-mail - [2864114](#)
- Comment utiliser un autre outil de différenciation/fusion - [2867241](#)
- Comment renommer un projet - [2956757](#)

Meilleures pratiques

- [Meilleures pratiques pour le verrouillage d'écran et l'infrastructure de bureau virtuel déconnectée \[page 12\]](#)
- [Meilleures pratiques pour la bibliothèque PDF \[page 20\]](#)
- [Meilleures pratiques pour les appels de service Web \[page 21\]](#)
- [Meilleures pratiques pour le traitement d'une quantité importante de données lors de l'exécution d'un scénario \[page 31\]](#)
- [Meilleures pratiques pour les captchas \[page 31\]](#)
- [Meilleures pratiques pour l'automatisation des écrans SAP GUI avec des hiérarchies horizontales \[page 32\]](#)
- [Meilleures pratiques pour la gestion des fichiers \[page 34\]](#)
- [Meilleures pratiques pour la gestion de la fenêtre temporaire Serveur occupé \[page 39\]](#)

1.2.1 Meilleures pratiques pour le verrouillage d'écran et l'infrastructure de bureau virtuel déconnectée

L'outil SAP Intelligent RPA nécessite parfois que l'utilisateur gère des applications avec des actions de bas niveau, comme des clics de souris ou des frappes de clavier. Dans certaines circonstances, ces actions de bas niveau peuvent entraîner des restrictions.

Ce comportement n'est pas limité à une exécution non surveillée, car l'utilisateur peut être amené à lancer des scénarios surveillés sur sa machine et à les laisser fonctionner pendant que l'écran est verrouillé.

Contexte

L'utilisateur confirme qu'une tâche est correctement exécutée lors des phases d'implémentation et de test :

- Avec un débogueur et un outil de test
- En supervisant l'exécution de la tâche en temps réel sur sa machine

Cependant, dès que l'agent est sur une infrastructure de bureau virtuel et que l'écran est verrouillé, la tâche ne s'exécute pas correctement et s'arrête au milieu de l'automatisation pendant certaines actions techniques.

Ce problème survient lorsque :

- L'écran est verrouillé sur la machine de l'utilisateur lors de l'exécution d'une tâche.
- Une session est ouverte sur l'infrastructure du bureau virtuel mais il n'y a aucune activité de la part de l'utilisateur.
- L'infrastructure du bureau virtuel est réduite après une dizaine de minutes.

Motif

Pour traiter des actions d'automatisation spécifiques telles que les clics de souris et les séquences de frappes de clavier, l'utilisateur doit avoir une session ouverte qui empêche le verrouillage de l'écran pendant l'exécution d'une automatisation. Cependant, l'utilisation d'un clavier physique ou virtuel n'est pas nécessaire.

Ce comportement est caractéristique des technologies et applications manipulées dans le cadre de l'automatisation.

Ce type de problème provient principalement des simulations d'utilisation d'une souris et d'un clavier :

- `application.page.item.mouseClick()` ;
- `application.page.item.keystroke("text")` ;

1.2.1.1 Causes

Les rubriques suivantes expliquent les principales causes du verrouillage d'écran et des restrictions de l'infrastructure du bureau virtuel.

1.2.1.1.1 Clic de souris

L'utilisation des actions de clic de souris simulées suivantes entraîne des restrictions :

- `application.page.item.click(true)`
- `application.page.item.clickMouse()`
- Toutes les méthodes `ctx.mouse`

Pour identifier ce type d'action, vérifiez si le curseur se déplace vers la position de l'élément lors de l'appel de l'action de clic.

1.2.1.1.2 Frappes de clavier

L'utilisation des actions de clavier suivantes entraîne des restrictions :

- `application.page.item.keyStroke(e.key.xxx)`
- `application.page.item.keyStroke("Test")`
- `ctx.keyStroke("Test")`

Pour vérifier si l'action peut être exécutée avec un écran verrouillé ou une infrastructure de bureau virtuel déconnectée, vous pouvez envoyer la séquence de frappes de clavier suivante :

```
ctx.wait(function(ev){
  application.page.item.keystroke("text");
},10000);
```

Cette séquence de frappes de clavier va créer un délai pendant lequel vous pourrez verrouiller manuellement l'écran ou déconnecter la session.

1.2.1.1.3 Automatisation IU

Le connecteur UIAutomation vous aide à automatiser toute application de bureau qui implémente l'interface UIAutomation. Si un composant automatisé ne répond pas à une demande d'action de niveau supérieur, le connecteur UIAutomation envoie des clics de souris ou des frappes de clavier. Dans ce cas, les actions simulées telles que `click()` ou `set()` peuvent entraîner des restrictions.

Par exemple, dans Acrobat Adobe Reader, si vous cliquez sur un *menu de fichier* avec la séquence `application.page.item.click()` au lieu de la séquence `application.page.item.clickMouse()`, l'action échouera lorsque l'écran est verrouillé.

Le même comportement peut se produire si vous gérez l'*explorateur de fichiers Windows* avec le connecteur UIAutomation.

Pour identifier ce type d'action :

- Vérifiez si le curseur se déplace vers la position de l'élément lors de l'appel de l'action de clic.
- Combinez cette action à un délai pendant lequel vous pouvez verrouiller manuellement l'écran ou déconnecter la session :

```
ctx.wait(function(ev){
  application.page.item.click();
},10000);
```

1.2.1.1.4 Automatisation de surface avec OCR

L'automatisation de surface repose sur des clics de souris et des frappes de clavier qui nécessitent un bureau avec un écran déverrouillé.

1.2.1.2 Résolutions

Les rubriques suivantes illustrent les meilleures pratiques concernant la résolution des problèmes de verrouillage d'écran et les restrictions de l'infrastructure de bureau virtuel.

[Utiliser l'activité Débloquer \[page 15\]](#)

[Suivre les actions de clic de souris et de frappe de clavier \[page 15\]](#)

[Automatisation des applications Web \[page 16\]](#)

[Maintenir l'écran déverrouillé \[page 19\]](#)

[Maintenir une connexion dans l'infrastructure de bureau virtuel \[page 20\]](#)

1.2.1.2.1 Utiliser l'activité Débloquer

Si votre bot s'exécute en mode non surveillé, vous pouvez utiliser l'activité *Débloquer* pour résoudre les problèmes relatifs aux clics de souris et aux séquences de frappe de clavier.

ⓘ Remarque

Pour la technologie SAP GUI, l'activité *Débloquer* (ou

```
ensureUnlocked
```

fonction) n'est pas requise pour les séquences de frappes.

Vous pouvez ajouter l'activité *Débloquer* dans Desktop Studio ou utiliser la fonction suivante :

```
ctx.workstation.ensureUnlocked();
```

L'activité *Débloquer* débloque une session verrouillée et vous permet d'effectuer des clics de souris ou des séquences de frappe de clavier pendant l'exécution d'une automatisation. Cette activité verrouille à nouveau automatiquement la session 10 secondes après l'action.

Vous devez configurer le mot de passe de la session pour :

- Ouvrir automatiquement la session lorsque votre machine démarre et sans intervention humaine
- Débloquer la session pendant quelques secondes lorsque le bot exécute une automatisation

Vous pouvez directement stocker vos informations d'identification Windows dans Desktop Agent. Pour en savoir plus sur la définition des informations d'identification Windows dans Desktop Agent, voir [Définition du mot de passe Windows dans Desktop Agent](#).

Pour en savoir plus sur le stockage des informations d'identification Windows pour lancer Desktop Agent en mode non surveillé, voir [3010368](#).

1.2.1.2.2 Suivre les actions de clic de souris et de frappe de clavier

Pour résoudre les problèmes relatifs aux actions de clic de souris et de frappe de clavier, vous devez suivre et identifier ces actions. Ces actions peuvent avoir été mises en œuvre dans le projet de manière à éviter un code compliqué ou une gestion inappropriée des éléments par l'outil.

Si vous trouvez l'action de frappe de clavier suivante :

- `ctx.keystroke / application.page.item.keystroke`

Vous pouvez la remplacer par un événement simulé :

- `application.page.item.set()`

Si vous trouvez l'action de clic suivante :

- `ctx.mouse.click / application.page.item.clickMouse / click(true)`

Vous pouvez la remplacer par un événement simulé :

- `application.page.item.click()`

1.2.1.2.3 Automatisation des applications Web

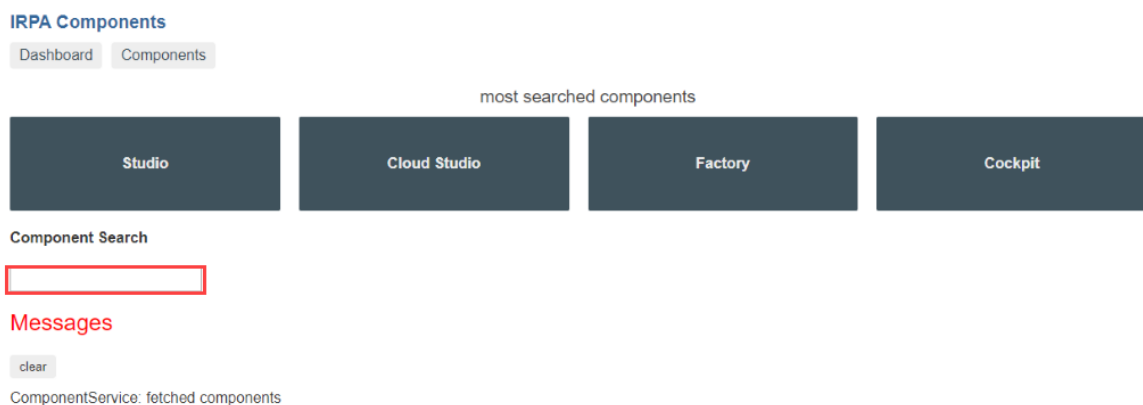
La définition d'une valeur à l'aide de la méthode de définition standard peut ne pas fonctionner sur chaque site Web, en particulier sur un site Web doté d'une infrastructure riche (par exemple, Bootstrap, Angular...).

Ce comportement est lié à :

- des Événements Javascript rattachés aux éléments sur la page Web
- des contrôles de surface
- une structure de page complexe

L'exemple suivant illustre un cas standard pour un site Web avec une infrastructure riche.

1. Lors de la définition d'une valeur, l'utilisateur obtient une liste d'entrées possibles sous forme d'info-bulle dans la barre de recherche "Recherche de composants".



Un nouveau message apparaît sur la page pour indiquer le type de recherche qui a été effectué et si une valeur inconnue a été définie.

2. L'utilisateur définit l'événement de définition simulé suivant :

```
IRPAComponent.pMain.oSearchBox.set("IRPA");
```


IRPA Components

Dashboard Components

most searched components



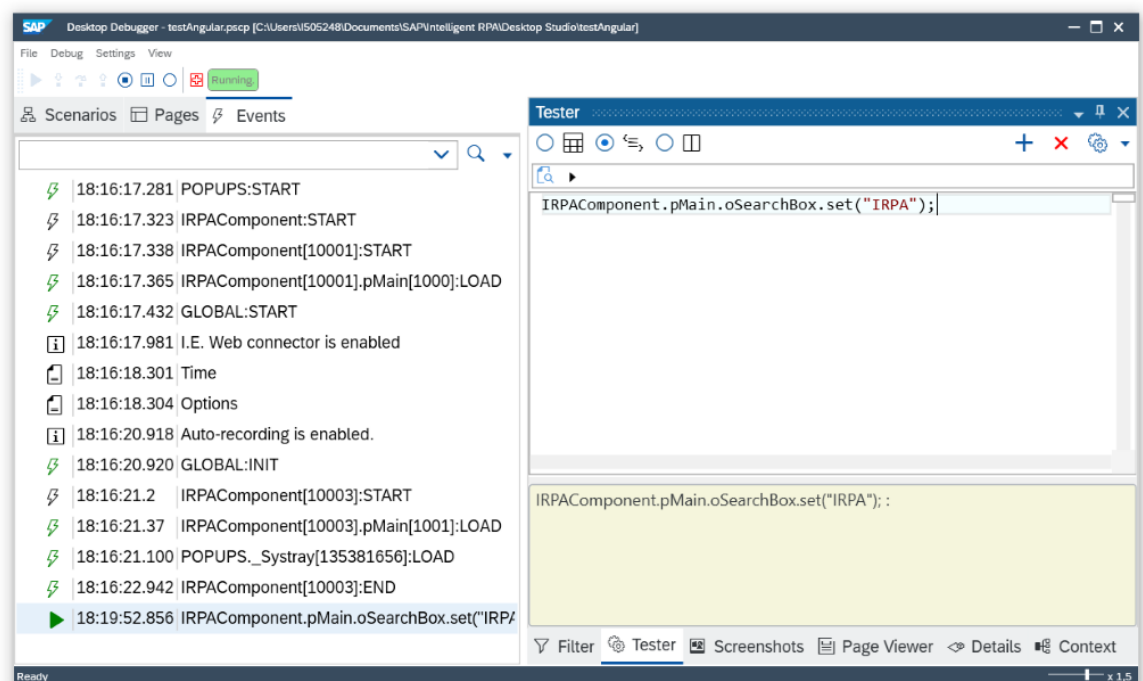
Component Search

IRPA

Messages

clear

ComponentService: fetched components



L'événement de définition ne fonctionne pas pour les raisons suivantes :

- Aucune info-bulle n'est affichée.
- Aucun nouveau message n'est affiché.

3. L'utilisateur définit les séquences de frappes de clavier simulées suivantes :

```
IRPAComponent.pMain.activate();
IRPAComponent.pMain.activate();
IRPAComponent.pMain.oSearchBox.setFocus();
ctx.keyStroke("IRPA");
```

ⓘ Remarque

Dans un navigateur Web tel que Chrome, il est nécessaire d'activer deux fois puis de définir le focus sur l'élément auquel vous voulez envoyer une séquence de frappes de clavier.

Pour éviter une séquence de frappes de clavier simulée, vous ne devez pas l'envoyer directement à l'élément.

En conséquence, une info-bulle et un message s'affichent.

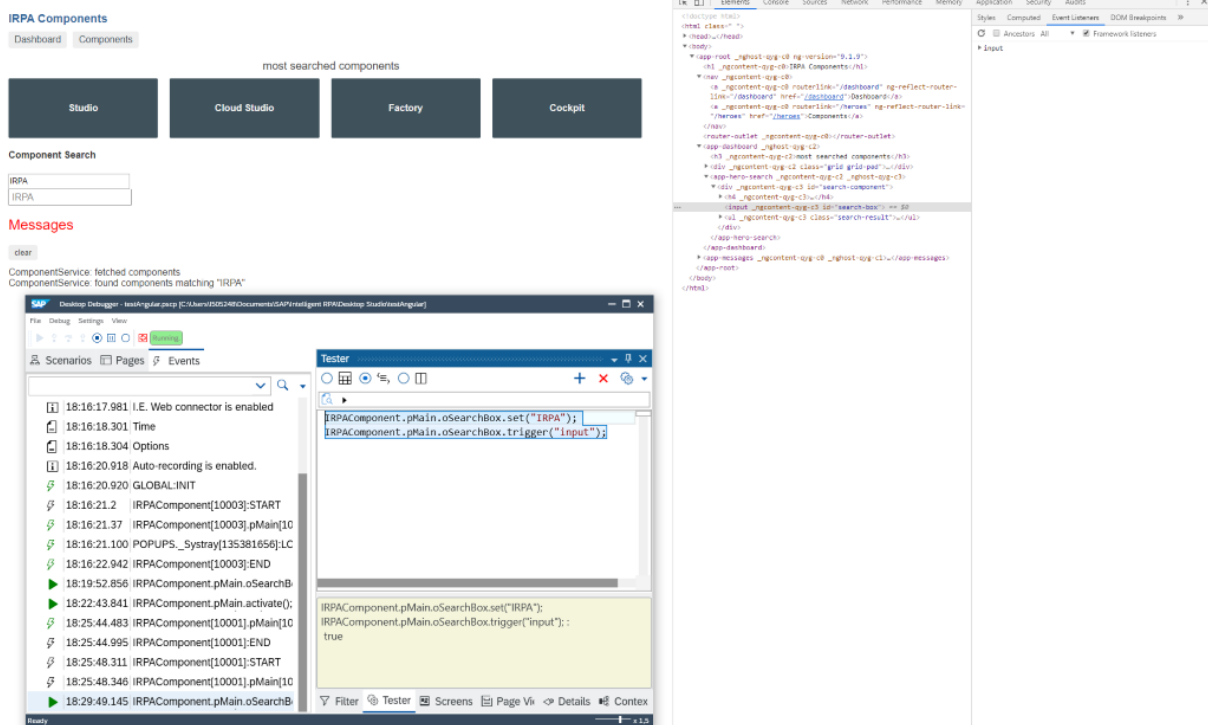
The screenshot displays the SAP Studio interface. At the top, there are tabs for 'Dashboard' and 'Components'. Below this, a section titled 'most searched components' features four dark blue buttons: 'Studio', 'Cloud Studio', 'Factory', and 'Cockpit'. Underneath is the 'Component Search' section with two input fields, both containing the text 'IRPA'. A 'Messages' section follows, with a 'clear' button and two lines of log text: 'ComponentService: fetched components' and 'ComponentService: found components matching "IRPA"'. The bottom portion of the image shows a 'Desktop Debugger' window. The left pane lists a series of log events with timestamps and component names, such as '18:16:17.281 POPUPS:START' and '18:22:43.841 IRPAComponent.pMain.activate();'. The right pane, titled 'Tester', shows a code editor with the following JavaScript code:

```
IRPAComponent.pMain.activate();  
IRPAComponent.pMain.activate();  
IRPAComponent.pMain.oSearchBox.setFocus();  
ctx.keyStroke("IRPA");
```

Cette solution peut être appliquée aux écrans déverrouillés, mais elle ne s'applique pas aux machines dont l'écran est verrouillé et l'infrastructure de bureau virtuel déconnectée. Avec un écran verrouillé, `ctx.keyStroke` est bloqué.

Un autre moyen de résoudre ce problème est d'utiliser des événements rattachés à l'élément :

```
IRPAComponent.pMain.oSearchBox.set("IRPA");  
IRPAComponent.pMain.oSearchBox.trigger("input");
```



L'info-bulle et le message correspondant sont alors affichés. Cette solution peut être appliquée à des machines avec un écran verrouillé et des infrastructures de bureau virtuel déconnectées, tant que la session est active.

Une page Web plus complexe peut nécessiter que l'utilisateur travaille directement dans le navigateur en utilisant le mode développeur pour trouver une solution.

Par exemple, il est possible de simuler un événement de souris spécifique sur la page :

```
application.page.item.scriptItem( "dispatchEvent( new MouseEvent( 'mouseover', { 'bubbles': true, 'cancelable': true } ) )");
```

Il est également possible de le faire pour un événement de clavier, pour une entrée qui nécessite uniquement une entrée de touche de clavier spécifique.

1.2.1.2.4 Maintenir l'écran déverrouillé

Configurer votre environnement pour empêcher le verrouillage de l'écran fait partie des meilleures pratiques les plus courantes.

Cela permet d'utiliser des clics de souris et des frappes de clavier sans être bloqué et renvoyé au verrouillage d'écran au lieu de l'application souhaitée.

1.2.1.2.5 Maintenir une connexion dans l'infrastructure de bureau virtuel

Si vous utilisez une infrastructure de bureau virtuel et que des clics de souris et des frappes de clavier sont nécessaires, vous pouvez :

- configurer l'infrastructure du bureau virtuel de façon à empêcher le verrouillage de l'écran ;
- vous connecter directement à une infrastructure de bureau virtuel à partir d'un ordinateur local.

Il se peut que vous disposiez d'un projet qui lance cette connexion à partir d'un ordinateur local.

