



**Global Office 365**  
**Developer Bootcamp**

# Microsoft Graph ハンズオン

2018/10/27

篠原敬志 (@karamem0)

# 自己紹介

Name	篠原敬志 (Takashi Shinohara)
Company	アバナード株式会社 シニアコンサルタント
Award	Microsoft MVP for Office Development (2018-)
Twitter	@karamem0
Blog	からめもぶろぐ。 ( <a href="http://blog.karamem0.jp">http://blog.karamem0.jp</a> )
Bio	SharePoint が得意なフレンズ



# 本日のアジェンダ

Microsoft Graph とは

ハンズオン: Graph Explorer

OAuth とは

Microsoft Graph における OAuth

ハンズオン: Hello Microsoft Graph



# Microsoft Graph とは

# Microsoft Graph

**Microsoft Graph は  
Microsoft 365 向けの API  
です**

Office 365、Windows 10、Enterprise Mobility + Security に接続して、創造性とコラボレーションを強化しましょう。

[詳細情報 >](#)



# Microsoft 365



Office 365



Windows 10



Enterprise  
Mobility + Security

企業向けのサブスクリプション型の統合型ソリューション  
生産性を最大限に高めるアプリケーション  
ビジネスを安全に守る高度なセキュリティとデバイスの管理

# 統合型プログラミング モデル

1 つのエンドポイント

<https://graph.microsoft.com>

豊富なリソース

Azure Active  
Directory

Office 365

Enterprise Mobility  
+ Security

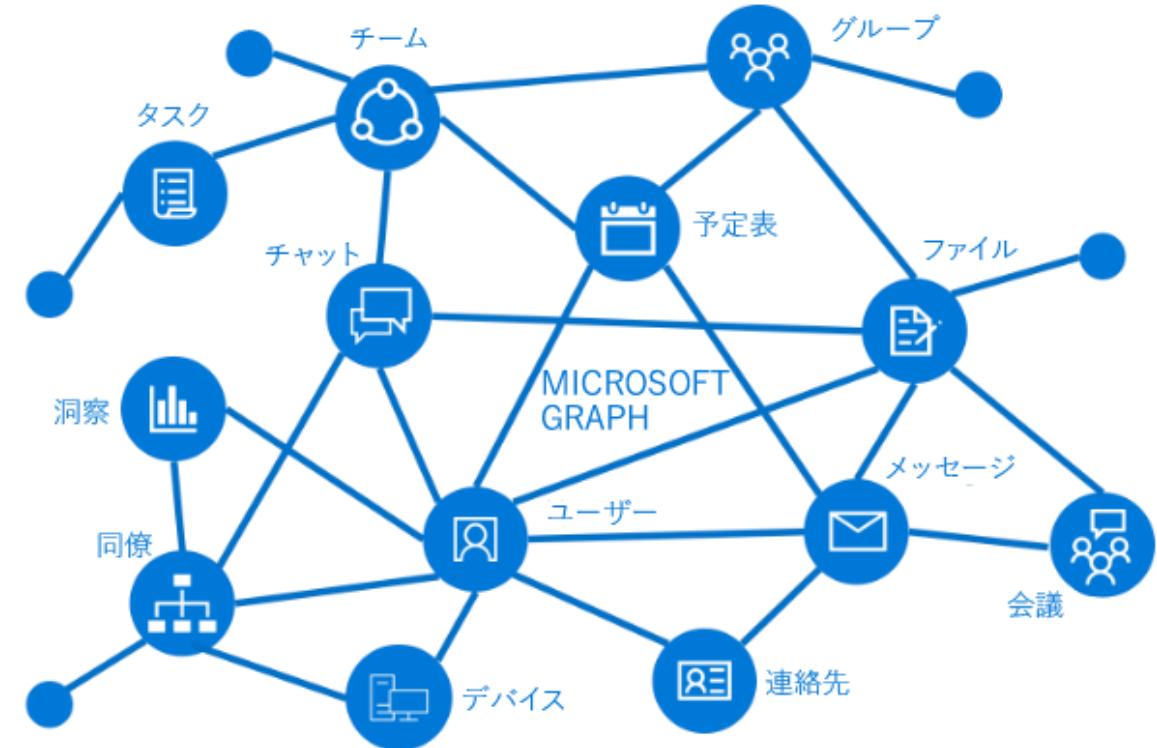
Windows 10

教育

# リレーションシップ

Graph: 「つながり方」に着目して抽象化された「点とそれらをむすぶ線」の概念

/me からはじまる



# 提供されるサービス

サービス	機能
ユーザーとグループ	ユーザー、グループ
ID およびアクセス管理	ID およびアクセス管理
生産性	予定表、ファイル、メール、メモ、個人用連絡先、ブックとグラフ
グループ作業	サイトとリスト、タスクとプラン、チームワーク (プレビュー)
ソーシャル インテリジェンスおよび分析機能	連絡先、ドキュメントのインサイト (プレビュー)
デバイスの管理	デバイスとアプリ
セキュリティ	セキュリティ統合 (プレビュー)、Identity Protection (プレビュー)
クロスデバイス エクスペリエンス	クロスデバイス エクスペリエンス
利用状況レポート	レポート
教育	教育
ビジネス アプリケーション	顧客の予約 (プレビュー)

