

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が必要になりました ([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を眺め、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元に、Access_Token を取得： 手順(6)
5. IRIS ルーチンで、取得した Access_Token を使って Gmail 送信： 手順(6)

ここでは、順を追って、**具体手順**をお伝えします。

例、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、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上) > 以を入力し [**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

> 以を入力し [**探して次へ**] をクリック

アプリ名: fromG (任意)

ユザサポートメール: xxxxxx@gmail.com

デバイス間の連絡先情報: xxxxxx@gmail.com

> スcopeで既入力せず [**探して次へ**] をクリック

> テストユザで +ADD USERS から xxxxxx@gmail.com を追加して [**探して次へ**] 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 > 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)
認可済みの JavaScript 生成: <http://127.0.0.1/>
認可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サービス > API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 新規構成作成 以下を入力し [**保存**] をクリック

構成: **GMAILSSL** (任意)

その他はデフォルト値のままです

(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/
&access_type=offline&include_granted_scope=true&response_type=code
&redirect_uri=http://127.0.0.1/test.html
&client_id=123-abc.googleusercontent.com](https://accounts.google.com/o/oauth2/v2/auth?scope=https://mail.google.com/&access_type=offline&include_granted_scope=true&response_type=code&redirect_uri=http://127.0.0.1/test.html&client_id=123-abc.googleusercontent.com)

redirect_uri と client_id は手順(2)で設定・取得した値

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



日本語 ▾

ヘルプ

プライバシー

規約

警告#表示されるが [続行] > [続行] を選択する、指定したリダイレクト先

<http://127.0.0.1/test.html> に飛ぶ

する。ブラウザのURLに code #含まれたいようになる (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 が認可コードになるので、この値 `4/0ABCXYZ999` をメモしておく

(6) IRIS で認可コードを元に 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。

```

-----
;; 認可コードを元に 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
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("client_id","123-abc.googleusercontent.com") ; 手順(2)で入手したクライアントID
do req.SetParam("client_secret","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("redirect_uri","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("grant_type","authorization_code")
do req.SetParam("access_type","offline")
do req.Post("https://accounts.google.com/o/oauth2/token") ; Google から Access Token
; 取得するためのリクエスト

set resData = req.HttpResponse.Data
set json=resData.Read(10000) ; Google から戻った JSON
write json,!
set AccessToken={}.%FromJSON(json)."access_token" ; JSON から "access_token" 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

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>