1.2.1.2.6 Identifier une autre résolution fonctionnelle

Si aucune des résolutions proposées précédemment ne fonctionne, essayez de trouver une solution de contournement fonctionnelle. Une autre séquence d'écran, un autre bouton sur lequel cliquer, un autre moyen de valider votre formulaire peut également fonctionner sans clic de souris ni frappe de clavier.

Si aucune des solutions de contournement ne s'applique, cela peut bloquer votre projet.

Il est important de repérer ces restrictions dès le début de la démonstration de faisabilité, de la conception ou de l'implémentation. Pour ce faire, vous pouvez :

- Utiliser une action + un délai + un verrouillage manuel de l'écran
- Réaliser des tests réguliers sur une infrastructure de bureau virtuel déconnectée ou sur une machine avec un écran verrouillé

1.2.2 Meilleures pratiques pour la bibliothèque PDF

La bibliothèque PDF permet d'extraire des données de documents PDF pouvant faire l'objet d'une recherche textuelle.

Dans certains cas, la [bibliothèque PDF](#) ne peut pas localiser correctement le texte recherché dans un document. En effet, l'ordre des éléments de texte renvoyés peut ne pas correspondre aux coordonnées visuelles que vous utilisez pour les afficher.

Pour relever ces défis, nous vous recommandons ce qui suit :

1. Utilisez la méthode [getText](#) pour récupérer le texte entier du document PDF.
2. Analysez la structure de l'index extrait pour localiser les éléments de texte que vous souhaitez récupérer.
3. En fonction de la structure de l'index, utilisez une méthode ou activité appropriée pour récupérer l'élément de texte ([getWordAfter](#) ou [getWordBefore](#) par exemple).

ⓘ Remarque

Vous ne pouvez pas appliquer de solution de contournement unique ici. Dans certains cas, passer de [getWordAfter](#) à [getWordBefore](#) (ou l'inverse) peut fonctionner. Dans des cas plus complexes, vous fouiller davantage la structure d'index et en rechercher une qui reste stable.

1.2.3 Meilleures pratiques pour les appels de service Web

Contexte

Lorsque vous utilisez la fonction `ajax.call` pour communiquer avec un service Web, vous recevez un statut du serveur comme :

- 200.
- 300.
- Etc.

Parfois, il semble que l'appel ne fonctionne pas et vous recevez un message d'erreur générique. Cela peut être :

- Le statut uniquement.
- Un message générique.

Il peut être difficile de savoir d'où vient le problème.

Motifs

Les services Web peuvent combiner de nombreux paramètres différents :

- Certificats côté serveur/client.
- Les certificats SSL, qui chiffrent les informations, fournissent une authentification, etc. Ils sont nécessaires si vous souhaitez vous authentifier sur le serveur.
- Jetons.

Il existe différents types de services Web :

- SOAP.
- ODATA.
- REST.

Si vous combinez tous ces services Web et paramètres, vous obtiendrez de nombreuses possibilités. Ainsi, il existe de nombreuses manières de configurer la méthode `ajax.call`.

De plus, vous pouvez être confronté(e) à une mauvaise configuration des serveurs ou des certificats pouvant entraîner des problèmes spécifiques.

1.2.3.1 Causes et résolutions

Dans cette partie, vous trouverez les principales causes des problèmes et comment les résoudre.

Dans les rubriques suivantes, vous en apprendrez davantage sur :

- Les certificats client.
- Les certificats de serveur.
- Les cookies.
- Les jetons.

1.2.3.1.1 Certificats client

Vous utilisez des certificats client pour vous authentifier sur un serveur. Parfois, vous ne pouvez pas vous connecter à un serveur avec un certificat.

L'option `ignoreClientCertificate` vous permet d'éviter l'authentification avec certificat.

Cette option utilise une implémentation C++ de l'appel.

Exemple :

```
ctx.ajax.call({
  url: url,
  method: 'GET',
  contentType: "application/json",
  headers: headers,
  ignoreClientCertificate: true,
  success: function(res, status, xhr) {
    rootData.xCSRF = xhr.headers["x-csrf-token"];

    var value = xhr.headers["Set-Cookie"];
    rootData.Cookies = value.split('; ')[0] + "; ";

    sc.endStep();
  },
  error: function(xhr, status, statusText) {
    ctx.log('error get')
    sc.endStep();
  }
});
```

1.2.3.1.2 Certificats de serveur

L'API peut avoir besoin de tels certificats. Ils sont généralement obligatoires.

`ctx.ajax.call` n'est pas en mesure de forcer l'appel sans ces certificats.

De plus, vous ne pouvez pas indiquer les certificats à utiliser car vous n'êtes pas dans un environnement de navigateur.

Pour en savoir plus sur la manière de résoudre ce problème, voir : **Comment résoudre les problèmes d'appel Ajax** dans [Appel Ajax \[page 27\]](#)

1.2.3.1.3 Cookies

Les cookies vous permettent d'authentifier des demandes client et de conserver toutes les informations de session.

Si vous avez besoin d'utiliser des cookies dans vos appels, vous devez définir un cookie pour conserver ces informations. Vous pourrez ensuite l'utiliser dans un autre appel.

Exemple :

```
ctx.ajax.call({ //First call, get the cookies information
  url: "[url]",
  method: e.ajax.method.get,

  contentType: e.ajax.content.json,
  usePassport: true,
  ignoreClientCertificate:true,

  headers : {
    'Authorization': 'Basic [base 64 login:password]',
    'X-CSRF-Token': 'Fetch'
  },
  success: function(res, status, xhr) {

  ctx.wait(function(ev) {
    try {

      var csrf = xhr.headers['x-csrf-token'];
      var cookie = xhr.headers['Set-Cookie'];
      var myUrl = "https://domainName/sap/opu/odata/sap/xxx";

      ctx.wait(function(ev) {
        ctx.ajax.call({ //Second call, use the cookies in the headers object
          url: myUrl,
          method: e.ajax.method.post,

          ignoreClientCertificate:true,
          contentType: 'application/json;charset=utf-8',
          headers : {
            'Content-type': 'application/json;charset=utf-8',
            'X-CSRF-TOKEN': csrf,
            'Cookie': cookie
          },
          success: function(res, status, xhr) {

          },
          error: function(xhr,res,status) {
          }
        });
      }, 0);

    } catch (ex) {

    }
  }, 0);
},
});
```

1.2.3.1.4 Jetons

Un jeton est une donnée créée par un serveur qui contient des informations permettant d'identifier un utilisateur et la validité du jeton. Si vous utilisez le jeton de manière incorrecte dans l'appel Ajax, cela peut entraîner un problème. Comme les cookies, le jeton nécessite deux appels.

Par exemple, vous effectuez un appel d'API dans Factory. Vous utilisez le premier appel pour obtenir le jeton. Le deuxième appel déclenche l'appel d'API et crée un job dans la file d'attente des jobs. Dans ce cas, les appels sont liés. Le deuxième appel a lieu si le premier s'est déroulé correctement.

Exemple :

```
ctx.ajax.call({ //First call, get the token

    url: serviceAccount.urls.accessTokenURL, //Url got from the service key "url
parameter + /oauth/token?grant_type=client_credentials"

    method: e.ajax.method.get,

    contentType: e.ajax.content.json,

    header:{

        Accept: e.ajax.content.json,

        Authorization: 'Basic ' +
ctx.base64.encode(serviceAccount.credentials.clientID + ':' +
serviceAccount.credentials.clientSecret, false) // clientID and clientSecret got
from the service key

    },

    success: function(res, status, xhr) {

        ctx.log('Token generated');

        sc.localData.Token = 'Bearer ' + ctx.get(res, 'access_token');

        ctx.ajax.call({ // Second call

            url: serviceAccount.urls.apiTriggerURL, //Url from the API trigger
(end with "/run")

            method: e.ajax.method.post,

            contentType: e.ajax.content.json,

            header:{

                Accept: e.ajax.content.json,

                'irpa-trigger-token':
serviceAccount.urls.irpaTriggerToken, //Token get from the API trigger

                Authorization: sc.localData.Token

            },

            data:{

                //Payload of your API Trigger

            },
```



```

        success: function(res, status, xhr) {
            ctx.log('success');
            sc.endStep();
            return;
        },
        error: function(xhr, status, statusText) {
            //web service failed
            sc.setError(e.error.Fail, 'Web service failed: [' + status + ']' +
            statusText + ' - ' + xhr.responseText);
            sc.endScenario();
            return;
        }
    });
},
error: function(xhr, status, statusText) {
    //web service failed
    sc.setError(e.error.Fail, 'Web service failed: [' + status + ']' +
    statusText + ' - ' + xhr.responseText);
    sc.endScenario();
    return;
}
});

```

1.2.3.2 Exemples des meilleures pratiques

Voici quelques exemples des meilleures pratiques pour les appels de service Web.

Comme indiqué précédemment, en raison de la complexité de l'appel Ajax, vous pouvez faire face à différents problèmes.

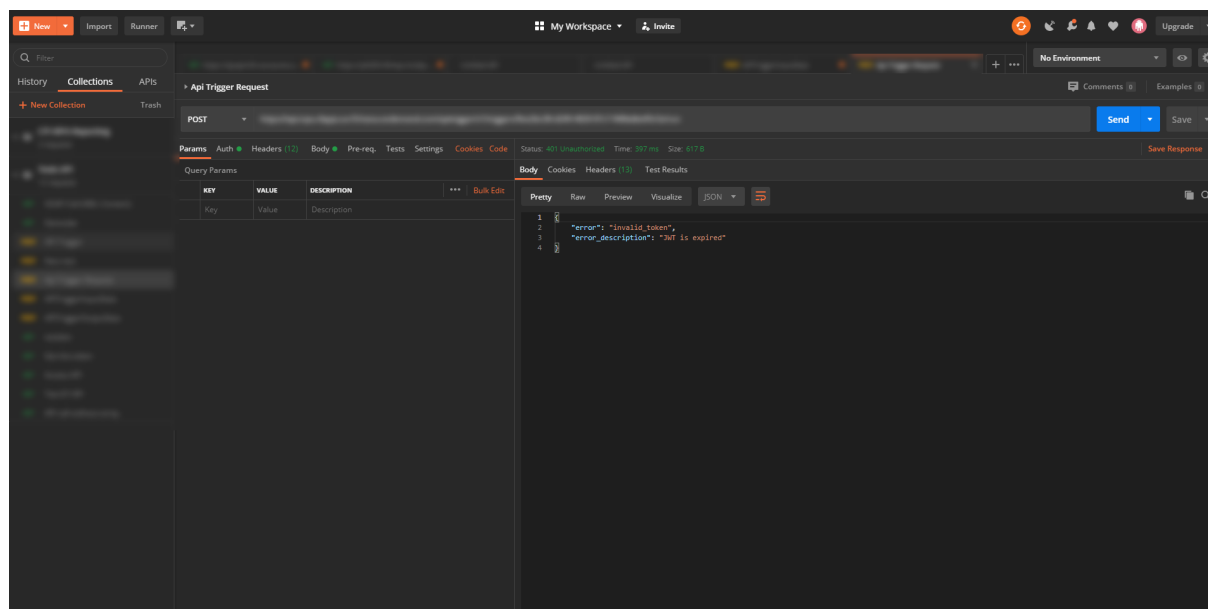
Pour évaluer ces problèmes avec l'appel Ajax, vous pouvez utiliser plusieurs outils :

- Postman.
- cURL.
- Commandes de terminal.
- Appel Ajax.

1.2.3.2.1 Postman

Postman est un outil qui vous permet d'interroger et de tester les services Web et les appels d'API.

Cet outil présente plusieurs fonctionnalités et est facile à utiliser.



1.2.3.2.2 cURL

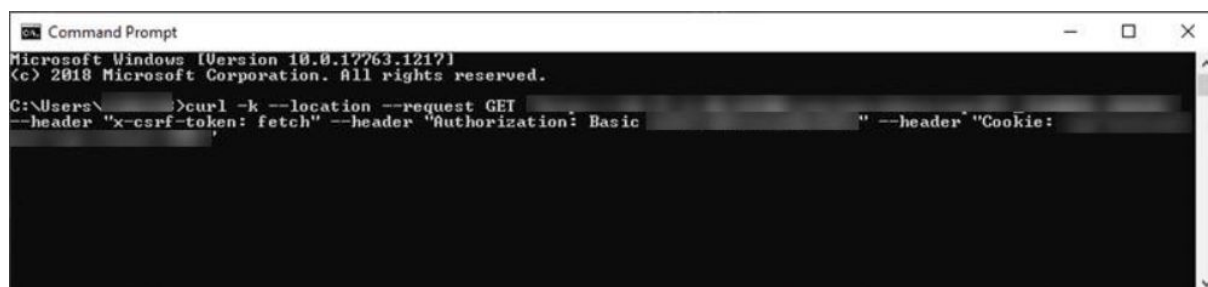
cURL est un outil de ligne de commande que vous pouvez utiliser pour obtenir ou envoyer des données, y compris des fichiers, à l'aide de la syntaxe URL.

Sous Windows 10 build 17063, cURL est inclus par défaut.

1.2.3.2.3 Commandes de terminal

Dans l'invite de commande, vous pouvez tester la commande cURL pendant l'évaluation.

Si le test réussit, l'agent appelle la commande cURL via le terminal.



1.2.3.2.4 Appel Ajax

Comment configurer votre fonction d'appel Ajax

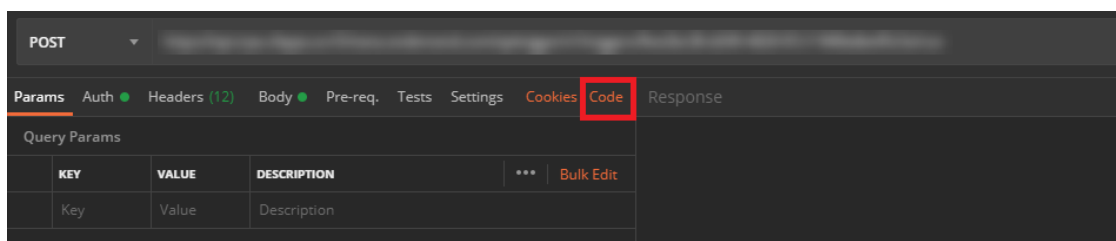
Si vous ne savez pas comment configurer la fonction d'appel Ajax, suivez ce processus :

1. Faites en sorte que la demande fonctionne sur Postman.

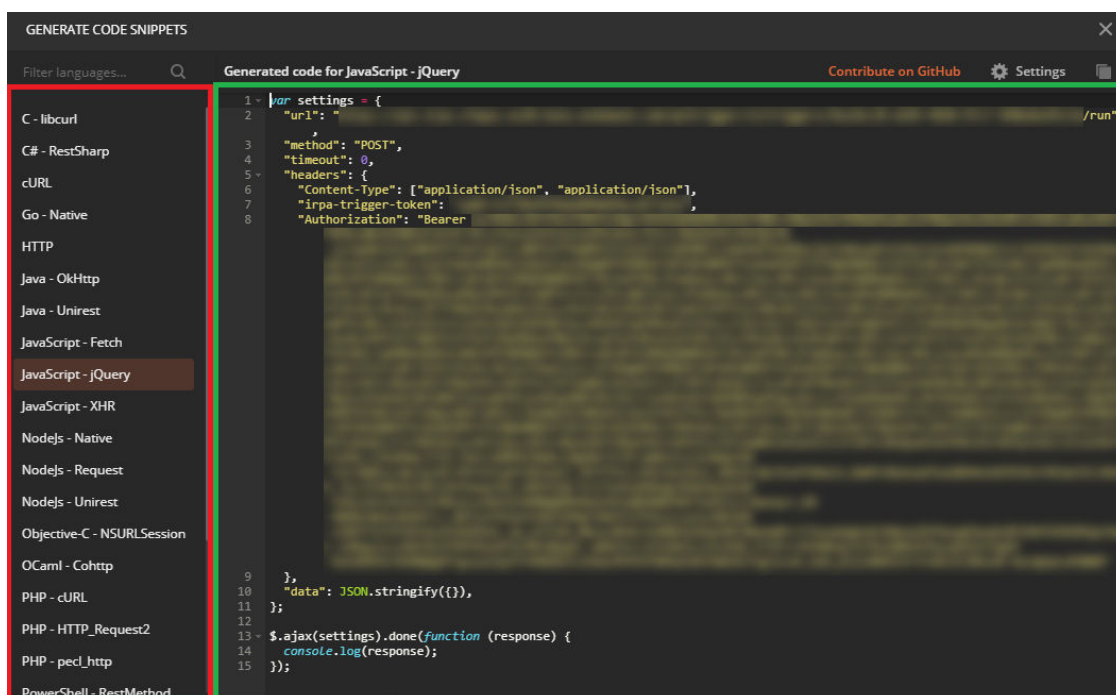
ⓘ Remarque

Si la demande ne fonctionne pas même avec Postman, cela signifie que vous ne pouvez pas la faire fonctionner avec la fonction d'appel Ajax.

1. Lorsque la demande fonctionne, cliquez sur le bouton [Code](#).



2. Sélectionnez le langage de programmation avec lequel vous travaillez.



ⓘ Remarque

Vous pouvez voir différentes langues dans le cadre rouge à gauche. Ici, vous sélectionnez le langage JavaScript-jQuery car la fonction d'appel Ajax a été développée avec la méthode jQuery.ajax(). Dans le cadre vert, vous remarquez le code généré par Postman.

2. Utilisez ce code dans la fonction d'appel Ajax.
Copiez-collez ce code dans Desktop Studio et ajustez-le si nécessaire. Pour en savoir plus, voir [call](#).

```
function getCall(success, error, localData){
    var result;
    var url = localData.fetchAPI = "";

    //API call to get csrf token and cookies
    ctx.ajax.call({
        method: e.ajax.method.get,
        url: url,
        contentType: e.ajax.content.json,
        headers: {
            'Authorization': localData.auth,
            'x-csrf-token' : 'fetch',
            'Cookie' : 'fetch',
            'Accept' : 'application/json'
        },
        ignoreClientCertificate: localData.clientCert,
        success: function(res, status, xhr) {

            if(res.d){
                result = xhr;
                success(result);
            } else {
                result = res;
                error(result);
            }
        },
        error: function(xhr, error, statusText) {

            result = res;
            error(result);
        }
    });
};
```

Comment résoudre les problèmes d'appel Ajax

Si vous faites encore face à des problèmes pendant l'appel (problèmes issus de l'objet xhr) après avoir effectué le test dans le processus précédent, essayez d'effectuer les tests suivants :

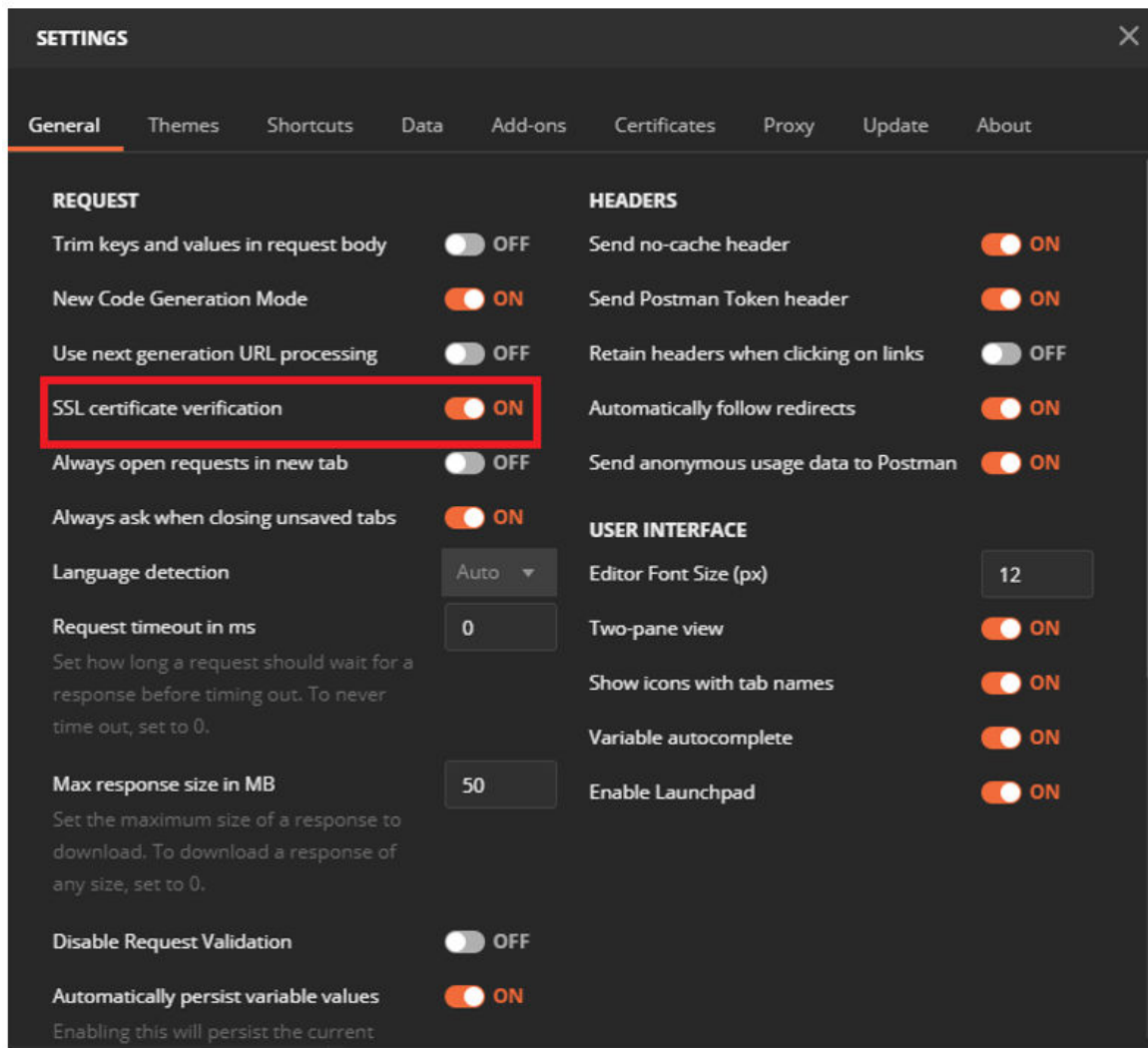
1. Essayez `ctx.ajax.call` avec l'option `ignoreClientCertificate` comme vraie.