# API のバージョン

一般提供	<a href="https://graph.microsoft.com/v1.0">https://graph.microsoft.com/v1.0</a>
プレビュー版	<a href="https://graph.microsoft.com/beta">https://graph.microsoft.com/beta</a>

プレビュー機能が一般提供になった場合は最新バージョンに取り込まれるバージョンが新しくなる場合、現在のバージョンは非推奨となり、24か月後に非サポートとなる

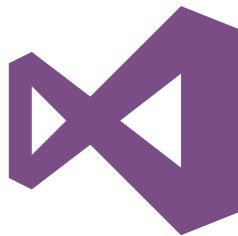
# HTTP メソッド

メソッド	説明
GET	リソースからデータを読み取ります。
POST	新しいリソースを作成、または処理を実行します。
PATCH	リソースを新しい値で更新します。
PUT	リソースを新しいものと置換します。
DELETE	リソースを削除します。

# クエリ パラメーター (OData)

名前	説明
\$count	一致するリソースの総数を取得します。
\$expand	関連リソースを取得します。
\$filter	結果 (行) をフィルターします。
\$format	指定したメディア形式で結果を返します。
\$orderby	結果を並べます。
\$search	検索条件に基づいて結果を返します。
\$select	プロパティ (列) をフィルターします。
\$skip	結果セットにインデックスを作成します。
\$skipToken	複数ページにわたる結果セットから、結果の次のページを取得します。
\$top	結果のページ サイズを設定します。

# SDK



(プレビュー)

(プレビュー)

# Graph Explorer

Microsoft Graph をお手軽に試すための Web サービス  
サンプル アカウントが使用可  
豊富なサンプル クエリ

A blurred photograph of a city street, likely Stockholm, showing multi-story apartment buildings with many windows, a road with cars, and a bridge in the distance.

