記事
Seisuke Nakahashi · 2022年5月31日 6m read

OAuth 2.0 を利用して IRIS から Gmail を送信する

2022年５月30日より、外部アプリからGmailを送信するには OAuth 2.0が必須になりました (リンク)
そのため、**IRIS から Gmail を送信したい** 場合は、今後は **OAuth 2.0 を利用して Gmailアカウントに接続する** 必要があります。 そこで、IRIS コード ObjectScript を使って、OAuth 2.0 でGmailアカウントに接続＆メール送信するサンプルを作成しました。 ぜひご活用ください！

処理の流れ概要

1. 【準備】Google で、OAuth 2.0 を利用するための設定をする : 手順(1)(2)(3)
2. 【準備】IRIS で、SSLクライアントの設定をする : 手順(4)
3. ブラウザで、指定URLを開き、Google アカウントでログインし、認可コードを取得 : 手順(5)
4. IRIS ルーチンで、取得した認可コードをもとに、AccessToken を取得 : 手順(6)
5. IRIS ルーチンで、取得した AccessToken を使って Gmail 送信 : 手順(6)

それでは、順を追って、具体的な手順をお伝えします。

以下、アカウント xxxxxx@gmail.com を利用するとします。

(1) Google のAPIコンソールで、プロジェクト作成し、OAuth 同意情報を登録

Google API コンソール https://console.developers.google.com/ を開きます。

OAuth 同意画面 > プロジェクトを作成 (右上) > 以下を入力し [**作成**] をクリック
プロジェクト名: IRISGMAIL (任意)
場所: 組織なし

> User Type で **外部** を選び [**作成**] をクリック
> 以下を入力し [**保存して次へ**] をクリック

アプリ名: fromG (任意)
ユーザサポートメール: xxxxxx@gmail.com
デベロッパーの連絡先情報: xxxxxx@gmail.com

> スコープ で何も入力せず [**保存して次へ**] をクリック
> テストユーザ で **+ADD USERS** から xxxxxx@gmail.com を追加して [**保存して次へ**] をクリック

(2) Google のAPIコンソールで、認証情報を追加

認証情報 > 認証情報を作成 (上) > OAuthクライアントID > 以下を入力し [**作成**] をクリック

アプリケーションの種類: ウェブアプリケーション
名前: forIRIS (任意)
承認済みの JavaScript 生成元: http://127.0.0.1/
承認済みのリダイレクトURI: http://127.0.0.1/test.html (任意、ただし存在するページにしてください)

ここで、クライアントIDと、クライアントシークレットが表示されるのでメモします (以下はあくまでサンプル)

クライアントID	123-abc.googleusercontent.com
クライアントシークレット	GOCSPX-xyz999

(3) Google のAPIコンソールで、Gmail API を有効化

有効なAPIとサービス > APIとサービスの有効化 (上) > Gmail API 選択 > [**有効にする**] をクリック

(4) IRIS で SSLクライアント構成

IRIS管理ポータル を実行

システム管理 > セキュリティ > SSL/TLS構成 > 新規構成の作成 > 以下を入力し [**保存**] をクリック

構成名: GMAILSSL (任意)