ⓘ Remarque

Si cela ne fonctionne toujours pas, vérifiez le code d'erreur qui peut vous montrer l'origine de l'erreur.

```
▲ [Object] response = {...} [Object]
  responseText = "A security error occurred" [String]
  status = 12175 [Number]
  statusText = "A security error occurred" [String]
```

2. Modifiez les paramètres dans Postman.

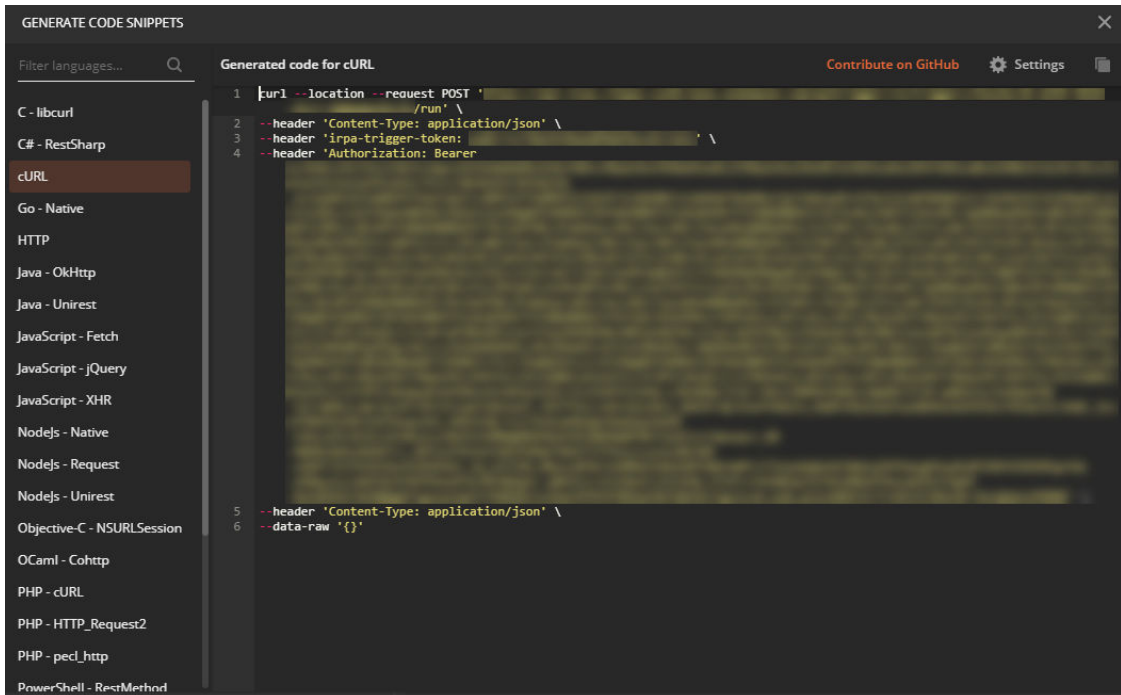


Par défaut, la *vérification de certificat SSL* n'est pas activée dans Postman. Cela signifie qu'il force l'appel même si le certificat n'est pas disponible.

Si l'appel sur Postman ne fonctionne pas après avoir activé cette option, cela signifie que ce certificat est obligatoire pour passer l'appel. Comme indiqué précédemment, l'appel Ajax n'a pas cette option.

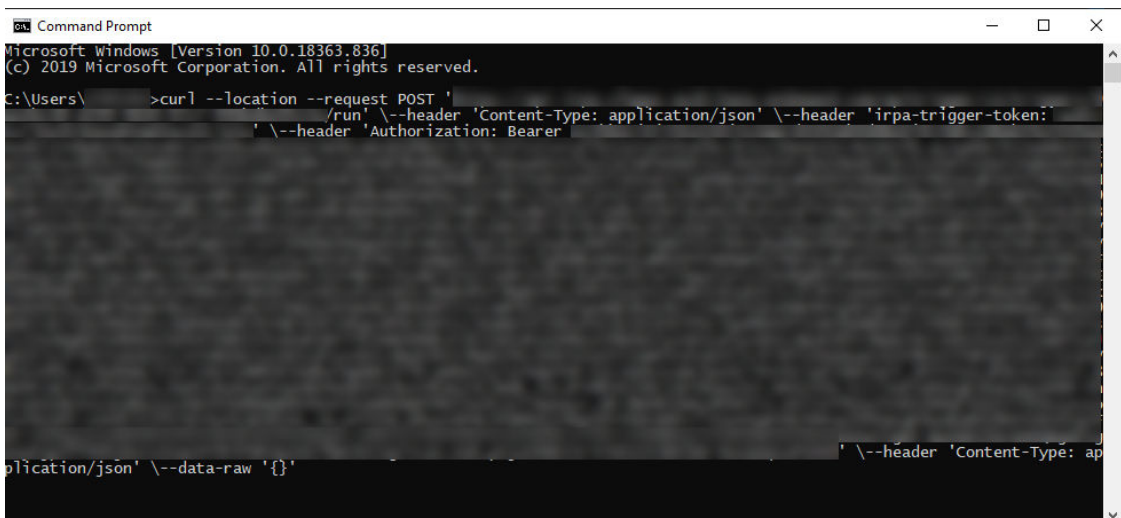
Une solution peut consister à utiliser cURL.

3. Essayez le code cURL fourni par Postman.
 - Suivez le processus **Comment configurer votre fonction d'appel Ajax** ci-dessus, de l'**étape 1.a.** à l'**étape 1.b.** pour récupérer le bon code cURL.



- Dans ce code, supprimez tous les caractères \ et remplacez tous les guillemets simples par des guillemets doubles.
 - Exécutez le code cURL sur l'invite de commande.
- Exemple :

```
curl --location --request GET "http://XXXX"
```



Si vous avez un message d'erreur qui s'affiche avec le code cURL, effectuez une recherche sur Internet. Les messages d'erreur sont généralement très explicites. Si le code est correct, vous pouvez l'intégrer dans un appel SAP Intelligent RPA.

- Implémentez ce code dans Desktop Studio.

```

var cmdCurl = 'curl -k --location --request GET "http://XXXX"
ctx.exec(cmdCurl, 60, function(reqres) {
    var reqdata = ctx.json.parse(reqres.output);

```

```
});
```

ⓘ Remarque

L'option `-k` est utilisée pour ignorer les certificats client.

1.2.4 Meilleures pratiques pour le traitement d'une quantité importante de données lors de l'exécution d'un scénario

Les développeurs doivent s'assurer que de grandes quantités de données ne sont pas placées dans l'objet de données de scénario lors de l'exécution du scénario, car cet objet entraîne un retour vers Cloud Factory.

Pour contourner ce problème, si une grande quantité de données est générée et manipulée au cours du scénario, cela peut entraîner un vidage des données (partiel ou total) lors de la dernière étape du scénario.

Exemple :

```
...  
// during scenario execution  
sc.data.bigData = ctx.fso.file.open(theBigFile);  
...  
// last step before ending the scenario  
sc.data.bigData = undefined;  
sc.endStep();
```

1.2.5 Meilleures pratiques pour les captchas

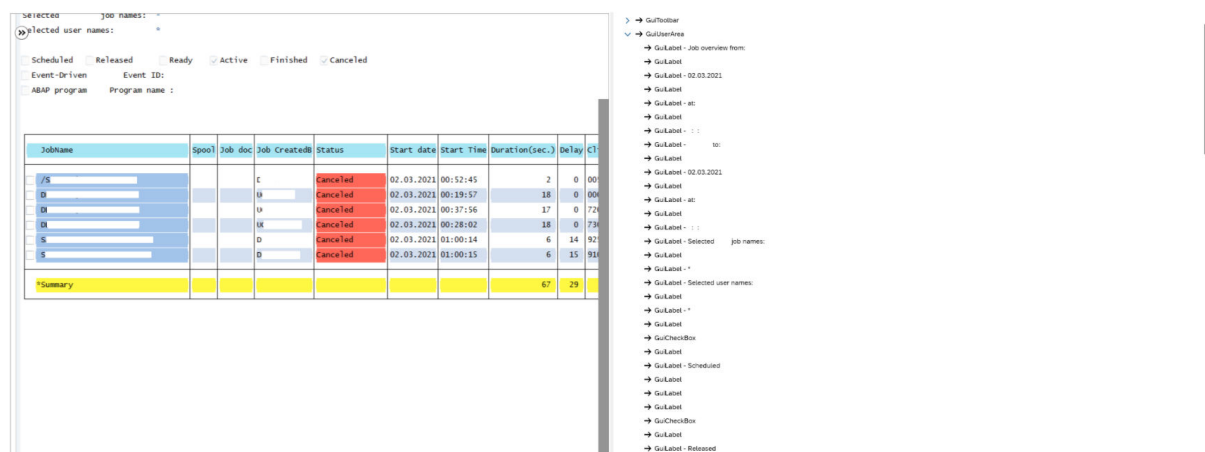
Vous pouvez parfois rencontrer des captchas lorsque vous souhaitez automatiser un site Web.

Vous pouvez traiter les captchas de deux façons :

- Vous pouvez essayer de résoudre les captchas, par exemple à l'aide d'une bibliothèque de reconnaissance d'images externe (le développement du processus peut s'avérer difficile).
- Vous pouvez également essayer de détecter le moment où le captcha s'affiche et éviter ce comportement. Par exemple, un captcha peut s'afficher lorsque les champs d'un formulaire sont renseignés trop rapidement. Dans ce cas, la meilleure pratique consiste à ajouter un délai de trois secondes entre chaque étape de votre automatisation, par exemple.

1.2.6 Meilleures pratiques pour l'automatisation des écrans SAP GUI avec des hiérarchies horizontales

Lorsque vous capturez une table avec des hiérarchies horizontales dans Desktop Studio, un grand nombre d'éléments sont affichés dans le testeur d'application.



Reconnaître tous les éléments à l'écran nécessite un grand nombre de tests pouvant entraîner une mauvaise performance lors de l'exécution du bot.

Pour gérer ce comportement, vous pouvez éviter de lire la table dans Desktop Studio et lire à la place la table dans Excel. Pour exporter la table dans un fichier Excel, vous devez appuyer sur **MAJ** + **4** et capturer deux fenêtres temporaires.

Une fois que vous avez capturé les deux fenêtres temporaires, vous pouvez automatiser les écrans capturés avec la procédure suivante.

Procédure

1. Exécutez l'action *Frappe* à l'écran que vous avez capturé avec la table.
La première fenêtre temporaire s'affiche.

☰
Select Spreadsheet
✕

Formats:

Excel (in MHTML Format)
 OpenOffice (in OpenDocument Format 2.0)
 Select from All Available Formats

31 Excel - Office Open XML Format (XLSX) ▾

Always Use Selected Format

Continue
Cancel

2. Cliquez sur *Poursuivre*.
La deuxième fenêtre temporaire s'affiche.

☰
Job Overview
✕

Directory: C:\TEMP

File Name: EXPORT.XLSX

Generate
Replace
Extend
⊗

3. Cliquez sur *Remplacer*.

Résultat

La table est ouverte et sauvegardée dans le fichier Excel.

	A	B	C	D	E	F	G	H	I	J	K
	JobName	Spool list	Job documentation	Job CreatedBy	Status	Start date	Start Time	Duration(s ec.)	Delay (sec.)	Client	Reason for Delay
1											
2	/S			D	Canceled	05.03.2021	00:52:47	0	0	005	
3	/S			D	Canceled	05.03.2021	12:52:49	2	2	005	
4	/S			D	Canceled	05.03.2021	00:52:45	2	0	005	
5	DL			U	Canceled	05.03.2021	00:46:26	16	0	000	
6	DL			U	Canceled	05.03.2021	01:05:42	19	0	720	
7	DL			U	Canceled	05.03.2021	01:05:42	19	0	730	
8	S			D	Canceled	05.03.2021	01:00:21	6	21	910	
9	S			D	Canceled	05.03.2021	01:00:21	5	21	925	
10								69	67		
11											
12											

1.2.7 Meilleures pratiques pour la gestion des fichiers

Cette section présente les meilleures pratiques pour la gestion des fichiers.

Fichier de configuration utilisé pour différents utilisateurs et ordinateurs

Un exemple typique est que vous souhaitez créer un fichier ou un dossier à un emplacement local. Demander aux utilisateurs finaux de créer un dossier manuellement pour le bot n'est pas très pratique. De plus, il est difficile de trouver un nom de chemin d'accès conventionnel qui fonctionnerait pour tous les utilisateurs de votre organisation.

Vous pouvez éviter cela en faisant en sorte que votre bot crée ces dossiers par défaut. De cette façon, si le bot ne trouve pas le dossier requis sur l'ordinateur, il le crée. Vous pouvez rendre la procédure encore plus pratique en ajoutant une condition pour vérifier l'existence du dossier.

Exemple de code

```
// this will create a folder 'Config' under
'Desktop\Automation\ProjetName' only if this folder does not exist

var configFile
=ctx.fso.folder.getSpecialFolder(e.shell.specialFolder.Desktop) + "\
\Automations\\"+ ctx.options.projectName + "\\config";
if (!configFile.exists){
    ctx.fso.folder.create(configFile);
}
```

Fichiers de travail

Pour les fichiers temporaires utilisés par le bot, les utilisateurs Windows peuvent avoir besoin d'autorisations spécifiques. La meilleure pratique consiste à créer ces fichiers sous le dossier de journaux du projet. C'est la façon la plus pratique de gérer les fichiers de travail.

Exemple de code

```
//this will create a file 'TempFiles' in your project folder
ctx.fso.folder.create(ctx.options.path.log+"\\TempFiles\\")
```

Gestion du chemin d'accès au dossier à l'aide de variables Factory

Lors de la gestion d'un fichier/dossier commun pour un groupe d'utilisateurs qui ont tous accès à un référentiel partagé. Vous pouvez utiliser la variable de texte Cloud Factory pour archiver le chemin d'accès, puis le récupérer dans votre script comme dans l'exemple ci-dessous.

New Variable

*Name:

Description:

*Type:

*Value:

Pour en savoir plus sur les variables Cloud Factory, voir la section [Declaring Environment Variables from SAP Intelligent RPA Factory](#).

Exemple de code

```

var mypath;
//
-----
// Step: get_path_var
//
-----
GLOBAL.step({ get_path_var: function(ev, sc, st) {
var rootData = sc.data;
ctx.workflow('Custom_getPathVar', 'c514ca1b-3d9d-445b-
a2e3-069c9a7b9397') ;

// Describe functionality to be implemented in JavaScript
later in the project.

// Declare setting path
ctx.setting({ MyPath: {
comment: "sc.localData.mytenant",
server: true
}});
// Get setting
ctx.settings.MyPath.get(function(code, label, setting) {
if (code == e.error.OK) {
// get value from setting.value
mypath = setting.value;
ctx.log(mypath);

} else{
ctx.log('Error during setting retrieval');
sc.endStep(); // pSapGlobalCloud8L_man
return;

}
sc.endStep(); // code_
return;

});

});
}
});

```

Fichiers de sortie

Si votre scénario requiert un ou plusieurs fichiers de sortie, il est recommandé de créer le dossier de sortie localement.

Exemple de code

```
//this will create a folder 'output' under the path  
'Desktop\Automations\ProjectName'  
  
ctx.fso.folder.create(ctx.fso.folder.getSpecialFolder(e.shell.specialFolder.Desktop) + "\\Automations\\" + ctx.options.projectName + "\\output")
```

1.2.8 Meilleures pratiques pour l'extension Chrome

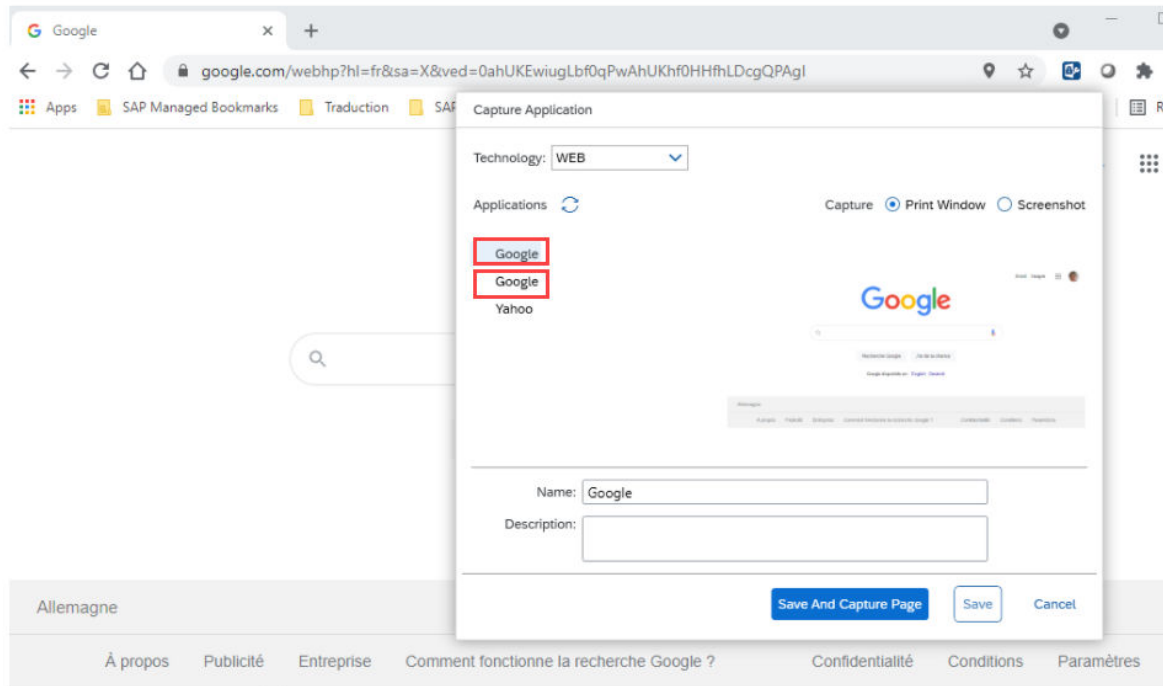
Vous pouvez parfois rencontrer un problème si vous avez activé deux fois l'*extension SAP Intelligent RPA* pour Chrome.