# ハンズオン: Graph Explorer

# Graph Explorer

サンプル データを表示する

自分のユーザー情報を取得する

メールを送信する

OneDrive の Excel ファイルに値を書き込む

# Graph Explorer

サンプル データを表示する

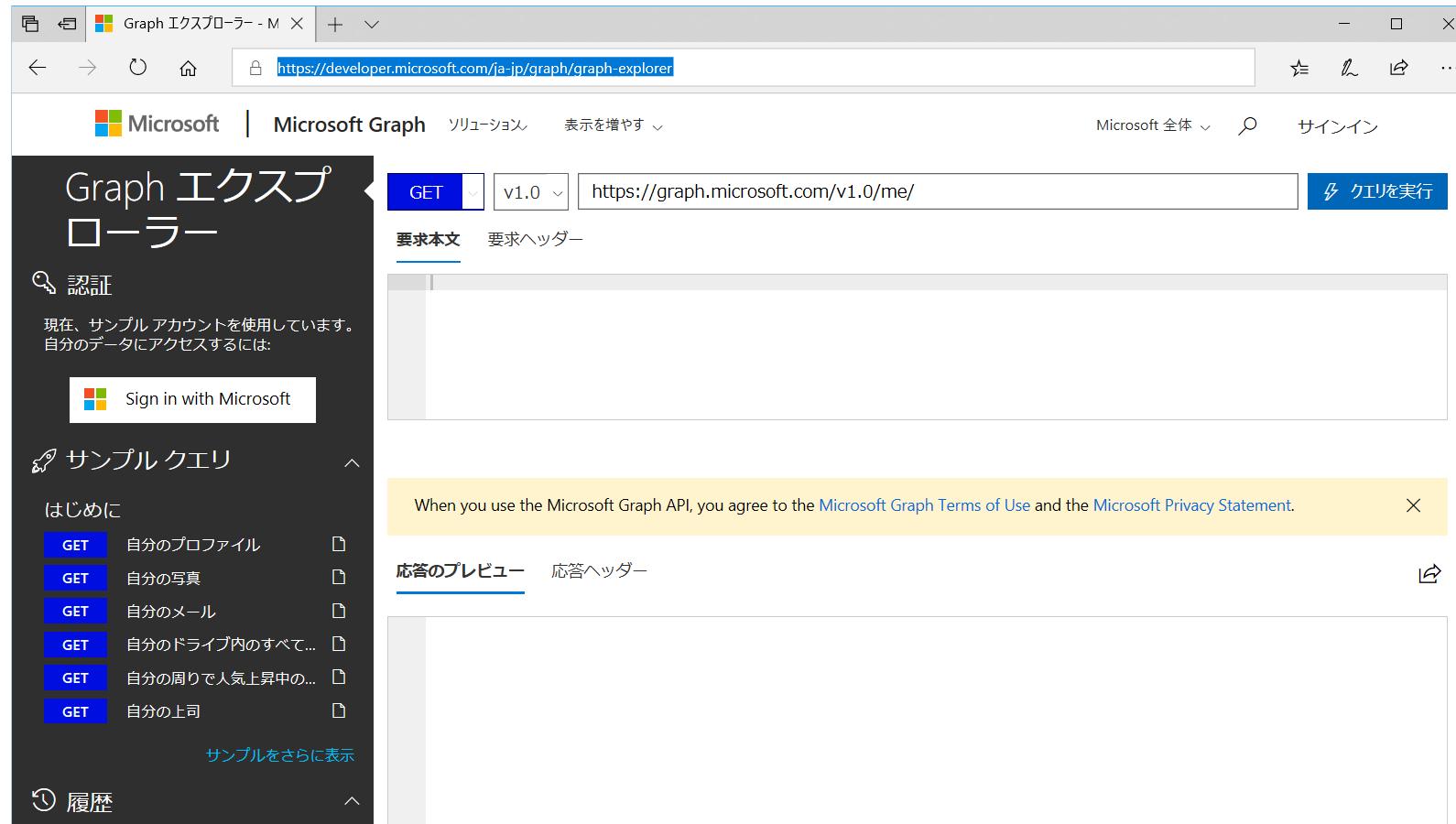
自分のユーザー情報を取得する

メールを送信する

OneDrive の Excel ファイルに値を書き込む

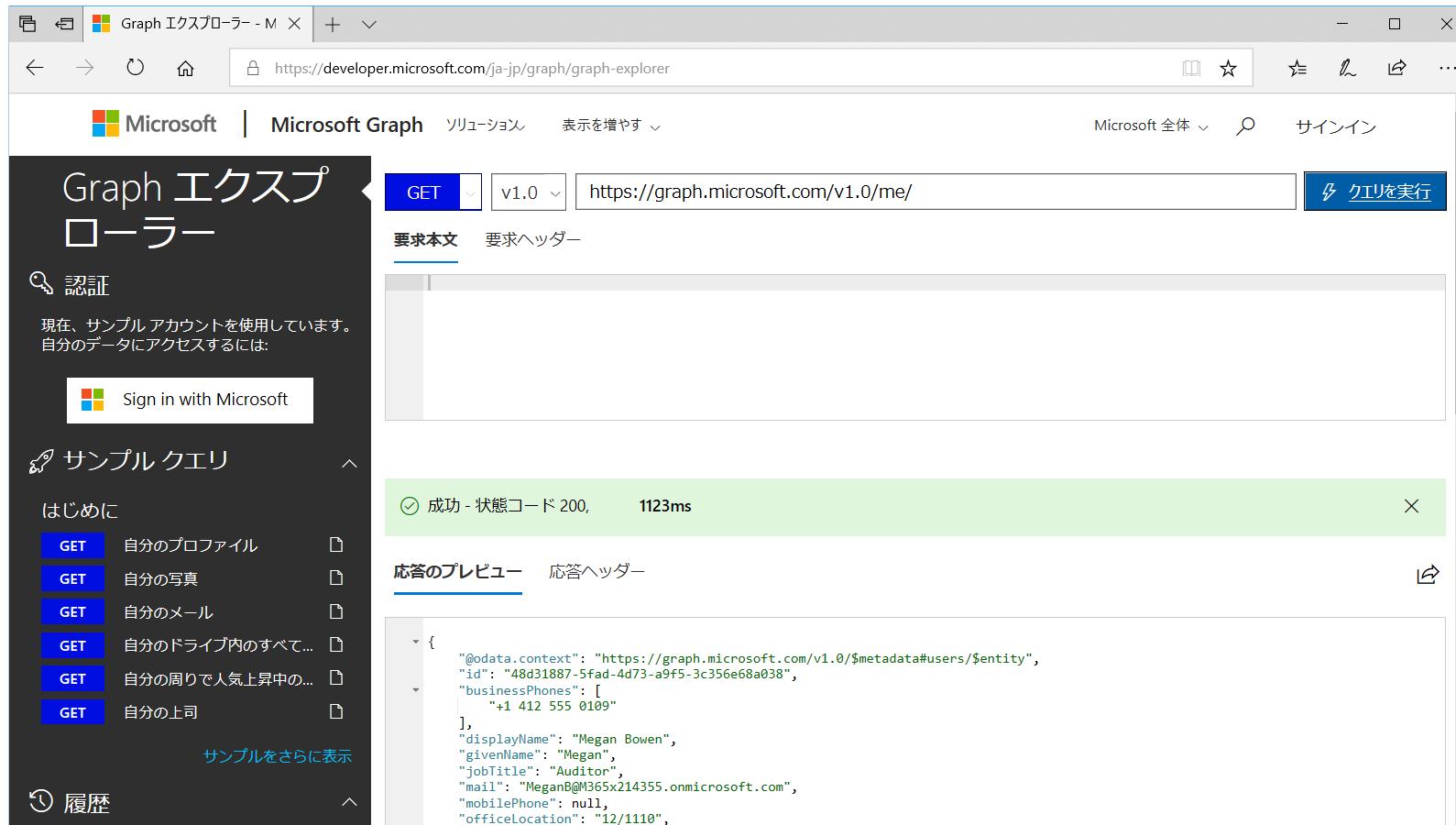
# サンプル データ: Graph Explorer

<https://developer.microsoft.com/ja-jp/graph/graph-explorer> に  
アクセス



# サンプル データ: クエリの実行

[クエリを実行] をクリック



The screenshot shows the Microsoft Graph Explorer interface. In the top navigation bar, the URL is https://developer.microsoft.com/ja-jp/graph/graph-explorer. The main area displays a GET request to https://graph.microsoft.com/v1.0/me/. The 'クエリを実行' (Run Query) button is highlighted in blue. Below the request, there are tabs for '要求本文' (Request Body) and '要求ヘッダー' (Request Headers). A green success message at the bottom indicates a status code of 200 and a response time of 1123ms. The '応答のプレビュー' (Response Preview) section shows the JSON response for the user profile:

```
{  
    "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users/$entity",  
    "id": "48d31887-5fad-4d73-a9f5-3c356e68a038",  
    "businessPhones": [  
        "+1 412 555 0109"  
    ],  
    "displayName": "Megan Bowen",  
    "givenName": "Megan",  
    "jobTitle": "Auditor",  
    "mail": "MeganB@m365x214355.onmicrosoft.com",  
    "mobilePhone": null,  
    "officeLocation": "12/1110",  
}
```