※その他はデフォルト値のままでOKです

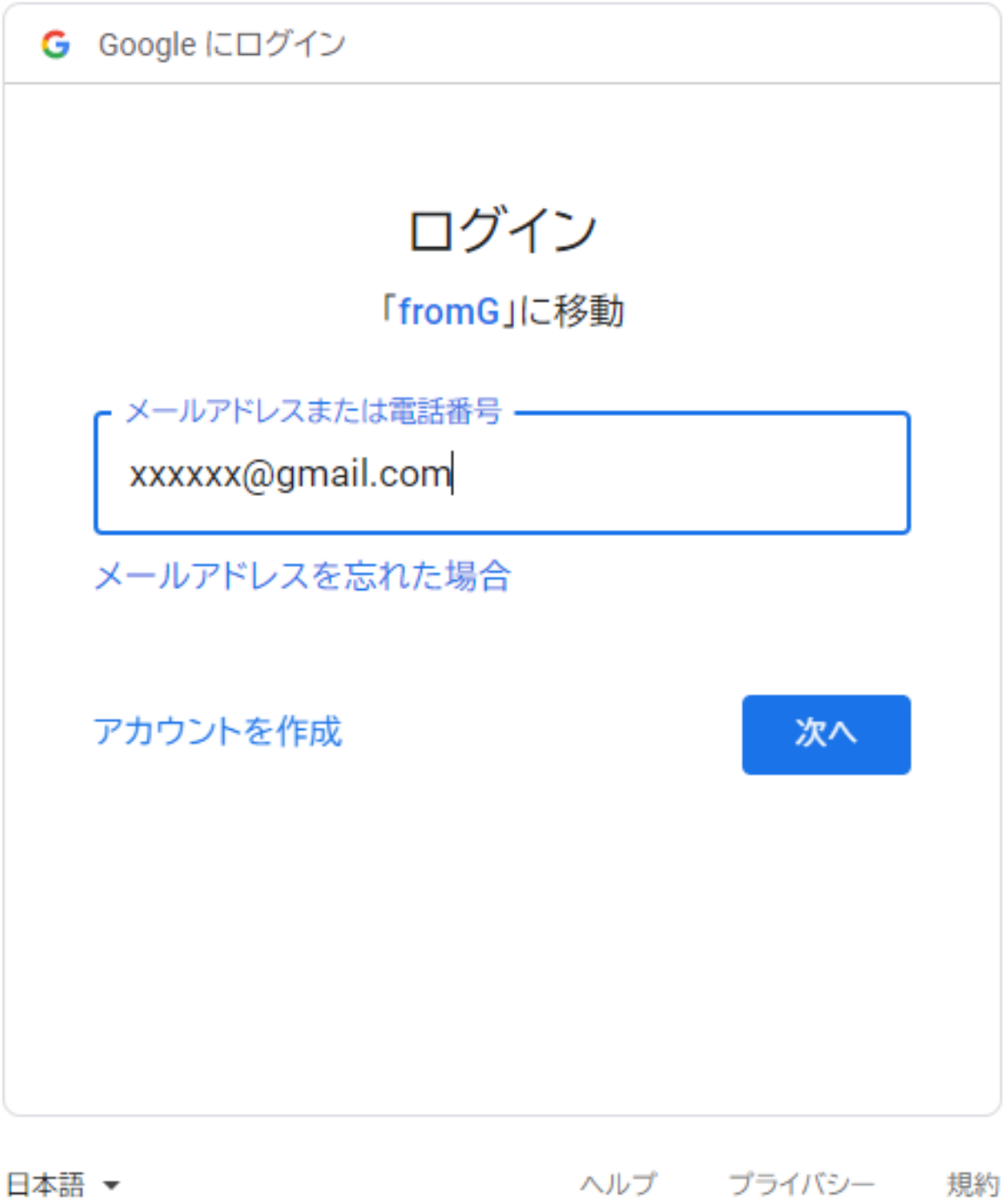
(5) Googleの認可コードをブラウザ経由で取得

「Gmail へのスコープとして、Google アカウントへのアクセスリクエストを許可する」URLをブラウザから実行
(**以下のURLは、実際は1行**)

https://accounts.google.com/o/oauth2/v2/auth?scope=https://mail.google.com/
&accesstype=offline&includegrantedscope=true&responsetype=code
&redirecturi= http://127.0.0.1/test.html
&clientid= 123-abc.googleusercontent.com

redirecturi と clientid は手順(2)で設定・取得したもの

実行すると、Googleアカウントを入力する画面になるので、 xxxxxx@gmail.com でログイン



警告が表示されるが [**続行**] > [**続行**] を選択すると、指定したリダイレクト先 http://127.0.0.1/test.html に飛ぶ

するとブラウザのURLに code が含まれた以下のようになる (以下はあくまでサンプル)

http://127.0.0.1/test.html/?code=4/0ABCXYZ999&scope=https://mail.google.com/

この code が認可コードとなるので、この値 4/0ABCXYZ999 をメモしておく

(6) IRIS で認可コードをもとに AccessToken を取得、そのTokenを使ってメール送信

以後は IRIS ルーチンから、ObjectScript を使って、メール送信します。

```
 ;;-----------------------------------------------------------------------------------------
  ;; 認可コードをもとに Google から Access Token を取得
 ;;-----------------------------------------------------------------------------------------
  set req = ##class(%Net.HttpRequest).%New()
  set req.SSLConfiguration="GMAILSSL"  ; 手順(4)で設定した SSL構成名
  do req.SetParam("code","4/0ABCXYZ999")  ; 手順(5)で入手した認可コード
テストプロジェクトのため、毎回取得しなおす必要あり
  do req.SetParam("clientid","  123-abc.googleusercontent.com")   ; 手順(2)で入手した クライアントID
  do req.SetParam("clientsecret","  GOCSPX-xyz999") ; 手順(2)で入手した クライアントシークレット
  do req.SetParam("redirecturi","  http://127.0.0.1/test.html")  ; 手順(2)で設定した リダイレクト先
  do req.SetParam("granttype","authorizationcode")
  do req.SetParam("accesstype","offline")
  do req.Post("https://accounts.google.com/o/oauth2/token")  ; Google から Access Token
を取得するためのリクエスト

  set resData = req.HttpResponse.Data
  set json=resData.Read(10000) ; Google から戻ってきたJSON
  write json,!!
  set AccessToken={}.%FromJSON(json)."accesstoken"    ; JSON から "accesstoken" の値を取得
  write AccessToken,!!

 ;;-----------------------------------------------------------------------------------------
  ;; Access Token を指定して Gmail から送信
 ;;-----------------------------------------------------------------------------------------
  set mbox= ##class(%Net.SMTP).%New()
  set mbox.smtpserver="smtp.gmail.com"
  set mbox.port="587"
  set mbox.UseSTARTTLS=1
  set mbox.SSLConfiguration="GMAILSSL" ; 手順(4)で設定した SSL構成名

  set auth=##class(%Net.Authenticator).%New()
  set auth.MechanismList="XOAUTH2"
  set auth.UserName="xxxxxx@gmail.com"
  set auth.AccessToken=AccessToken ; 上記で取得した Access Token
  set mbox.authenticator=auth

  ; メール本文
  set message=##class(%Net.MailMessage).%New()
  do message.To.Insert("xxxxxx@abc.com")  ; TO
  set message.From="xxxxxx@gmail.com"  ; FROM
  set message.Subject="test from IRIS"  ; Subject
  set status=message.TextData.Write("Hello!")  ; message

  ; 送信
  set status=mbox.Send(message)
  write $system.OBJ.DisplayError(status),!
```

メールが送信できればテスト完了です。おつかれさまでした

上記で作成した Google Cloud プロジェクトは 「リソースの管理」から 削除できます。
https://console.developers.google.com/cloud-resource-manager

#OAuth2 #ObjectScript #InterSystems IRIS

ソースURL:
https://jp.community.intersystems.com/post/oauth-20-%E3%82%92%E5%88%A9%E7%94%A8%E3%81%97%E3%81%A6-iris-%E3%81%8B%E3%82%89-gmail-%E3%82%92%E9%80%81%E4%BF%A1%E3%81%99%E3%82%8B