Contexte

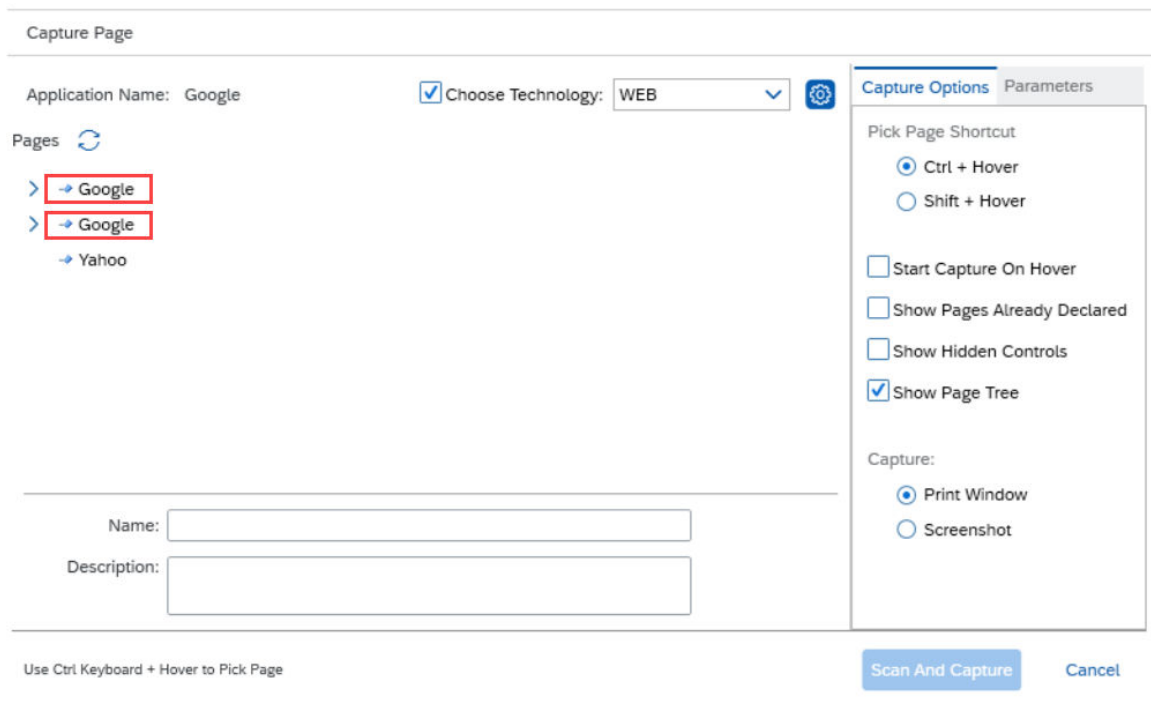
Pour en savoir plus sur la façon d'activer l'extension du navigateur Web pour Chrome, voir [Configuration des navigateurs Internet](#).

Si vous avez activé deux fois l'*extension SAP Intelligent RPA* pour Chrome, vous pouvez rencontrer les problèmes suivants :

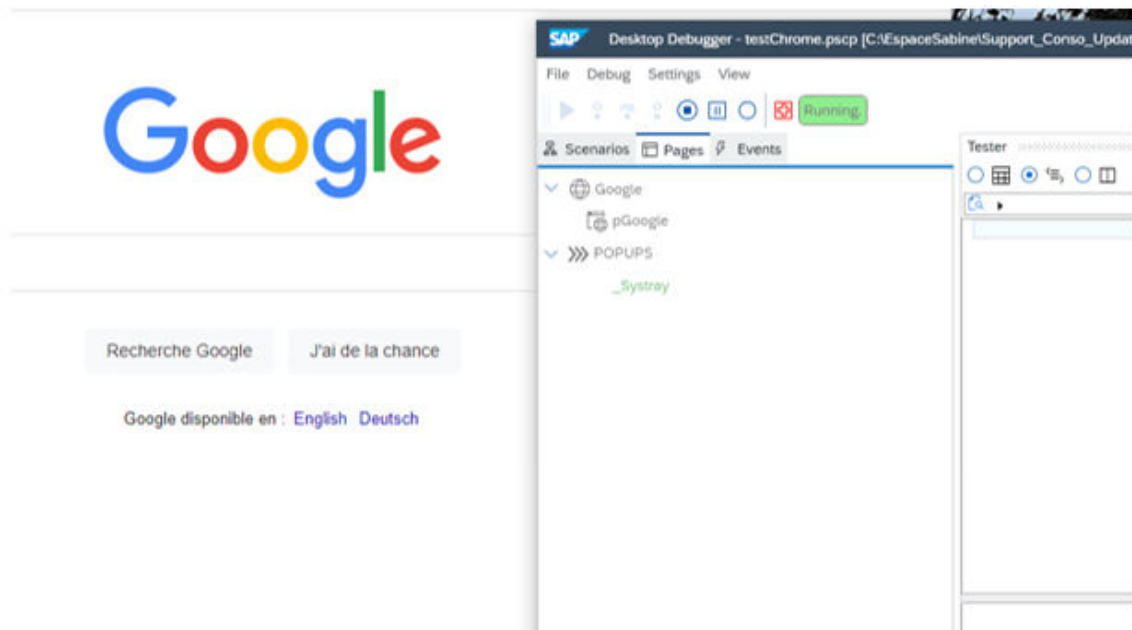
- Lorsque vous voulez capturer une application, vous constatez qu'elle apparaît deux fois dans le panneau *Capturer application*. Cela se produit même si l'une de ces applications est lancée dans le navigateur Chrome.



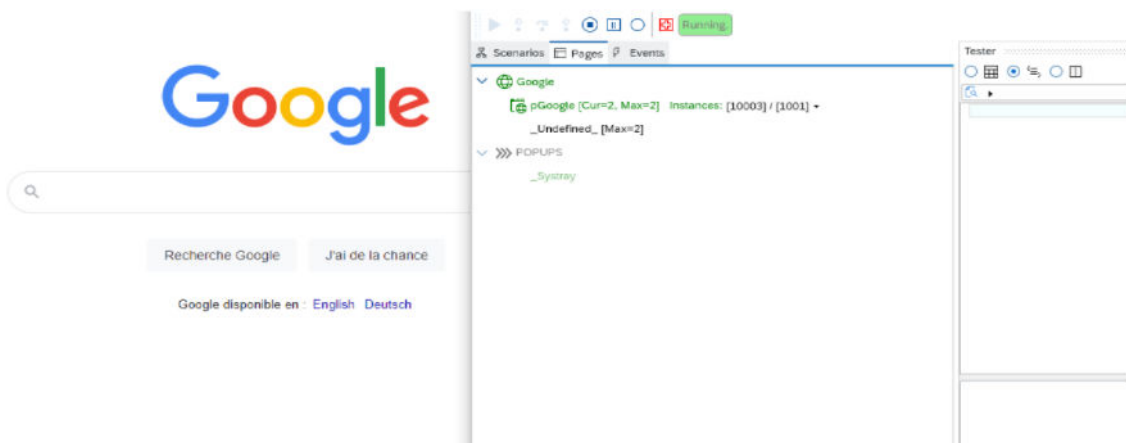
- Lorsque vous voulez capturer des pages à partir d'une application, vous constatez que la même page apparaît deux fois dans le panneau *Capter page*.



- Lorsque l'application Web est lancée dans le navigateur Chrome, la page et l'application ne sont pas reconnues.

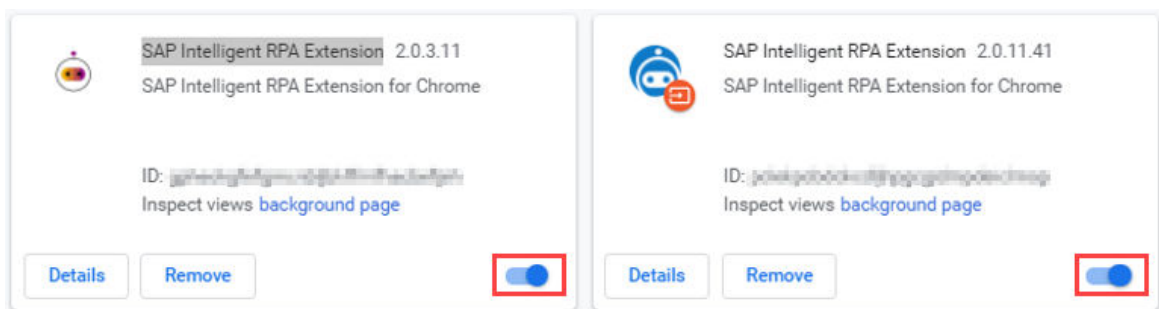


- Pour résoudre ce problème, vous pouvez actualiser la page manuellement. Lorsque la page est reconnue, vous constatez que la même page apparaît deux fois.

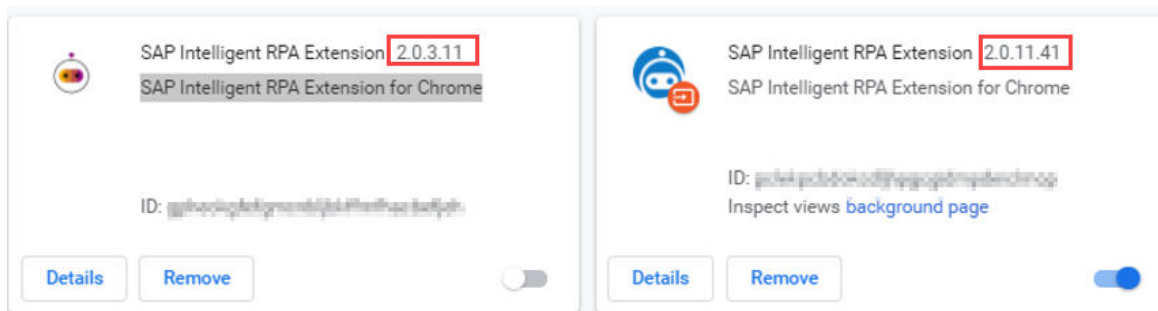


Désactivation de l'une des extensions

1. Saisissez **chrome://extensions/** dans la barre de recherche Google. Vous remarquerez que vous avez deux vignettes Extension SAP Intelligent RPA pour Chrome.



2. Vérifiez les versions des extensions et désactivez celle dont la version est la plus ancienne.



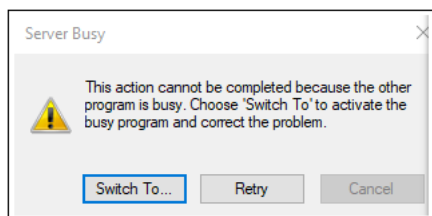
Informations associées

[Configuration des navigateurs Internet](#)

1.2.9 Meilleures pratiques pour la gestion de la fenêtre temporaire **Serveur occupé**

Cette section décrit les meilleures pratiques pour la gestion d'une fenêtre temporaire **Serveur occupé** lors de l'exécution d'une automatisation.

Symptôme



- Le flux d'opérations est figé. Le processus est suspendu et rien n'est traité par la suite.
- Souvent, la fenêtre temporaire *Serveur occupé* ne peut pas être fermée.
- La plupart du temps, le systray de l'Agent ne réagit plus.
- Pour arrêter l'Agent, il faut interrompre le processus.
- La fenêtre temporaire *Serveur occupé* peut parfois disparaître au bout d'un certain temps, puis le processus reprend.

Origine

La fenêtre temporaire *Serveur occupé* est déclenchée lorsque

- L'application cible est entièrement occupée et ne peut pas répondre aux demandes de l'Agent.
 - Un long JavaScript est exécuté sur une page Web.
 - La page télécharge ou charge de grandes quantités de données (téléchargement d'un fichier, recherche de données dans le backend, etc.).
 - L'application est lente ou exécutée dans un environnement sous-dimensionné.
- L'Agent tente d'effectuer une action sur cette même application en même temps (**click()** ou **refresh()** ou ...).

La solution dépend du connecteur technique, mais de manière générale, il est nécessaire de s'assurer que l'application est disponible avant de faire quoi que ce soit d'autre.

Dans certains cas, cela provient entièrement de l'application cible. Cela signifie que cette fenêtre temporaire peut être déclenchée par une simple action manuelle sans aucun Agent en cours d'exécution.

Dans ce cas, il vous appartient de déterminer quand cela peut se produire et comment vous pouvez l'éviter manuellement. Vous devrez alors répercuter cela dans votre flux d'opérations.

L'expérience a toutefois révélé que, la plupart du temps, la fenêtre temporaire est déclenchée par l'Agent lui-même (ctxtrun.exe), même si cela est lié à l'indisponibilité de votre application.

La fenêtre temporaire Serveur occupé devrait disparaître une fois l'Agent arrêté.

Si la fenêtre temporaire provient de l'Agent, il existe deux origines techniques possibles qui doivent être identifiées.

- SDK
 - SDK demande au connecteur technique d'effectuer des actions et demande un feedback.
 - Interrogation personnalisée et interrogation sur un élément (item.wait())
 - Clic sur un élément
 - execScript
 - Connecteur technique
 - Le connecteur, en plus de l'exécution des demandes SDK (voir ci-dessus), demande principalement le statut de l'application traitée.
 - CHARGEMENT, attente de pages
 - Interrogation sur des pages
 - Lancement de l'application

Rappel

Dans tous les cas, il est essentiel d'identifier les actions qui peuvent conduire à cette fenêtre temporaire. La plupart du temps, cette fenêtre temporaire apparaîtra de façon récurrente sur les mêmes étapes du flux d'opérations.

Même si vous demandez de l'aide, nous vous demanderons de nous fournir ces informations.

Pour ce faire, il est vivement recommandé d'utiliser des traces de projet, voir [2776434](#), **SAP Knowledge Base Article**.

Comment vérifier si la fenêtre temporaire *Serveur occupé* vient du côté du connecteur

- Identifiez l'événement qui mène à la fenêtre temporaire dans votre flux d'opérations et toutes les actions environnantes, en utilisant éventuellement des traces de projet. Voir [2776434](#), **SAP Knowledge Base Article**.
 - Un clic sur un bouton
 - Le CHARGEMENT d'une page spécifique
 - L'appel d'un script sur une page
 - Le téléchargement d'un fichier
 - ...
- Lancez le débogueur sans lancer de flux d'opérations.
- Reproduisez manuellement l'action qui aurait été effectuée par le flux d'opérations juste avant l'apparition de cette fenêtre temporaire.
- Si rien n'apparaît, alors ce problème n'est pas déclenché par le connecteur lui-même.
 - Cela signifie que les activités d'attente ou de CHARGEMENT **sur des pages** ne sont pas celles qui déclenchent cette fenêtre temporaire.
 - Il est possible d'attendre ces pages sans risque et la cause probable du problème doit être liée aux actions demandées par l'Agent.
- Si la fenêtre temporaire *Serveur occupé* apparaît :
 - Si l'application est capturée à l'aide de la technologie UIAutomation, consultez la sous-section **Côté connecteur** de la section **Résolution du problème et solution de contournement**.
 - Sinon, ouvrez un incident sous la **composante CA-ML-IPA**. C'est un bloqueur possible.

Comment vérifier si la fenêtre temporaire *Serveur occupé* vient du côté de SDK

- Identifiez l'événement qui mène à la fenêtre temporaire dans votre flux d'opérations et toutes les actions environnantes, en utilisant éventuellement des traces de projet. Voir [2776434](#), **SAP Knowledge Base Article**.
 - Un clic sur un bouton
 - Le CHARGEMENT d'une page spécifique
 - L'appel d'un script sur une page
 - Le téléchargement d'un fichier
 - ...
- Lancez le débogueur sans lancer de flux d'opérations.
- Copiez et collez les actions (action spécifique et actions environnantes) du flux d'opérations qui semblent entraîner ce problème.
 - Supprimez toutes les activités de scénario (endStep, endScenario, return, gestion des données, etc.).
 - Il s'agira souvent d'un clic sur un élément qui conduit à un gestionnaire d'attente ou de CHARGEMENT de page, qui conduit à son tour à une action spécifique comme un autre clic ou une interrogation d'élément ou l'exécution d'un script personnalisé.

Dans le cas de la fenêtre temporaire *Serveur occupé*, nous avons identifié plusieurs activités possibles qui entraînent son apparition (liste non exhaustive). Ce sont des activités SDK. SDK attend un feedback du connecteur technique, qui est peut-être en suspens dans l'attente d'un feedback de l'application indisponible :

- Interrogation personnalisée sur une page (ctx.polling)
- Interrogation d'élément (app.page.myElement.wait)
- refresh() avec la technologie UIAutomation

- execScript avec la technologie Web

Certaines activités ne demandent pas un feedback de ce type et peuvent ainsi résoudre le problème ou fournir une solution de contournement :

- Utiliser l'attente (wait) sur une page autant que possible
 - Une attente longue doit exister sur un élément pour éviter l'interrogation de l'élément (item.wait).
- lockRefresh/unlockRefresh avec la technologie UIAutomation pour éviter une saturation de l'application par le connecteur

Résolution du problème et solution de contournement

Dans ce cas, il est très important d'avoir une [application optimisée \[page 63\]](#). L'optimisation d'une application réduira le coût de demande de l'état de l'application, de la page et de l'élément.

Cela limitera alors la pression sur les applications potentiellement complexes.

📌 Remarque

La meilleure pratique en matière de reconnaissance avec UIAutomation consiste à éviter les activités Doit exister/Ne doit pas exister lorsque cela est possible. Dans le cas présent, il se peut que nous devions les utiliser en raison de la nécessité d'éviter les interrogations sur des éléments (ctx.polling/app.page.element.wait).

Côté SDK

Technologie UIAutomation/Web

Votre page a mis beaucoup de temps à CHARGER/télécharge ou charge des données et vous rencontrez ce problème lors de l'envoi d'une action à la page. Cette page n'est peut-être pas encore complètement disponible ou n'a pas été chargée avec des données complètes.

- Recherchez tout élément qui indiquera de façon certaine le CHARGEMENT complet de votre page.
 - Mise à jour du titre
 - Mise à jour de la pagination
 - Élément complètement chargé
 - Élément qui n'est plus grisé
 - Disparition de l'icône de chargement
 - ...
- Déclarez-le comme [devant exister \(MustExist\)](#) sur cette page, avec des critères appropriés indiquant l'état attendu pour l'élément (non grisé, n'en afficher aucun, etc.).
- L'attente de la page associée entraînera l'attente de cet élément sans interrogation de la par de SDK.
 - L'exécution d'une activité item.wait est une demande SDK.
 - Le fait d'avoir une activité Doit exister sur le même élément relève uniquement du connecteur (aucun feedback n'est demandé par SDK > aucune fenêtre temporaire Serveur occupé n'apparaît).

Technologie GUI

Sous-pages

Une sous-page peut provoquer ce problème. Vous devez alors éviter d'avoir des sous-pages GUI. Si une sous-page GUI n'est pas reconnue, SAPGUI se fige, ce qui entraîne l'affichage d'une fenêtre temporaire Serveur occupé.