# Graph Explorer

サンプル データを表示する

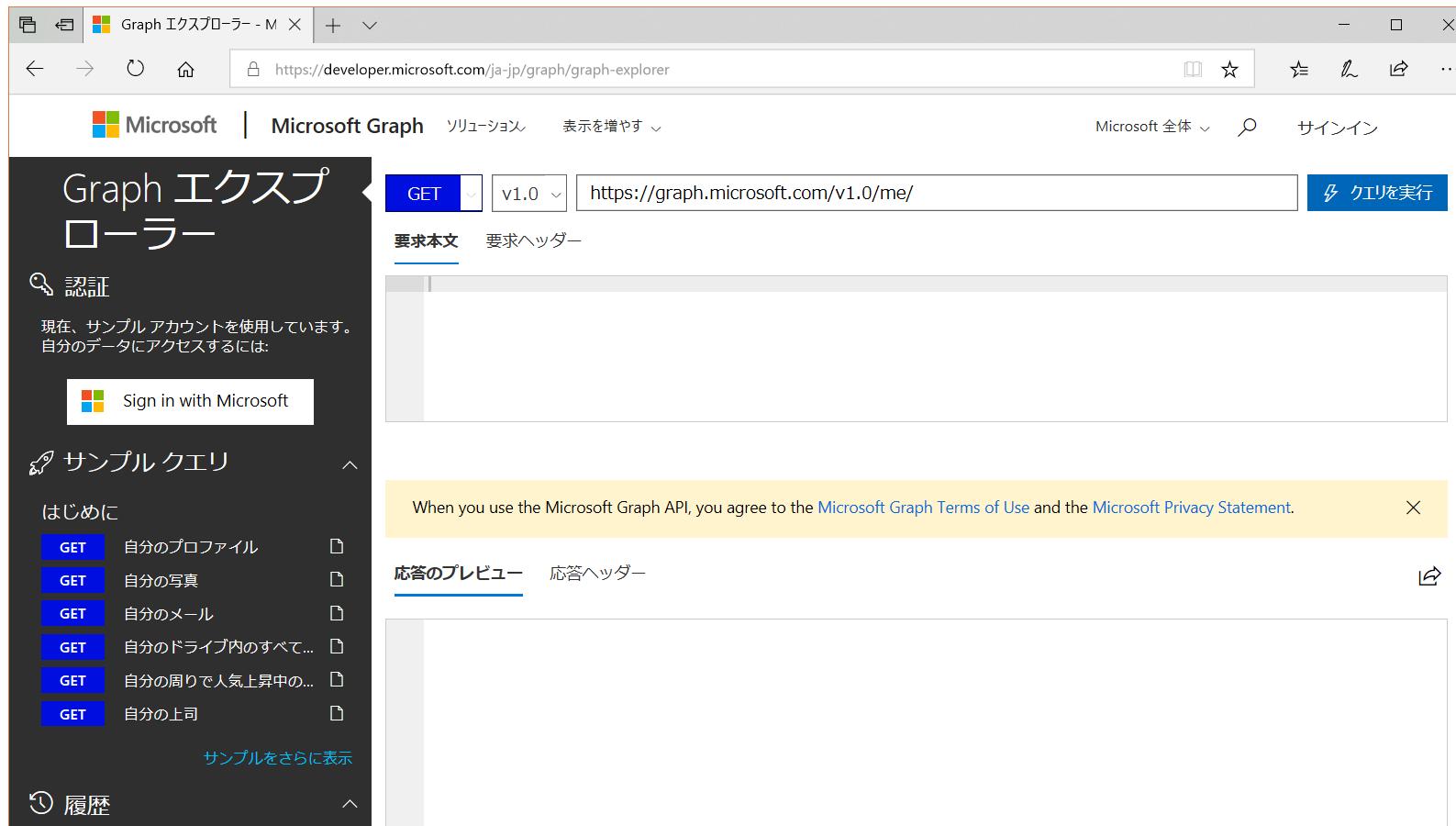
自分のユーザー情報を取得する

メールを送信する

OneDrive の Excel ファイルに値を書き込む

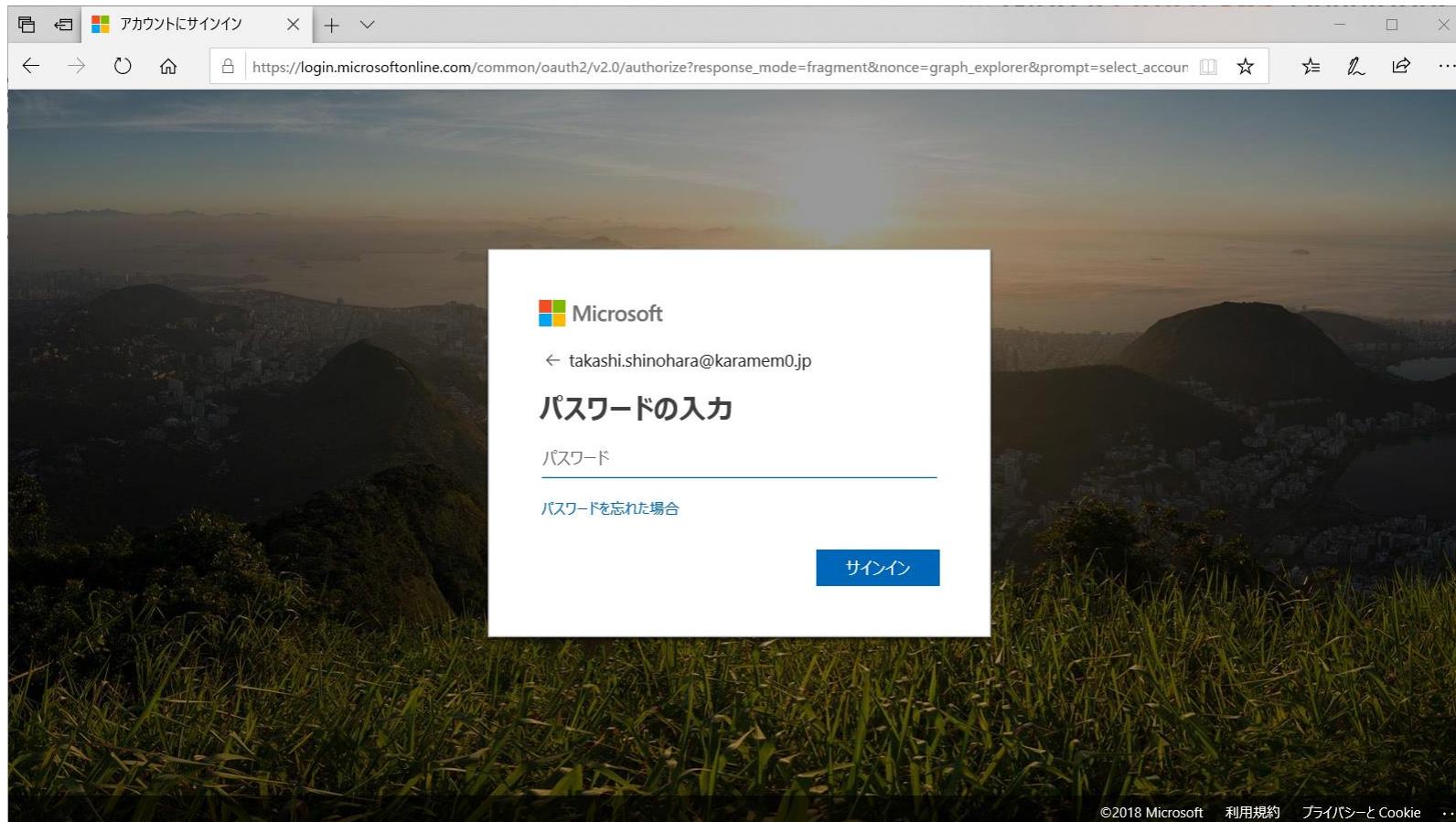
# 自分のユーザー情報: Sign in with Microsoft

[Sign in with Microsoft] をクリック



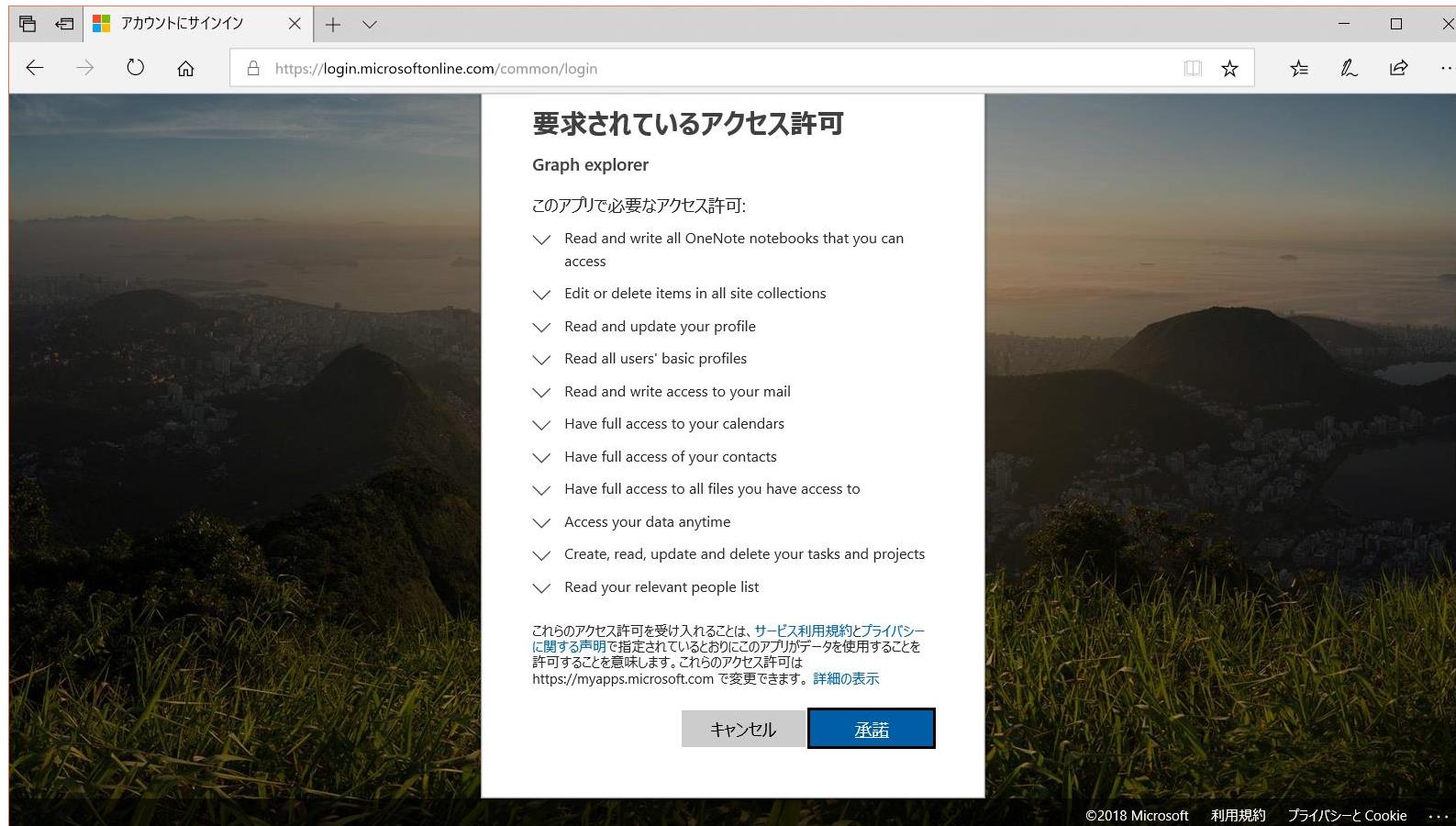
# 自分のユーザー情報: サインイン

## 組織アカウントでサインイン



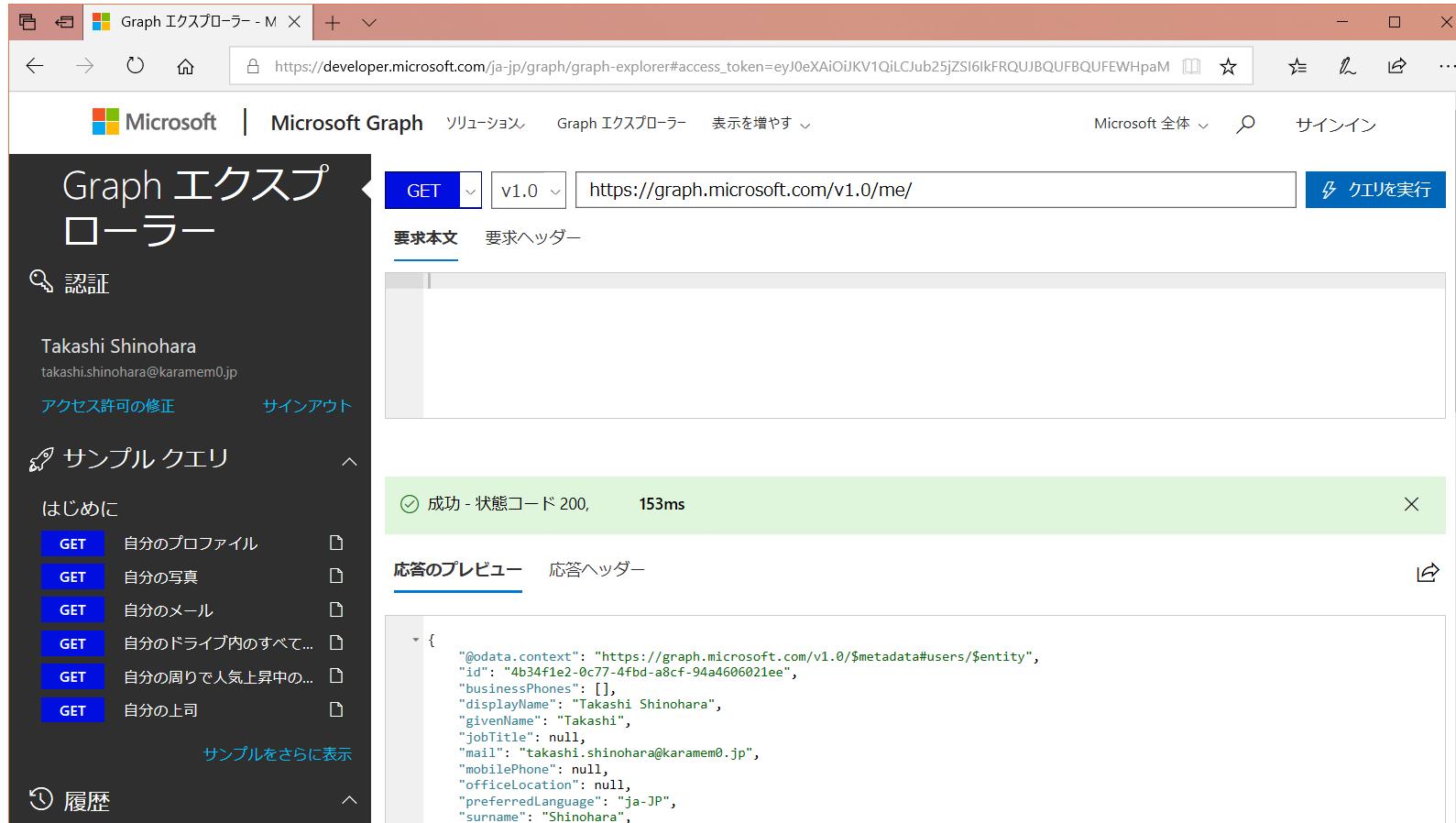
# 自分のユーザー情報: アクセス許可の承認

[承諾] をクリック



# 自分のユーザー情報: クエリの実行

[クエリを実行] をクリック

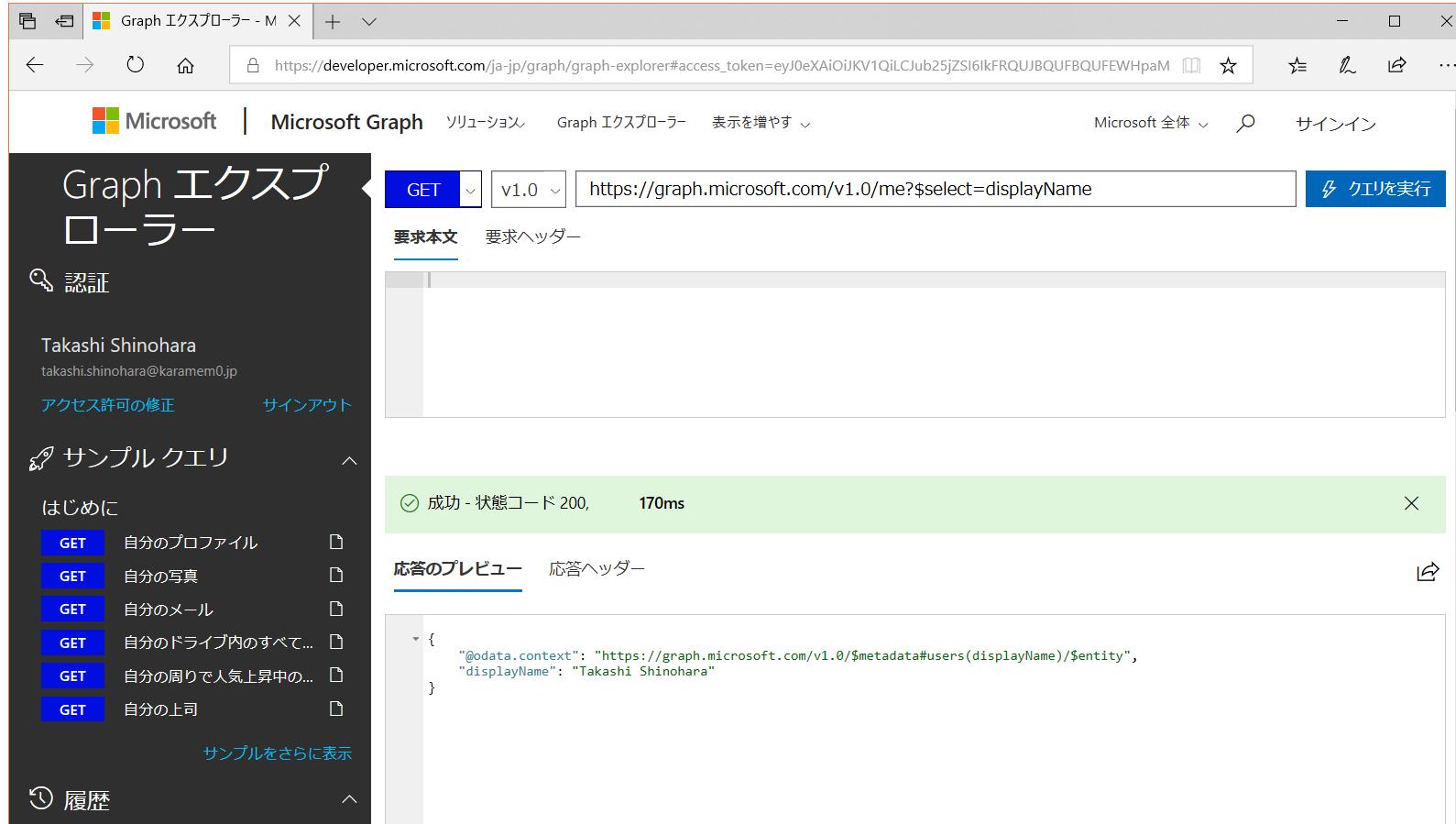


The screenshot shows the Microsoft Graph Explorer interface. In the top navigation bar, the URL is https://developer.microsoft.com/ja-jp/graph/graph-explorer#access\_token=eyJ0eXAiOiJKV1QiLCJub25jZSI6IkFRQUJBQUFBQUFEWHpaM. The main search bar contains GET v1.0 https://graph.microsoft.com/v1.0/me/. A blue button labeled 'クエリを実行' (Run Query) is visible. On the left sidebar, under 'サンプル クエリ', there is a list of sample queries: '自分のプロファイル', '自分の写真', '自分のメール', '自分のドライブ内のすべて...', '自分の周りで人気上昇中の...', and '自分の上司'. Below this list is a link 'サンプルをさらに表示'. At the bottom of the sidebar, there is a section titled '履歴' (History). The main content area displays a green success message: '成功 - 状態コード 200, 153ms'. Below this, the '応答のプレビュー' (Response Preview) section shows the JSON response:

```
  {
    "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users/$entity",
    "id": "4b34f1e2-0c77-4fb0-a8cf-94a4606021ee",
    "businessPhones": [],
    "displayName": "Takashi Shinohara",
    "givenName": "Takashi",
    "jobTitle": null,
    "mail": "takashi.shinohara@karamem0.jp",
    "mobilePhone": null,
    "officeLocation": null,
    "preferredLanguage": "ja-JP",
    "surname": "Shinohara",
  }
```