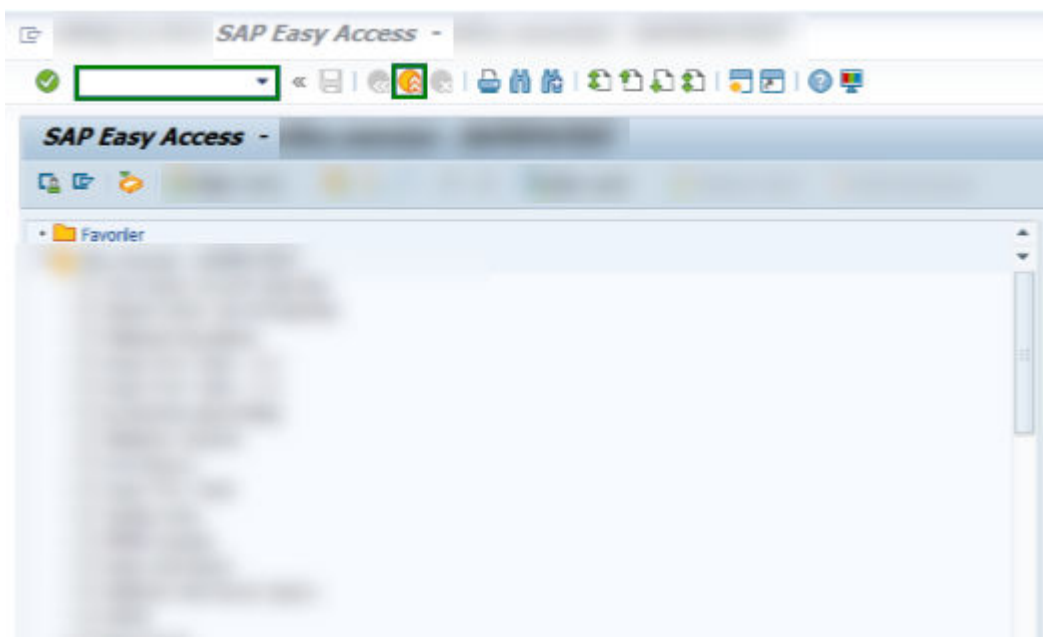
Doit exister/Ne doit pas exister (optimisation passive)

Le fait d'avoir trop d'éléments Doit exister/Ne doit pas exister augmente le coût de la reconnaissance de page. Cela aura un impact direct sur la disponibilité de l'application et sur l'apparition de la fenêtre temporaire. Il est donc conseillé de limiter leur nombre au maximum.

Optimisation active pour GUI

Malgré la suppression des sous-pages et la limitation du nombre d'éléments Doit exister et Ne doit pas exister, vous risquez tout de même de rencontrer le problème et, dans ce cas, vous devez appliquer l'optimisation active. Cela s'effectue par l'utilisation d'une activité et d'une méthode spécifiques dans le flux d'opérations : **setBusyWaitTime**, **resetBusyWaitTime**, **lockRefresh** et **unlockRefresh**.

- **setBusyWaitTime** et **resetBusyWaitTime**
La transaction SAPGUI peut être occupée lors du chargement de pages ou de l'exécution de tâches spécifiques. Il est important d'éviter d'effectuer une action lorsqu'une transaction est occupée. L'Agent tentera toujours d'effectuer une action lorsque la transaction n'est pas occupée. Si la session est encore occupée une fois le délai du temporisateur écoulé, un message Session occupée apparaît. Ces deux méthodes sont utilisées pour la [Extend the Wait Period When the SAP Session is Busy](#). La méthode **setBusyWaitTime** doit être appelée dans la première étape d'un flux d'opérations. Généralement, nous l'utilisons sur la page lors du traitement de la page SAP Easy Access.



SAPLOGON.pEasyAccess.setBusyWaitTime(//temps en millisecondes);

Par défaut, le temps d'attente en cas de session occupée est de 1 seconde (1000 ms) et peut être modifié. Toutes les sessions ouvertes au cours de votre processus utiliseront ce laps de temps pour traiter chaque action effectuée sur les pages SAP GUI.

Chaque action effectuée par le connecteur contrôlera régulièrement à l'aide du temporisateur si la session SAP GUI est occupée ou non.

La meilleure pratique consiste à toujours utiliser **resetBusyWaitTime** lorsque vous terminez votre flux d'opérations.

- **lockRefresh et unlockRefresh**

UIAutomation et le connecteur technique SAPGUI fonctionnent avec le mécanisme d'interrogation. Ils actualisent régulièrement l'état de l'application traitée du point de vue de l'Agent ainsi que son contenu. L'actualisation se fait en demandant ces états directement à l'application (ce qui peut la surcharger). Ces actualisations peuvent prendre du temps pour être traitées, et le mécanisme d'interrogation peut saturer l'application.

Lors de l'exécution d'une action au cours d'une interrogation, une fenêtre temporaire Serveur occupé peut apparaître.

Il n'y a aucun moyen de ralentir le mécanisme d'interrogation avec la technologie GUI (contrairement à UIAutomation).

Ces deux fonctions sont utilisées pour empêcher le connecteur d'actualiser le contenu de la page du point de vue de l'Agent. Cela permet d'avoir le temps d'effectuer une action tout en évitant la saturation de l'application traitée.

Utilisez-les dans un listener d'attente lorsque vous savez que la page est correctement chargée.

```
app.page.wait(function(ev) {  
app.page.lockRefresh();  
// >>> exécuter les actions <<<  
app.page.unlockRefresh();  
});
```

N'oubliez pas unlockRefresh à la fin de l'étape car le connecteur ne verra pas les événements suivants comme l'arrivée d'une nouvelle page.

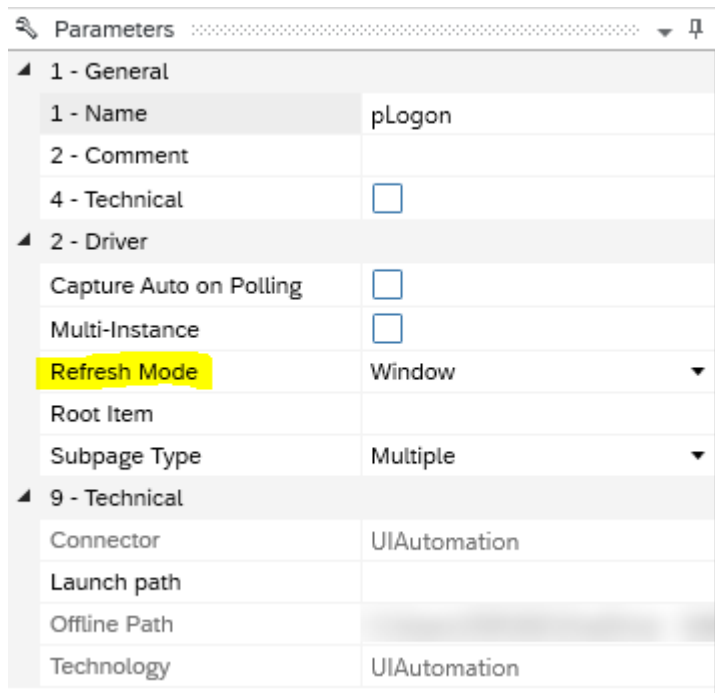
Remarque

L'interrogation automatique peut ralentir chaque action effectuée sur une page, en particulier si cette page ne peut pas être optimisée, si elle contient beaucoup d'éléments, si elle est très profonde ou si elle contient beaucoup d'éléments. Doit exister/Ne doit pas exister.

Côté connecteur

UIAutomation

- Si la fenêtre temporaire **n'est pas déclenchée par SDK** :
 - Essayez d'encadrer la page avec les fonctions lockRefresh et unlockRefresh. Vérifiez si l'erreur Serveur occupé n'arrive pas (voir la procédure décrite ci-dessus à la rubrique lockRefresh et unlockRefresh).
 - Si la fenêtre temporaire Serveur occupé apparaît, attendez quelques secondes pour voir si la fenêtre temporaire disparaît automatiquement et si l'exécution du bot se poursuit. Dans ce cas, ce n'est peut-être pas un bloqueur.
 - Désactivez l'interrogation automatique.
 - Dans Studio, accédez à la page "Paramètres" et définissez le "Mode d'actualisation" sur **Non**.



- Ce mode empêchera l'actualisation automatique de la page. Même si la page est modifiée (nouveau contenu, nouveaux éléments), ces mises à jour ne seront pas automatiquement affichées.
- Appelez `mainpage.refresh` ou `page.refresh` dans l'état de mise à jour du code de l'application du point de vue de l'Agent chaque fois que vous attendez à une mise à jour de la page.

1.2.10 Événements asynchrones dans les boucles For

Parfois, vous devez exécuter certaines actions plusieurs fois dans le flux d'opérations de votre automatisation. La plupart du temps, vous utilisez une boucle For directement dans le code à cet effet. Cette méthode fonctionne bien si vous effectuez des actions simples telles que définir des valeurs dans un fichier Excel ou obtenir des valeurs à partir d'un tableau.

Cependant, si vous effectuez des tâches asynchrones telles que des appels d'API ou que vous attendez des éléments, la boucle For n'attend pas la fin des actions asynchrones pour exécuter la prochaine itération et ne fonctionne donc pas comme prévu. L'itération suivante est exécutée une fois l'action asynchrone exécutée. Par conséquent, l'étape après la boucle For peut être exécutée avant la fin de toutes les actions asynchrones. Si l'étape suivante requiert des données provenant d'événements asynchrones antérieurs, ces données ne peuvent pas être définies à temps.

Appels d'API dans une boucle For

L'exemple suivant illustre le comportement d'exécution d'appels d'API dans une boucle For :

Parfois, vous devez effectuer un appel d'API plusieurs fois. Pour ce faire, commencez par créer votre appel d'API et entourez-le d'une boucle For, puis ajoutez l'étape suivante (par exemple une activité *Journal* simple).

Dans le code suivant, vous pouvez vous attendre à ce que le bot appelle l'API trois fois et imprime le résultat trois fois, puis imprime le message de journal terminé du flux d'opérations.





```

// -----
// Step: Call_a_web_service_1
// -----
GLOBAL.step({ Call_a_web_service_1: function(ev, sc, st) {
  var rootData = sc.data;
  ctx.workflow('forLoop', 'f4762d95-04a7-4cee-9768-081b9cf850b2' );
  for (var i = 0; i < 3; i++) {
    // Calls a web service.
    ctx.ajax.call({
      url: "https://postman-echo.com/get?foo1=bar1&foo2=bar2%27",
      method: e.ajax.method.get,
      data: rootData.test,
      contentType: e.ajax.content.json,
      success: function(res, status, xhr) {
        ctx.log(res.args.foo1);
      },
      error: function(xhr, error, statusText) {
        ctx.log(' ctx.ajax.call error: ' + statusText);
      }
    });
  }
  sc.endStep(); // Write_log_1
  return;
}});

// -----
// Step: Write_log_1
// -----
GLOBAL.step({ Write_log_1: function(ev, sc, st) {
  var rootData = sc.data;
  ctx.workflow('forLoop', 'd250fd23-93c3-448b-b781-f3b5f44a56fa' );
  // Add a message to the log file and in the debug window along with a severity level.
  ctx.log("Workflow finished", e.logIconType.Info);
  sc.endStep(); // end Scenario
  return;
}});

```

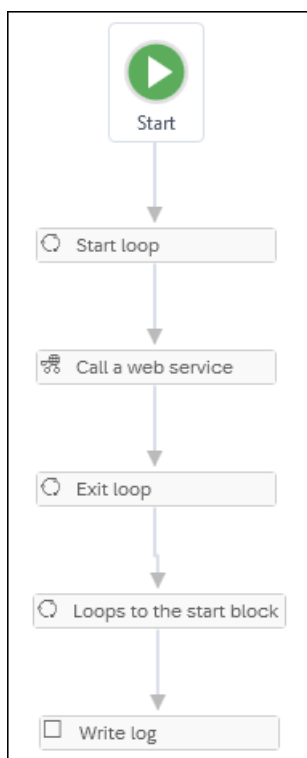
Dans le débogueur, vous constatez que le dernier message de journal est imprimé avant les résultats de tous les appels d'API. Ce comportement est dû à l'utilisation de la boucle For. La boucle For n'attend pas la fin de l'exécution de l'événement asynchrone pour effectuer la prochaine itération. Après toutes les itérations, la méthode `sc.endStep()` est appelée, de sorte que l'étape suivante est exécutée. Cependant, cette dernière étape est exécutée avant que tous les appels d'API aient terminé leur exécution. C'est la raison pour laquelle l'ordre de tous les journaux n'est pas dans le bon ordre dans le débogueur, comme attendu.

	11:25:05.171	Workflow finished
	11:25:05.657	bar1
	11:25:05.667	bar1
	11:25:05.678	bar1

Appels d'API avec des activités en boucle

Pour éviter ce comportement lors de l'exécution de votre bot, vous devez utiliser les activités de boucle disponibles dans Desktop Studio. Ces activités garantissent que l'étape suivante est exécutée une fois qu'un événement asynchrone est terminé.

Vous pouvez ajouter directement ces activités en boucle à l'éditeur de flux d'opérations de Desktop Studio. L'exemple précédent peut être transformé comme suit :



Comme indiqué précédemment, chaque étape est appelée après l'exécution de l'étape précédente. Par conséquent, l'activité *Sortir de la boucle* est exécutée une fois que l'activité *Appeler un service Web* a terminé son exécution et obtient le résultat de la part de l'API.

Dans le débogueur, vous constatez que l'ordre de tous les journaux est correct, comme attendu.

11:43:47.617	bar1
11:43:47.692	bar1
11:43:47.761	bar1
11:43:47.815	Workflow finished

Ainsi, l'implémentation des activités en boucle garantit que toutes les étapes sont exécutées dans le bon ordre.

Activité d'attente dans une boucle

L'exemple suivant illustre le comportement d'exécution d'une activité d'attente dans une boucle :

Vous devez parfois attendre un certain temps avant d'exécuter l'activité suivante et le même comportement se produit, comme indiqué ci-dessus dans la section *Appels d'API dans une boucle For*.

Dans le code suivant, vous constatez que la même erreur se produit, comme dans la section *Appels d'API dans une boucle For*.

```

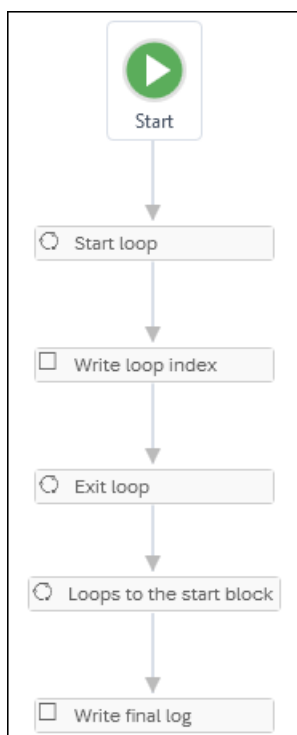
// -----
// Step: Custom
// -----
GLOBAL.step({ Custom: function(ev, sc, st) {
  var rootData = sc.data;
  ctx.workflow('waitForLoop', '629af868-dbf2-4a6c-8669-b6cfc7022321') ;
  // Describe functionality to be implemented in JavaScript later in the project.
  for (var i = 0; i < 3; i++) {
    ctx.wait(function(ev) {
      ctx.log(i);
    }, 1000);
  }
  sc.endStep(); // Write_log_2
  return;
}});

// -----
// Step: Write_log_2
// -----
GLOBAL.step({ Write_log_2: function(ev, sc, st) {
  var rootData = sc.data;
  ctx.workflow('waitForLoop', '5d5abf57-56a5-473b-9c9a-2647c7a5dab6') ;
  // Add a message to the log file and in the debug window along with a severity level.
  ctx.log("Workflow finished", e.logIconType.Info);
  sc.endStep(); // end Scenario
  return;
}});

```

Activité d'attente avec des activités en boucle

Pour exécuter toutes les étapes dans le bon ordre, vous devez modifier le code précédent et utiliser les activités en boucle comme suit :



Par conséquent, l'implémentation des activités en boucle garantit que tous les journaux d'index sont imprimés avant le journal final.

1.3 Cloud Factory

Les notes de base de connaissances suivantes se rapportant à Cloud Factory sont disponibles :

- "Impossible de créer le package. La version de package existe déjà pour 'NomProjet'". Erreur lors de l'importation d'un projet dans Factory - [2792210](#) ;
- "Impossible de charger Factory - insufficient_scope : périmètre insuffisant pour cette ressource" - [2802932](#) ;
- Aucune action lors du lancement d'un scénario non surveillé avec Desktop Agent - [2796491](#).
- Comment masquer les informations de machine et de connexion : [3056143](#)

1.3.1 Périmètre insuffisant

L'erreur "périmètre insuffisant" se produit lors que vous essayez d'accéder au service Factory de SAP Intelligent RPA.

Contexte

Ce message d'erreur signifie que le processus d'intégration à SAP Intelligent RPA n'a pas été totalement terminé. Les rôles appropriés n'ont pas été configurés et affectés.

Pour résoudre ce problème, vous devez ajouter des rôles et des utilisateurs à ces rôles.

Configurer les rôles

1. Dans l'onglet *Sécurité* de votre sous-compte, accédez à *Collections de rôles*.

ⓘ Remarque

Vous devez disposer du rôle [Administrateur de sécurité](#) pour accéder à l'onglet *Sécurité* dans le cockpit de SAP BTP (Business Technology Platform).

The screenshot shows the SAP Cloud Platform Cockpit interface. On the left, a navigation menu is visible with the following items: Overview, Spaces, Subscriptions, Connectivity, Security (highlighted with a red box), Administrators, Role Collections (highlighted with a red box), Trust Configuration, Quota Plans, and Members. The main content area displays the 'Subaccount: Trial-trial - Role Collections' page. At the top, there is a 'New Role Collection' button. Below it is a table with columns for 'Name', 'Description', and 'Roles'. The table currently shows one entry: 'No Role Collection'. At the bottom of the page, there is a link: 'Learn more about building roles and maintaining role collections.'

2. Créez une *Nouvelle collection de rôles*.

New Role Collection

*Name:

Description:

Save **Cancel**

3. Cliquez sur la collection de rôles que vous venez de créer et sélectionnez *Ajouter un rôle*.

New Role Collection

Name

Add Role

Application Identifier

4. Dans la liste déroulante *Identifiant d'application*, sélectionnez le rôle *IRPAOfficer* pour créer l'administrateur du locataire.

Add Role

*Application Identifier:

*Role Template:

*Role:

Save **Cancel**

5. Déconnectez-vous et reconnectez-vous pour que vos nouveaux rôles soient pris en compte.

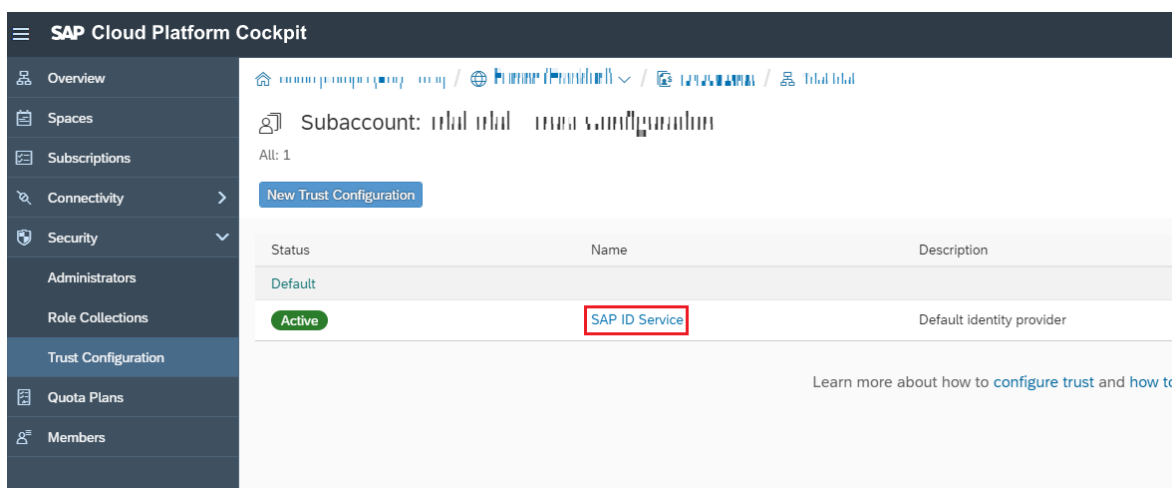
Affecter des rôles aux utilisateurs

ⓘ Remarque

Répétez cette opération pour :

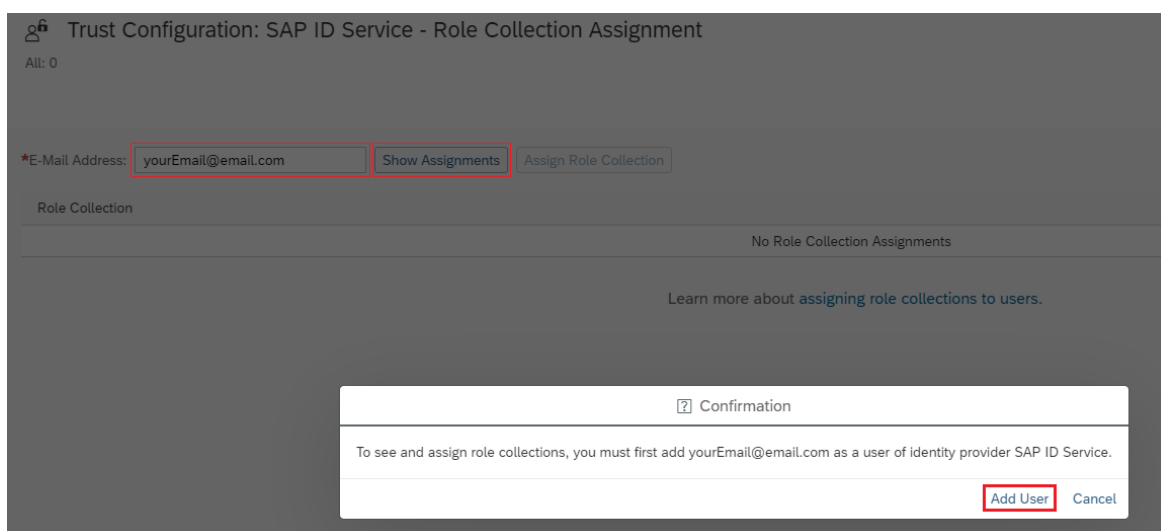
- Les utilisateurs qui souhaitent configurer des bots dans Cloud Factory : ajoutez le rôle *IRPAOfficer* à ces utilisateurs.
- Les utilisateurs qui souhaitent exécuter des bots : ajoutez le rôle *IRPAAgentUser* (assurez-vous que ce rôle est configuré dans la collection de rôles).

1. Accédez à *Configuration de confiance* et cliquez sur le service SAP ID.



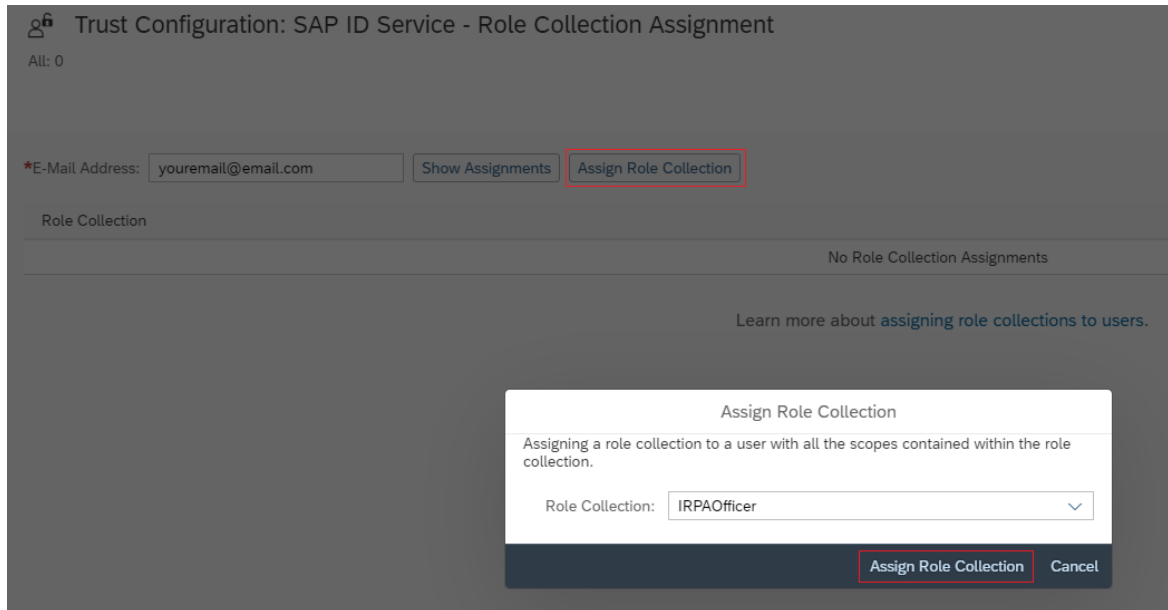
The screenshot shows the SAP Cloud Platform Cockpit interface. The left sidebar contains navigation options: Overview, Spaces, Subscriptions, Connectivity, Security (expanded), Administrators, Role Collections, Trust Configuration (selected), Quota Plans, and Members. The main content area displays the 'Subaccount: [id]' and 'All: 1' information. A 'New Trust Configuration' button is visible. Below it is a table with columns 'Status', 'Name', and 'Description'. The table contains one entry: 'Default' with status 'Active' and name 'SAP ID Service' (highlighted with a red box), described as 'Default identity provider'. A link 'Learn more about how to configure trust and how to...' is at the bottom right.

2. Saisissez l'adresse e-mail de votre utilisateur et cliquez sur *Afficher affectation*. Si nécessaire, cliquez sur *Ajouter utilisateur*.



The screenshot shows the 'Trust Configuration: SAP ID Service - Role Collection Assignment' dialog. It has a dark background. At the top, it says 'All: 0'. There is an input field for '*E-Mail Address:' containing 'yourEmail@email.com' (highlighted with a red box). To its right are buttons for 'Show Assignments' (highlighted with a red box) and 'Assign Role Collection'. Below the input field is a section for 'Role Collection' which currently shows 'No Role Collection Assignments'. At the bottom, there is a 'Confirmation' dialog box with the text: 'To see and assign role collections, you must first add yourEmail@email.com as a user of identity provider SAP ID Service.' It has two buttons: 'Add User' (highlighted with a red box) and 'Cancel'.

3. Cliquez sur *Affecter collection de rôles* et sélectionnez le rôle requis.



1.4 Cloud Studio

- Certaines activités ne sont plus modifiables après la mise à niveau inférieur d'un SDK - [3025037](#).
- Comment configurer Outlook pour utiliser un SDK Outlook pour SAP Intelligent RPA - [3048085](#).
- Procédure si l'activité Définir la valeur de la variable ne fonctionne pas avec les sous-propriétés d'objet - [3145604](#).

1.4.1 Mappage des fonctionnalités de Cloud Studio et Desktop Studio

Les fonctionnalités de Desktop Studio sont disponibles dans Cloud Studio avec la même nomenclature ou une nomenclature différente. Les fonctionnalités ayant des fonctionnalités avancées ou similaires ont la même finalité que dans Desktop Studio.

Contexte

Créer dossier

Dans Desktop Studio, le contexte est le gestionnaire de données et est utilisé pour gérer toutes les données générées lors de l'exécution d'un scénario. De la même manière, la même fonctionnalité peut être trouvée dans Cloud Studio avec une nomenclature différente comme [Paramètres d'entrée/de sortie](#).

Dans Desktop Studio, **Contexte>Créer dossier** = dans Cloud Studio, [Paramètres d'entrée/de sortie](#).

Vous pouvez avoir le type de données Liste. Dans Cloud Studio, vous devez cocher la case *Liste* pour que le type soit présenté sous forme de tableau (par exemple une liste).

Pour en savoir plus sur la gestion des données, voir la section [Gérer les données dans une automatisation](#).

Créer élément

Dans Desktop Studio, **Contexte**>**Créer élément** = dans Cloud Studio, [Types de données](#) (prédéfini ou éditeur de type de données).

1.4.2 Définition du contrôle Pour chaque pour des éléments sans rectangle englobant

Lorsque vous capturez des éléments qui n'ont pas de rectangle englobant, vous ne pouvez pas utiliser le contrôle *Pour chaque*.

Contexte

Lorsque vous capturez un écran contenant des éléments récurrents tels que plusieurs liens, vous pouvez utiliser le contrôle *Pour chaque*. Le contrôle *Pour chaque* s'applique à tous les éléments récurrents.

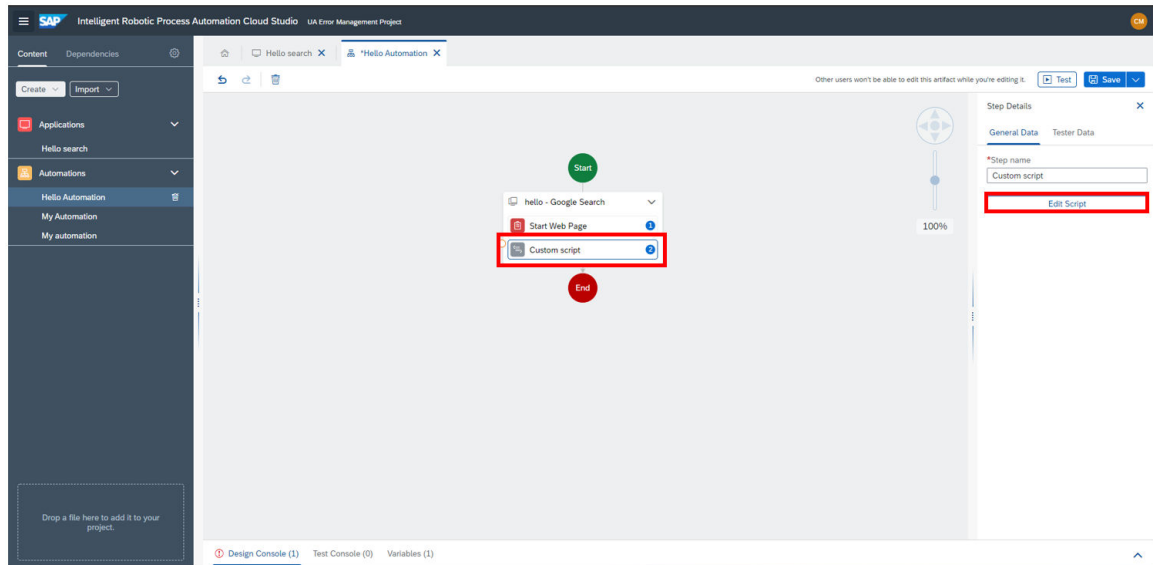
Si vous ajoutez des activités au contrôle *Pour chaque* contrôle, elles s'appliqueront à tous les éléments récurrents associés.

Lorsque vous travaillez sur des écrans capturés, les éléments capturés ne sont parfois pas visibles car ils n'ont pas de rectangle englobant. Par conséquent, vous ne pouvez pas utiliser le contrôle *Pour chaque*.

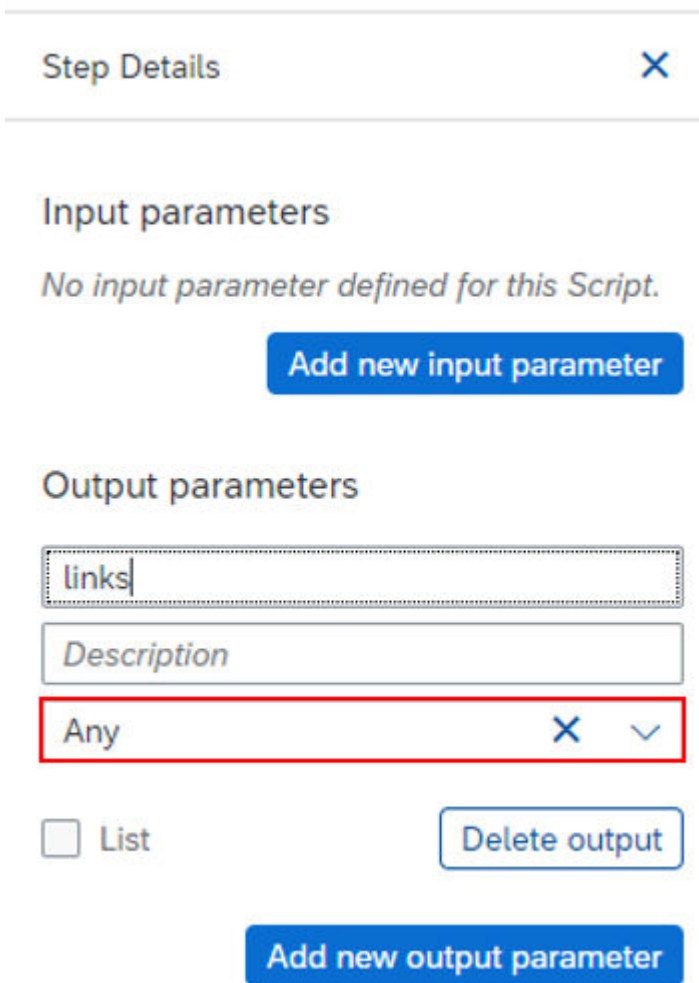
Pour résoudre cette limitation, vous pouvez utiliser l'activité *Script personnalisé*.

Définir le contrôle Pour chaque pour des éléments sans rectangle englobant

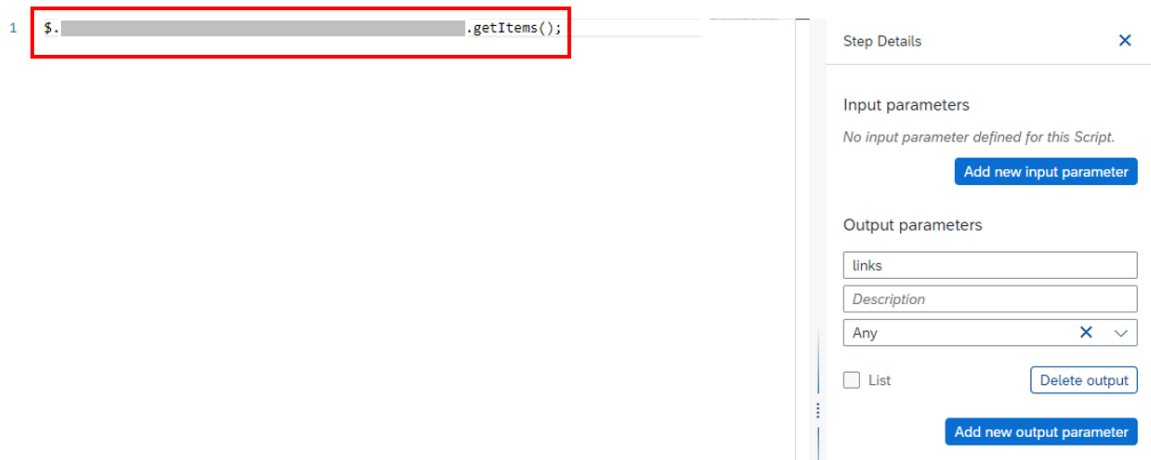
1. Dans le panneau latéral *Informations d'automatisation*, glissez-déplacez une activité *Script personnalisé* dans votre automatisation.
2. Dans le panneau latéral *Détails de l'étape*, cliquez sur *Modifier le script*.



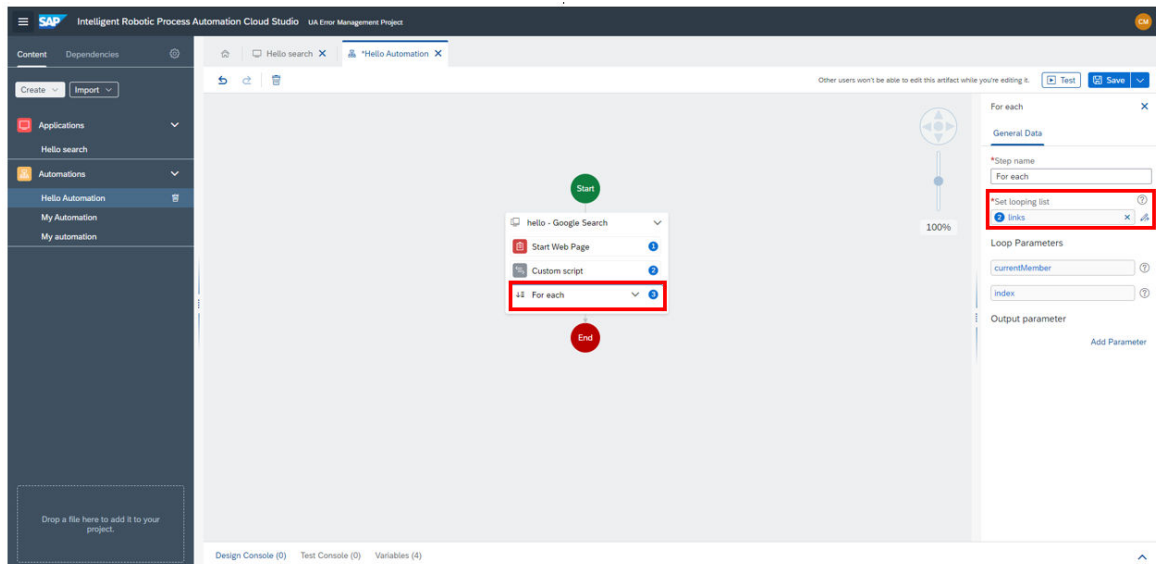
3. Sélectionnez une sortie de type *N'importe*.



4. Dans le panneau central, ajoutez la fonction `getItems` à la fin du chemin de tous les éléments masqués (car ils n'ont pas de rectangle englobant).



5. Ajoutez le contrôle *Pour chaque* à votre automatisation.
6. Dans le panneau latéral *Pour chaque*, sélectionnez comme entrée la sortie précédemment définie dans le champ *Définir la liste de boucles*.



Vous pouvez savoir utiliser le contrôle *Pour chaque*.

Pour en savoir plus, voir [For Each](#).

1.4.3 Déverrouillage d'une session verrouillée pendant l'exécution de l'automatisation

Lorsque votre bot est exécuté en mode non surveillé, vous devez parfois déverrouiller la session pour effectuer des activités. Par exemple, la session doit être déverrouillée pour exécuter des activités de frappe de clavier et de clic de souris. Si vous avez défini vos informations d'identification Windows dans Desktop Agent, la session est automatiquement déverrouillée et reverrouillée lorsque des activités de ce type doivent être effectuées.

ⓘ Remarque

Il est recommandé d'utiliser l'automatisation de l'IU dans les scénarios où la machine n'est pas débloquée.

Pour vous assurer que la session se déverrouille et se verrouille, vous devez d'abord configurer vos informations d'identification Windows dans Desktop Agent. Pour en savoir plus sur le stockage de vos informations d'identification Windows, référez-vous à [Définition du mot de passe Windows dans Desktop Agent](#).

Desktop Agent utilise vos informations d'identification pour déverrouiller automatiquement la session. Une fois l'activité exécutée, la session se verrouille automatiquement 10 secondes après l'action.

Pour en savoir plus, voir [Configuration d'un bot non surveillé](#).

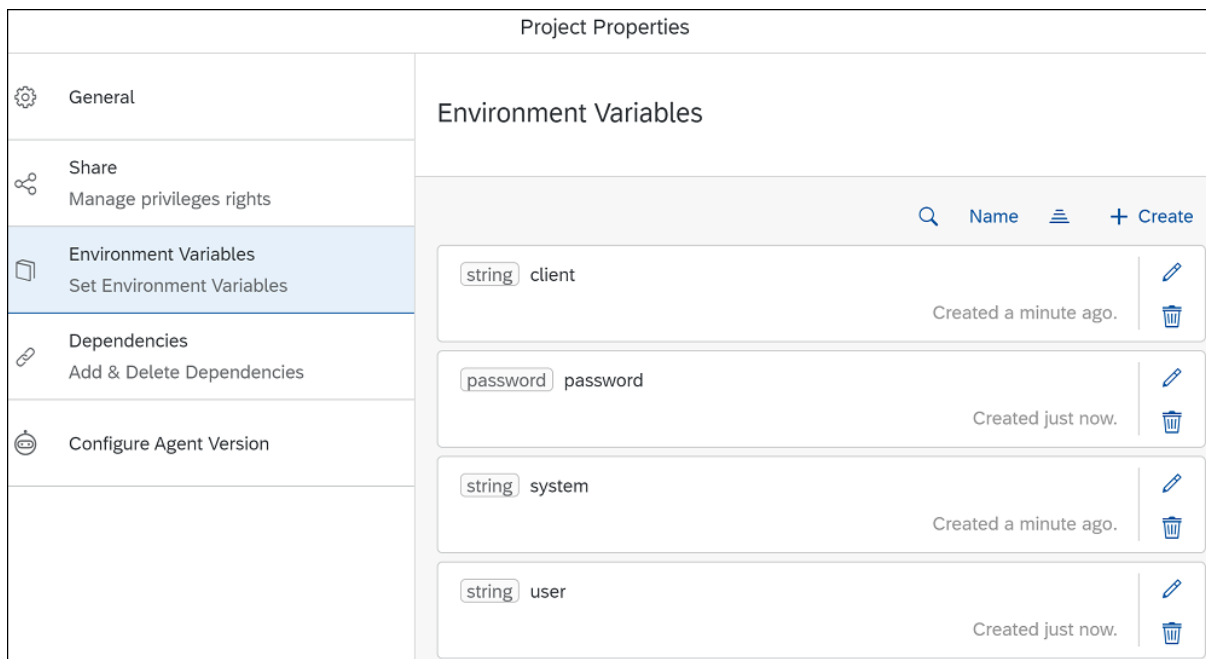
1.4.4 Application d'IU en cours d'exécution avec des privilèges plus importants

Dès que vous disposez d'une application qui s'exécute avec des privilèges plus importants que ceux de Desktop Agent (mode Admin), l'agent ne peut pas interagir avec l'application. Il pourra effectuer des activités en lecture seule telles que la mise en surbrillance, l'obtention de la valeur d'un champ, etc., mais il ne pourra pas effectuer d'activités telles que cliquer sur un élément, définir une valeur pour un champ, etc. Par conséquent, vous rencontrerez des problèmes techniques si votre application s'exécute avec un privilège plus important, tel que le mode "Admin".

1.4.5 Lancement du SAP Logon en mode non surveillé

Prérequis

Vous avez créé des variables d'environnement dans un projet, comme il est décrit dans [Variables d'environnement](#).



Vous avez créé une ou plusieurs automatisations dans SAP Build Process Automation, comme décrit dans [Création d'une automatisation](#).

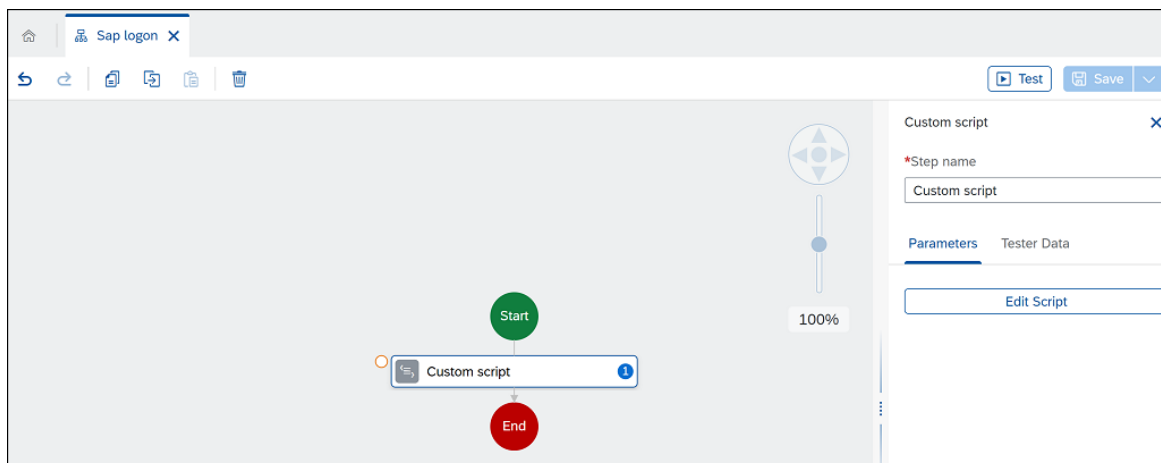
Contexte

Vous pouvez lancer l'application SAP Logon avec un bot non surveillé à l'aide de l'activité de script personnalisé, des variables d'environnement et du déclencheur planifié.

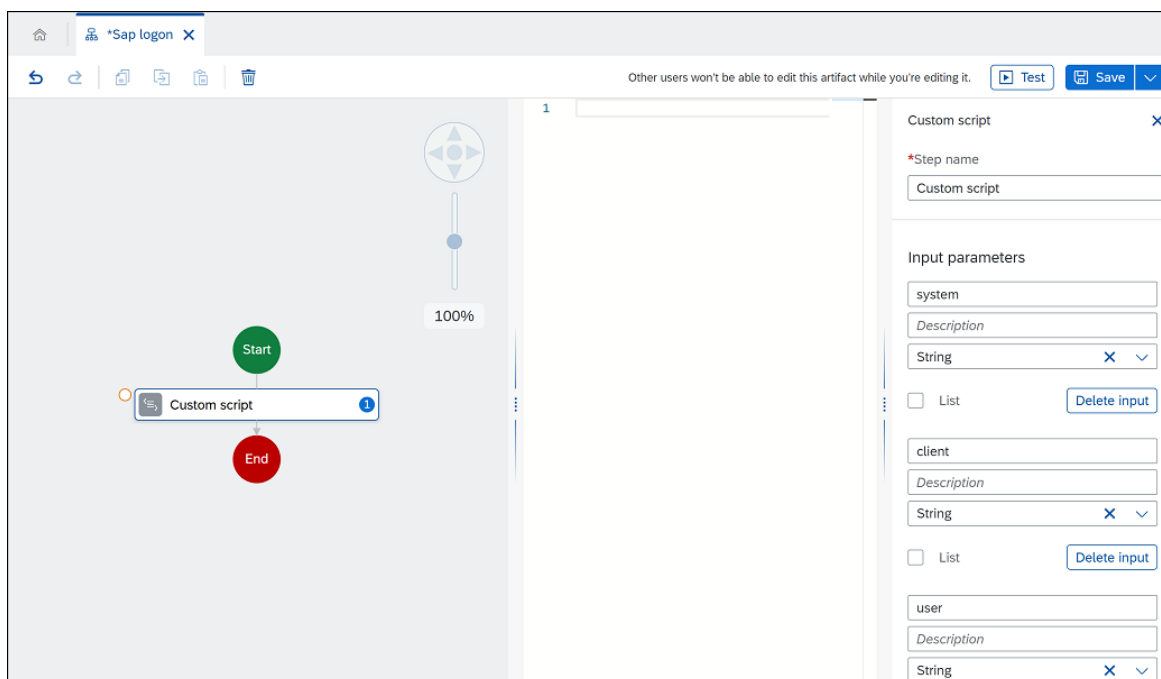
Le cas d'utilisation suivant montre comment lancer l'application SAP Logon en mode non surveillé.

Procédure

1. Dans votre automatisation, dans le panneau latéral de droite, accédez à la section [Commandes](#).
2. Glissez-déplacez l'activité [Script personnalisé](#) dans votre flux d'opérations.



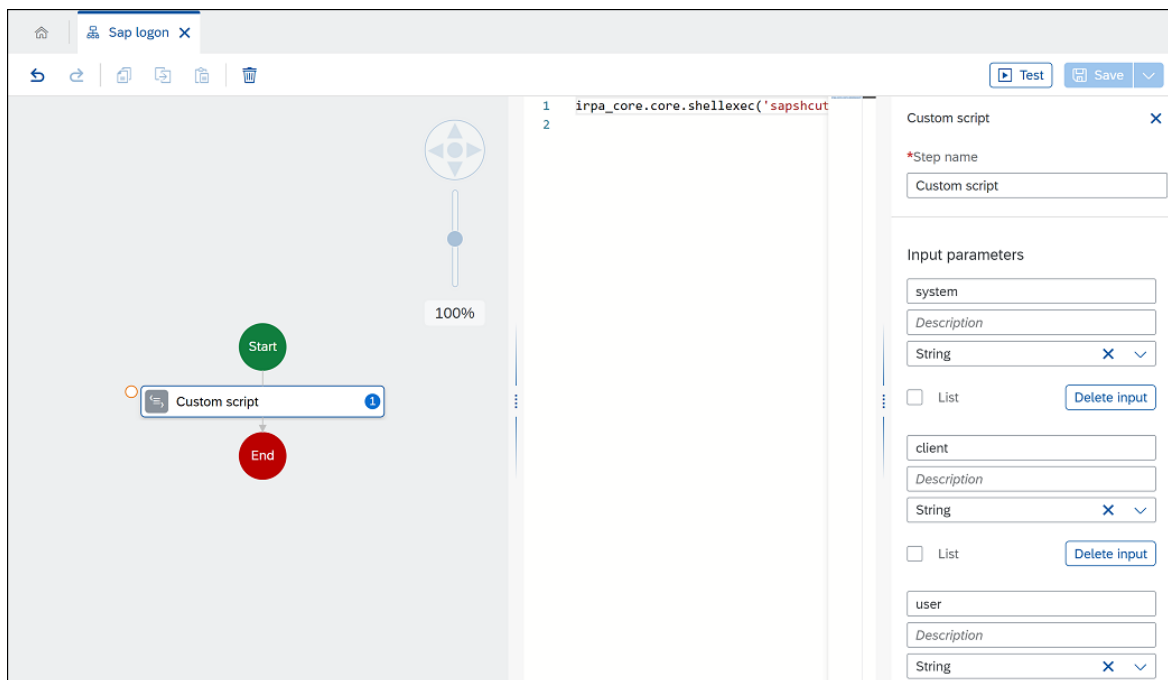
- Sélectionnez *Script personnalisé* et cliquez sur *Modifier le script*.



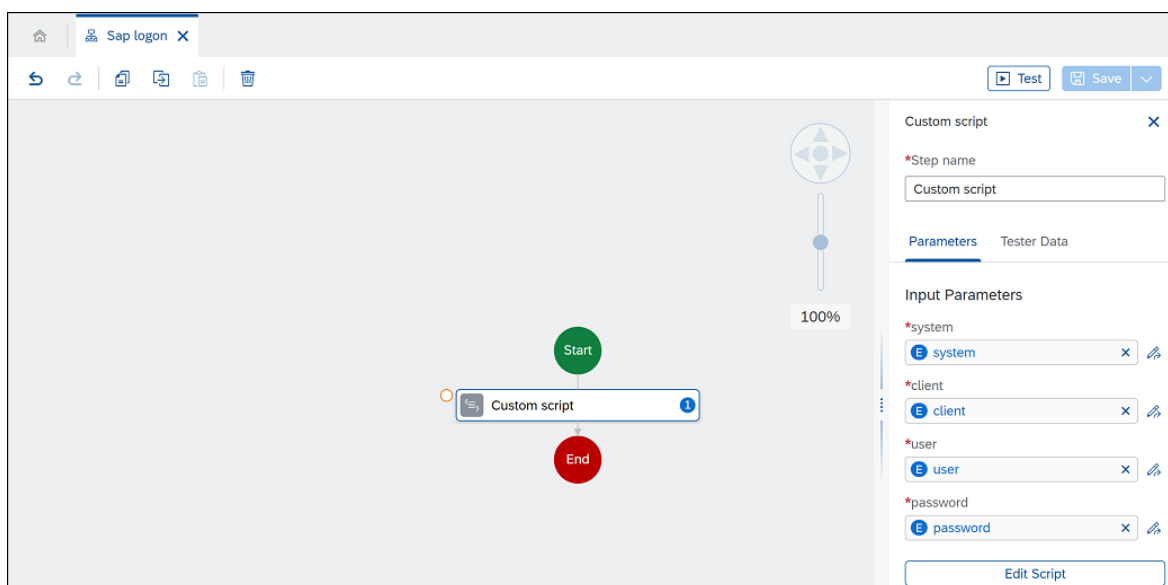
- Ajoutez les variables que vous souhaitez utiliser dans le script afin que, lorsque vous écrivez le script, la liaison entre ces variables et les valeurs de saisie soit effectuée. Vous devez utiliser le *système*, le *mandant*, l'*utilisateur* et le *mot de passe* comme valeurs de saisie. Pour la valeur de saisie *mot de passe*, indiquez le type "mot de passe" afin de pouvoir protéger votre mot de passe.
- Écrivez la ligne suivante dans le script :

```
irpa_core.core.shellexec('sapshcut.exe', '-system=' + system + ' -client=' + client + ' -user=' + user + ' -pw=' + password);
```

Cette ligne lance le SAP Logon et s'y connecte avec le système, le mandant, l'utilisateur et le mot de passe que vous avez indiqués. Vous concaténez vos variables dans la ligne qui est exécutée sur un shell par l'automatisation.



6. Ajoutez les variables à l'activité *Script personnalisé*. Pour chaque paramètre d'entrée, écrivez le nom de la variable, puis cliquez sur la variable qui suggère le E bleu avant. Cliquez sur la croix de fermeture pour fermer l'onglet de modification.

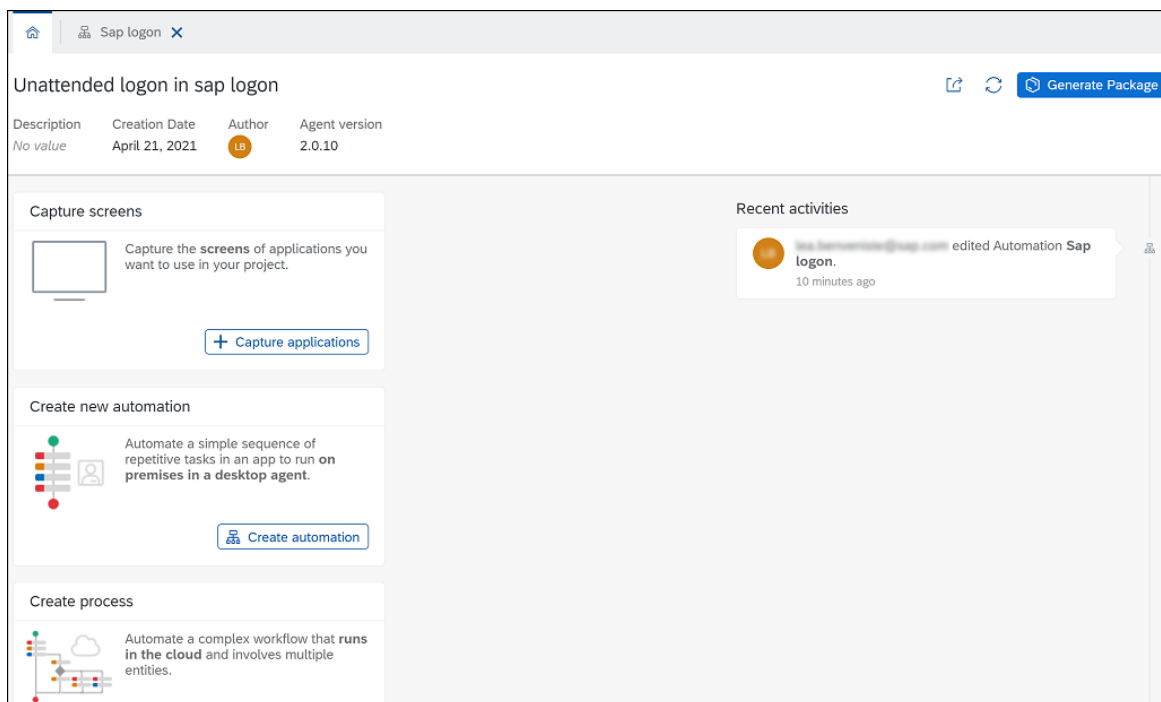


Résultats

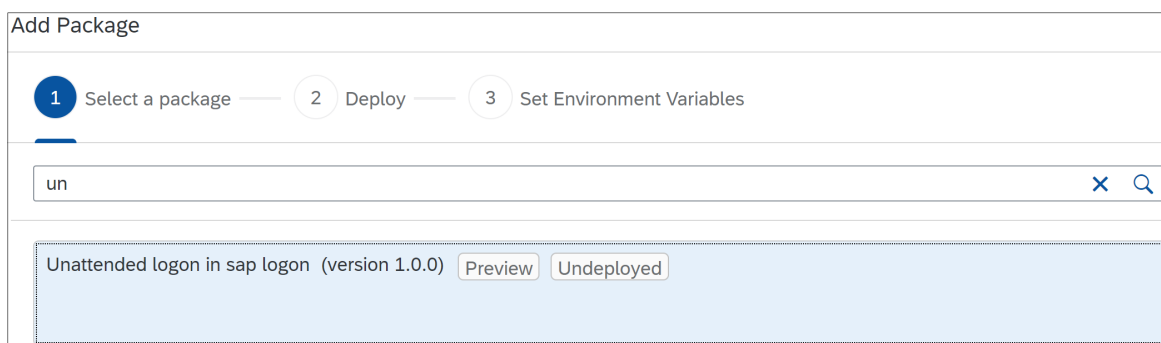
Désormais, l'automatisation de la connexion SAP est terminée et vous pouvez configurer l'environnement pour lancer cette automatisation en mode non surveillé.

Étapes suivantes

1. Maintenant que vous avez terminé la conception de votre projet, vous devez créer un package pour ajouter le projet à votre environnement. Pour en savoir plus, voir [Génération de packages](#).



2. Une fois le package généré, ajoutez-le à votre environnement. Pour en savoir plus, voir [Ajout de packages à un environnement](#).



Vous pouvez également définir des valeurs pour les variables d'environnement lors du déploiement d'un package dans l'environnement. Pour en savoir plus, voir [Variables d'environnement de Cloud Studio](#).

Add Package

1 Select a package — 2 Deploy — **3 Set Environment Variables**

i This step is not mandatory to create a trigger. However some triggers may require information to be properly executed.

Unattended logon in sap logon Version 1.0.1

password in Unattended logon in sap logon 1.0.1 Data type: password

password:

user in Unattended logon in sap logon 1.0.1 Data type: string

Set new value Use existing value

string:

client in Unattended logon in sap logon 1.0.1 Data type: string

Set new value Use existing value

string:

Previous **OK** Cancel

3. Ajoutez un déclencheur pour définir comment et quand l'automatisation sera exécutée. Pour en savoir plus, voir [Ajout de déclencheurs à un environnement](#).

Create Trigger

1 Select a package — 2 Deploy — 3 Set Environment Variables — 4 Select a trigger type

Search...

Unattended logon in sap logon (version 1.0.1) Updated 2 minutes ago.

Comme vous souhaitez exécuter l'automatisation en mode non surveillé, vous devez sélectionner le type de déclencheur *Planifié*. Pour en savoir plus, voir [Ajout d'un déclencheur planifié](#).

Create Trigger


1 Select a package — 2 Deploy — 3 Set Environment Variables — **4 Select a trigger type**

Lea - Unattended logon in sap logon Version 1.0.1

API
A trigger of type API opens a dedicated endpoint that allows an external application to execute a scenario or a process.

Attended
With an attended trigger, the deployed package is distributed to a specific group of agents, and users run the jobs manually.

Scheduled
With a scheduled trigger, the user does not need to start a job manually. Jobs are created according to the schedule you define in the trigger.

4. Une fois que vous avez finalisé la configuration de votre déclencheur, cliquez sur le bouton  (Plus d'options...) sur la vignette du déclencheur, puis sélectionnez *Exécuter maintenant*.

ⓘ Remarque

Vous devez sélectionner votre agent en mode *Non surveillé* directement dans le systray de l'agent. La première fois que vous lancez le bot, la fenêtre temporaire suivante peut s'afficher.

SAP GUI Security

SAP GUI connections are started via the connection string:

[Redacted connection string]

The connection data is:

System: *[Redacted]*
Client: *[Redacted]*
OK code:

Do you want to log on to this system?

Remember My Decision

Vous pouvez cocher la case *Mémoriser ma décision* et cliquer sur le bouton *Autoriser* pour ne plus afficher la fenêtre temporaire.

1.4.6 Meilleures pratiques pour l'optimisation d'un élément

Cette section décrit les meilleures pratiques pour l'optimisation d'un élément.

Introduction

L'automatisation de tout type d'application (Web, UIAutomation et bien d'autres encore) ne peut se faire correctement qu'avec une bonne reconnaissance des pages et des éléments.

ⓘ Remarque

La reconnaissance de votre application et de votre élément est aussi importante que le développement de votre flux d'opérations. Si la reconnaissance n'est pas effectuée correctement, l'automatisation risque de ne pas fonctionner correctement.

Cela inclut notamment :

- Utiliser des critères les plus uniques possibles pour chaque page et élément.
- S'assurer que l'élément est considéré comme unique par le connecteur (à l'aide du débogueur, par exemple).

Parfois, ce n'est pas suffisant, certains effets secondaires peuvent être observés pendant l'automatisation.

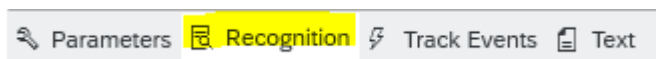
Par exemple, si le message d'erreur suivant s'affiche "Dépassement du délai d'attente pour la réponse du pilote", cela signifie que vous devez optimiser la reconnaissance de l'élément cible. Techniquement parlant, le connecteur envoie l'action à l'élément mais celui-ci prend trop de temps pour donner une réponse.

Onglet Reconnaissance

Dans SAP Intelligent RPA 1.0, il est possible de voir le nombre de tests effectués par le connecteur pour reconnaître une page ou un élément particulier.

Ces tests peuvent nous aider à savoir si une optimisation doit être effectuée ou non.

Dans Desktop Studio, du côté de la reconnaissance d'applications, regardez à droite des paramètres, vous devriez voir le bouton "Reconnaissance".



En cliquant sur chaque page d'une application, vous verrez le nombre de tests effectués pour reconnaître chaque élément.

Comme vous pouvez le voir ci-dessous, certains des éléments requièrent plus de 4000 tests avant d'être reconnus.

- > ♥ [4111 tests].
- ♥ [4280 tests].
- > ♥ [4616 tests].
- > ♥ [4616 tests].
- > ♥ [4616 tests].
- > ♥ [4616 tests].
- > ♥ [4616 tests].
- > ♥ [4616 tests].
- > ♥ [4616 tests].
- > ♥ [4616 tests].
- > ♥ [4616 tests].
- > ♥ [4616 tests].
- ♥ [3044 tests].

Nombre de tests

Ce que nous voulons, c'est avoir le moins de tests possible et atteindre 1 test si possible.

Pour ce faire, nous devons optimiser la reconnaissance avec une fonctionnalité de reconnaissance avancée.

ⓘ Remarque

Selon le connecteur, il n'est pas toujours nécessaire de procéder à une optimisation lorsque vous voyez un grand nombre de tests.

Web : Ce connecteur est rapide, 500 – 1000~ ne représente pas un grand nombre de tests, l'automatisation peut fonctionner rapidement et correctement.

SAPGUI : Lors de l'utilisation d'un ID/nom comme critère pour un élément (avoir un chiffre comme 3000 tests n'est pas un problème dans ce cas car le connecteur bénéficie déjà d'une optimisation interne).

Il n'y a pas de "bonne" plage de valeurs pour le nombre de tests, tout dépend de la technologie. Comme nous l'avons vu plus haut pour la technologie Web, il est possible d'avoir 1000 tests et que tout fonctionne correctement, mais pour UIAutomation ce serait trop.

Pour toute autre situation, l'utilisation de la reconnaissance avancée doit être implémentée.

Ce n'est pas obligatoire du tout, mais si vous constatez une automatisation lente par exemple et que vous souhaitez la rendre plus performante, optimiser la reconnaissance est probablement le meilleur moyen d'y parvenir.

ⓘ Remarque

: Il existe un compromis entre le fait d'avoir moins de tests et le nombre de critères utilisés pour reconnaître un élément.

Par exemple, vous aurez un nombre de tests peu élevé en utilisant la [Items Pattern Method](#) Cependant, en procédant ainsi, si le DOM de l'application est modifié, vous devrez à nouveau mettre à jour les critères.

Le développeur doit en prendre connaissance avant tout développement.

Nombre d'éléments dans une page

L'optimisation est également liée au nombre d'éléments reconnus dans votre page.

Il n'existe pas de nombre maximum d'éléments par page, mais la bonne pratique consiste à avoir uniquement des éléments utiles :

- Interaction avec des entrées pendant l'automatisation
- Éléments utilisés pour la reconnaissance avancée (Ancêtre, Doit exister, Ne doit pas exister et bien d'autres encore)

Gardez ce nombre aussi peu élevé que possible car le connecteur testera chaque élément dans l'ordre.

Si le connecteur teste un élément inutile avant l'élément avec lequel vous voulez interagir, vous perdrez un temps précieux.

Conclusion

Comme nous l'avons vu ensemble, lorsque nous parlons d'optimisation de la reconnaissance, vous devez consulter le panneau de la reconnaissance.

Il n'y a pas de "bon" nombre de tests, les connecteurs ne fonctionnent pas tous de la même manière en ce qui concerne la reconnaissance.

Si un élément doit être optimisé, suivez les méthodes de reconnaissance avancées.

La meilleure pratique consiste à conserver uniquement les éléments utiles. Le fait de conserver un élément qui n'est pas utilisé pendant l'automatisation ou qui n'est pas utilisé pour reconnaître un autre élément ralentira l'ensemble du processus de reconnaissance de votre page.

Désormais, vous en savez plus sur l'optimisation et la manière de vérifier si une application est bien reconnue.

Vous trouverez de plus amples informations sur le panneau de la reconnaissance en suivant ce lien vers la documentation officielle.

1.4.7 Fonctionnement des tests

Chaque fois que vous effectuez un test local avec Cloud Studio et Desktop Agent, plusieurs choses se passent en arrière-plan et peuvent prendre un certain temps pendant le test initial :

- L'agent doit redémarrer en "mode test" pour activer le canal de communication approprié entre **Cloud Studio**, l'**extension de navigateur** et **Studio Bridge**.
- Les packages SDK ajoutés dans votre projet sont :
 - téléchargés depuis Cloud Studio ;
 - décompressés ;
 - chargés dans la mémoire de l'agent.
- Les packages dépendants (le cas échéant) sont téléchargés, décompressés et chargés si nécessaire.

- Le projet lui-même est téléchargé, décompressé et chargé dans la mémoire de l'agent.
- Lorsque le contenu est téléchargé à partir de Cloud Studio et traité dans Desktop Agent, les ressources peuvent également être contrôlées avec des processus antivirus ou anti-logiciels malveillants réguliers en arrière-plan.

Lorsque vous appuyez sur le bouton de test sans apporter de modifications au projet, toutes les ressources sont déjà dans Desktop Agent et le test est lancé instantanément.

À partir de la version 2.0.20 de l'agent, des améliorations ont été apportées :

- Avant cette version, toute modification des parties prenantes du projet exige que l'agent redémarre entièrement.
- Avec cette version, une modification d'une automatisation ne nécessitera pas la relance complète de l'agent. Le temps d'attente entre deux tests sera considérablement réduit.

Toutefois, pour garantir la stabilité et l'intégrité du projet :

- Si plus de dix artefacts ont été mis à jour depuis la dernière exécution, l'agent redémarrera entièrement.
- Les modifications d'autres artefacts (comme la déclaration d'application) nécessiteront toujours le redémarrage de l'agent.

1.4.8 Exécution d'une automatisation à partir d'une autre automatisation

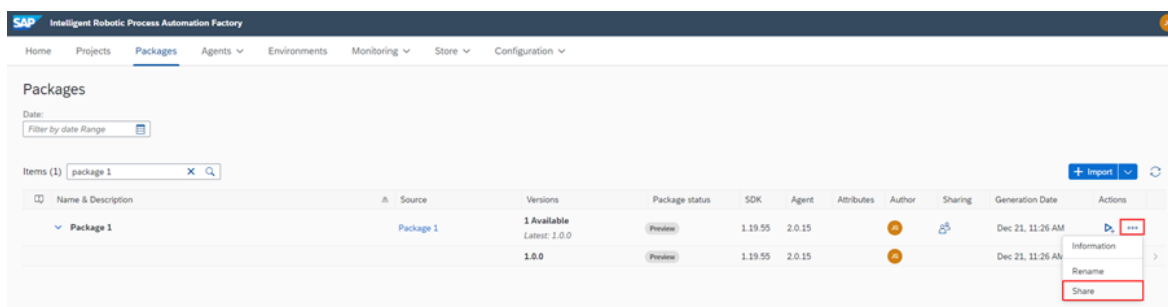
Vous pouvez exécuter une automatisation donnée lors de l'exécution d'une autre automatisation dans Cloud Studio. Il existe deux possibilités pour cela, soit à l'aide d'une dépendance directe, soit à l'aide d'un déclencheur API.

Utilisation d'une dépendance directe

Il s'agit de l'option recommandée car elle est la plus simple à implémenter.

Par exemple, supposons que vous vouliez exécuter 'automatisation 1' à partir du 'package 1' dans 'automatisation 2'.

1. Accédez à l'onglet [Packages](#) dans Cloud Factory.
2. Recherchez votre 'package 1' et partagez-le en cliquant sur **...**.

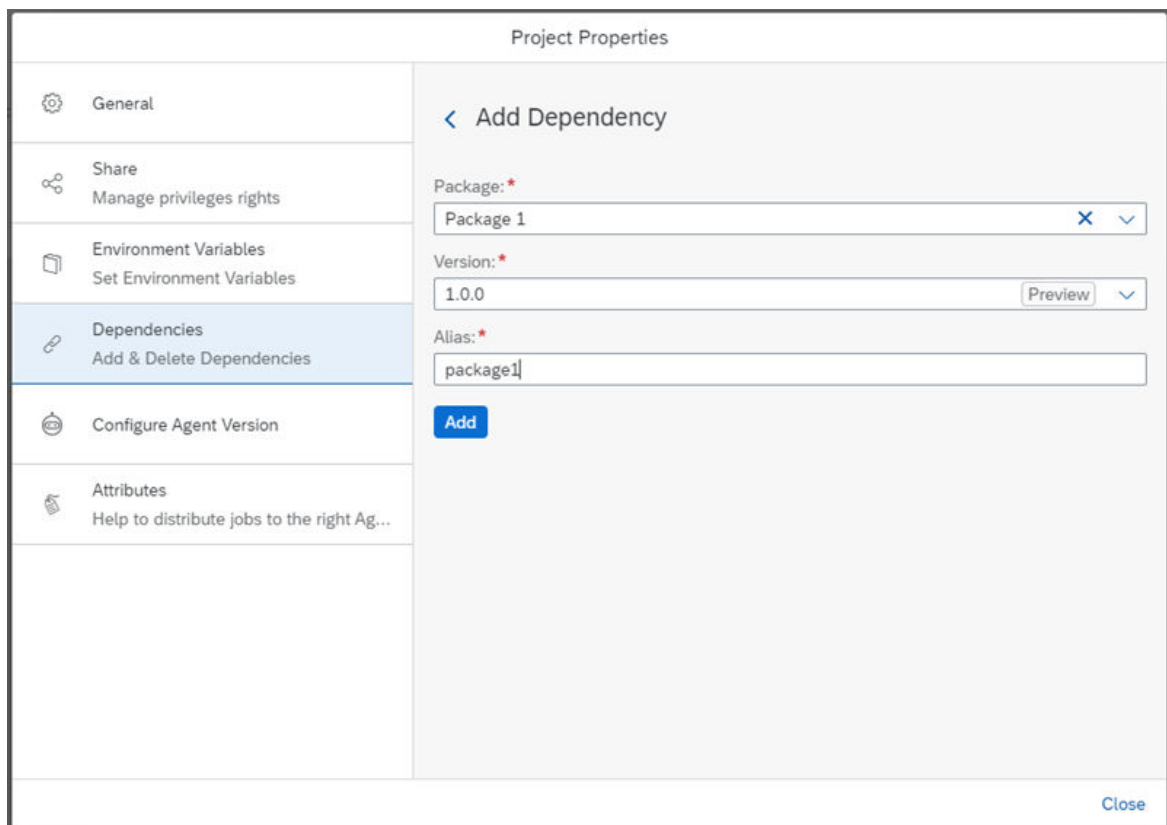


Remarque

Vous devez partager votre package avec *N'importe qui*, en sélectionnant *Lire* comme autorisation, comme affiché ci-dessous :




- Ouvrez le 'package 2' dans Cloud Studio et ajoutez le 'package 1' comme dépendance. Assurez-vous de sélectionner la bonne version de package.








- Le premier package est désormais disponible pour l'exécution dans la liste des automatisations de votre projet. Vous pouvez alors décider d'exécuter les automatisations de manière synchrone ou de les exécuter ultérieurement.


Automation Details

Tools Input/Output Info

Search... 

Automation 1 package1 

Automation 2 

> Activities

> Data

> Controls

Pour en savoir plus, voir la page dans [Dépendances](#).

Utilisation d'un déclencheur API

Cette option est plus complexe. Cette méthode fonctionne en ajoutant une nouvelle tâche à la file d'attente dans Cloud Factory. Ainsi, elle est exécutée ultérieurement dans votre environnement, mais il est impossible de l'exécuter de manière synchrone.

Vous devez créer un déclencheur API spécifique, puis effectuer un appel à un service Web pour exécuter l'automatisation.



Pour en savoir plus, voir la page sur la [manière d'ajouter un déclencheur API](#) et [Exécution d'un déclencheur API](#).

Clauses de non-responsabilité importantes et informations juridiques

Liens hypertexte

Certains liens affichent une icône et/ou du texte contextuel. Ils fournissent des informations complémentaires.

Explication des icônes :

- Liens accompagnés de l'icône  : vous accédez à un site Web non hébergé par SAP. En utilisant de tels liens, vous acceptez (sauf indication contraire expresse dans vos contrats avec SAP) ce qui suit :
 - Le contenu du site vers lequel redirige le lien n'est pas de la documentation SAP. Vous ne pouvez émettre aucune réclamation produit auprès de SAP sur la base de ces informations.
 - SAP n'accepte pas ou désapprouve le contenu affiché sur le site vers lequel vous êtes redirigé, ni ne garantit la disponibilité et l'exactitude dudit contenu. SAP ne saurait être tenue responsable des dommages causés par l'utilisation dudit contenu sauf si de tels dommages étaient causés par une négligence grave ou une faute intentionnelle de SAP.
- Liens accompagnés de l'icône  : vous quittez la documentation associée à un produit ou service SAP en particulier et accédez à un site Web hébergé par SAP. En utilisant lesdits liens, vous convenez (sauf indication contraire expresse dans vos contrats avec SAP) que vous ne pourrez pas émettre de réclamation produit auprès de SAP sur la base de ces informations.

Vidéos hébergées sur des plateformes externes

Certaines vidéos peuvent pointer vers des plateformes d'hébergement de vidéos tierces. SAP ne garantit pas la disponibilité future des vidéos stockées sur ces plateformes. Par ailleurs, toute annonce ou tout autre contenu hébergé(e) sur ces plateformes (par exemple, suggestions de vidéos ou navigation vers d'autres vidéos hébergées sur le même site) ne relève ni du contrôle ni de la responsabilité de SAP.

Fonctionnalités Beta et expérimentales

Les fonctionnalités expérimentales ne font pas partie des éléments officiellement fournis par SAP et garantis pour les versions à venir. Cela signifie que les fonctionnalités expérimentales peuvent être modifiées par SAP à tout moment pour quelle que raison que ce soit, sans préavis. Les fonctionnalités expérimentales ne sont pas conçues pour être utilisées en production. Vous ne pouvez pas faire la démonstration, tester, examiner, évaluer ou utiliser d'une quelconque autre manière les fonctionnalités expérimentales dans un environnement productif ou avec des données n'ayant pas été suffisamment sauvegardées.

Le but des fonctionnalités expérimentales est d'obtenir rapidement des avis afin que les clients et partenaires puissent influencer le produit futur. En partageant votre avis (par exemple sur SAP Community), vous acceptez que les droits de propriété intellectuelle des contributions ou œuvres dérivées constituent la propriété exclusive de SAP.

Exemple de code

Les codes et/ou fragments de code ne sont que des exemples. Ils ne sont pas destinés à une utilisation en production. L'exemple de code est utilisé uniquement pour mieux expliquer et visualiser les règles de syntaxe. SAP ne garantit pas l'exactitude ni l'exhaustivité de l'exemple de code. SAP ne saurait être tenue responsable des erreurs ou dommages causés par l'utilisation dudit exemple de code sauf si de tels dommages étaient causés par une négligence grave ou une faute intentionnelle de SAP.

Langage sans préjugés

SAP soutient une culture de diversité et d'inclusion. Chaque fois que cela est possible, nous utilisons un langage impartial dans notre documentation pour faire référence aux personnes de toutes cultures, ethnies, genres et capacités.

© 2024 SAP SE ou société affiliée SAP. Tous droits réservés.

Toute reproduction ou communication de la présente publication, même partielle, par quelque procédé et à quelque fin que ce soit, est interdite sans l'autorisation expresse et préalable de SAP SE ou d'une société affiliée SAP. Les informations du présent document sont susceptibles d'être modifiées sans préavis.

Certains logiciels commercialisés par SAP SE et ses distributeurs contiennent des composants logiciels qui sont la propriété d'éditeurs tiers. Les spécifications des produits peuvent varier d'un pays à l'autre.

Les informations du présent document sont fournies par SAP SE ou par une société affiliée SAP uniquement à titre informatif, sans engagement ni garantie d'aucune sorte. SAP SE ou ses sociétés affiliées ne pourront en aucun cas être tenues responsables des erreurs ou omissions relatives à ces informations. Les seules garanties fournies pour les produits et les services de SAP SE ou d'une société affiliée SAP sont celles énoncées expressément à titre de garantie accompagnant, le cas échéant, lesdits produits et services. Aucune des informations contenues dans le présent document ne saurait constituer une garantie supplémentaire.

SAP et tous les autres produits et services SAP mentionnés dans ce document, ainsi que leurs logos respectifs, sont des marques commerciales ou des marques déposées de SAP SE (ou d'une société affiliée SAP) en Allemagne ainsi que dans d'autres pays. Tous les autres noms de produit et service mentionnés sont des marques commerciales de leurs sociétés respectives.

Veillez consulter <https://www.sap.com/france/about/legal/trademark.html> pour plus d'informations sur les marques déposées.