# 自分のユーザー情報: OData による操作

/v1.0/me?\$select=displayName を実行



The screenshot shows the Microsoft Graph Explorer interface. In the top navigation bar, the URL is https://developer.microsoft.com/ja-jp/graph/graph-explorer#access\_token=eyJ0eXAiOiJKV1QiLCJub25jZSI6IkFRQUJBQUFBQUFEWHpaM. The main search bar contains the query GET v1.0 https://graph.microsoft.com/v1.0/me?\$select=displayName. A blue button labeled 'クエリを実行' (Run Query) is visible. Below the search bar, there are tabs for '要求本文' (Request Body) and '要求ヘッダー' (Request Headers). A green success message at the bottom indicates '成功 - 状態コード 200, 170ms'. The '応答のプレビュー' (Response Preview) section shows the JSON response: { "@odata.context": "https://graph.microsoft.com/v1.0/\$metadata#users(displayName)/\$entity", "displayName": "Takashi Shinohara" }.

# Graph Explorer

サンプル データを表示する

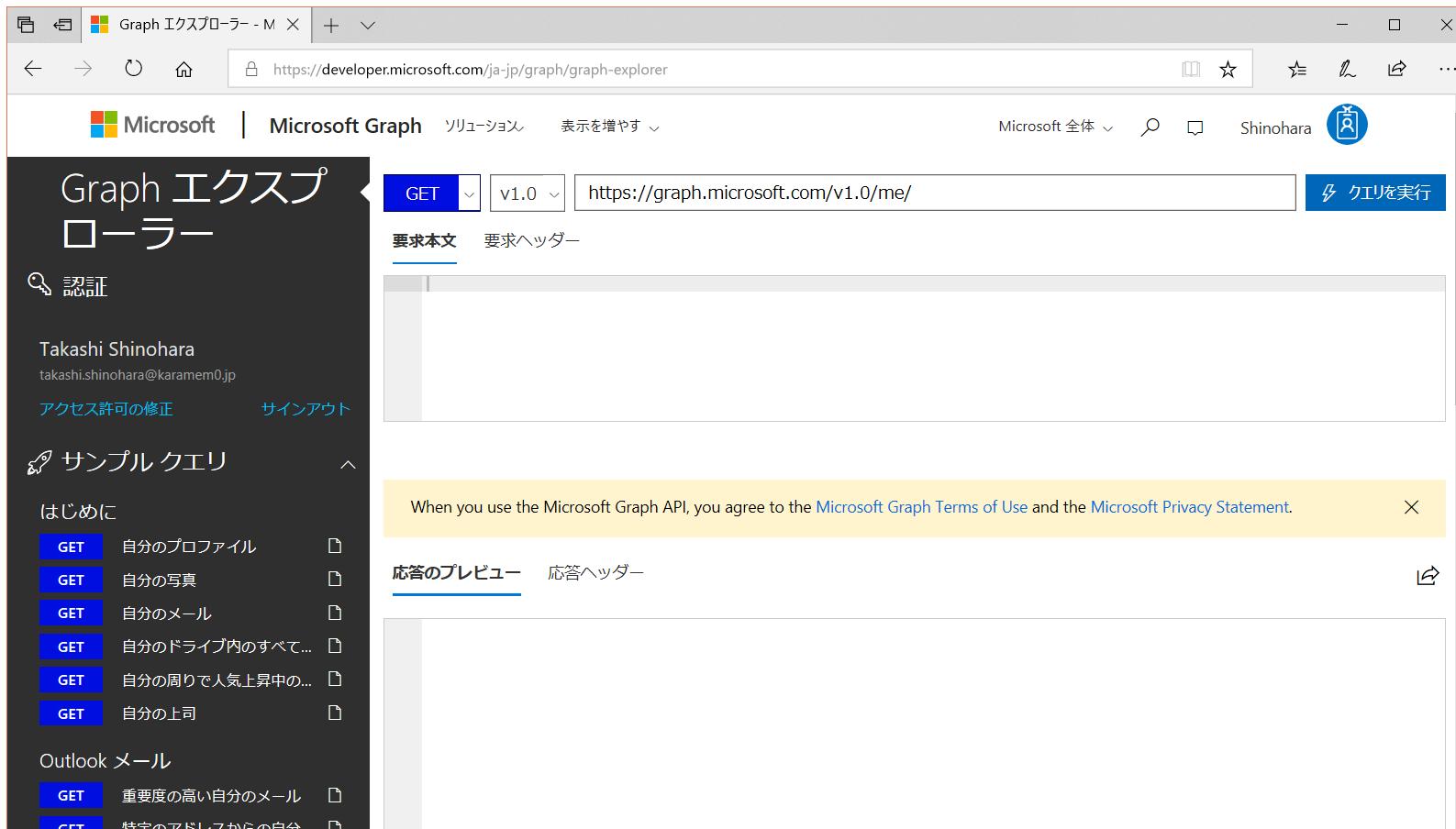
自分のユーザー情報を取得する

メールを送信する

OneDrive の Excel ファイルに値を書き込む

# メールの送信: アクセス許可の修正

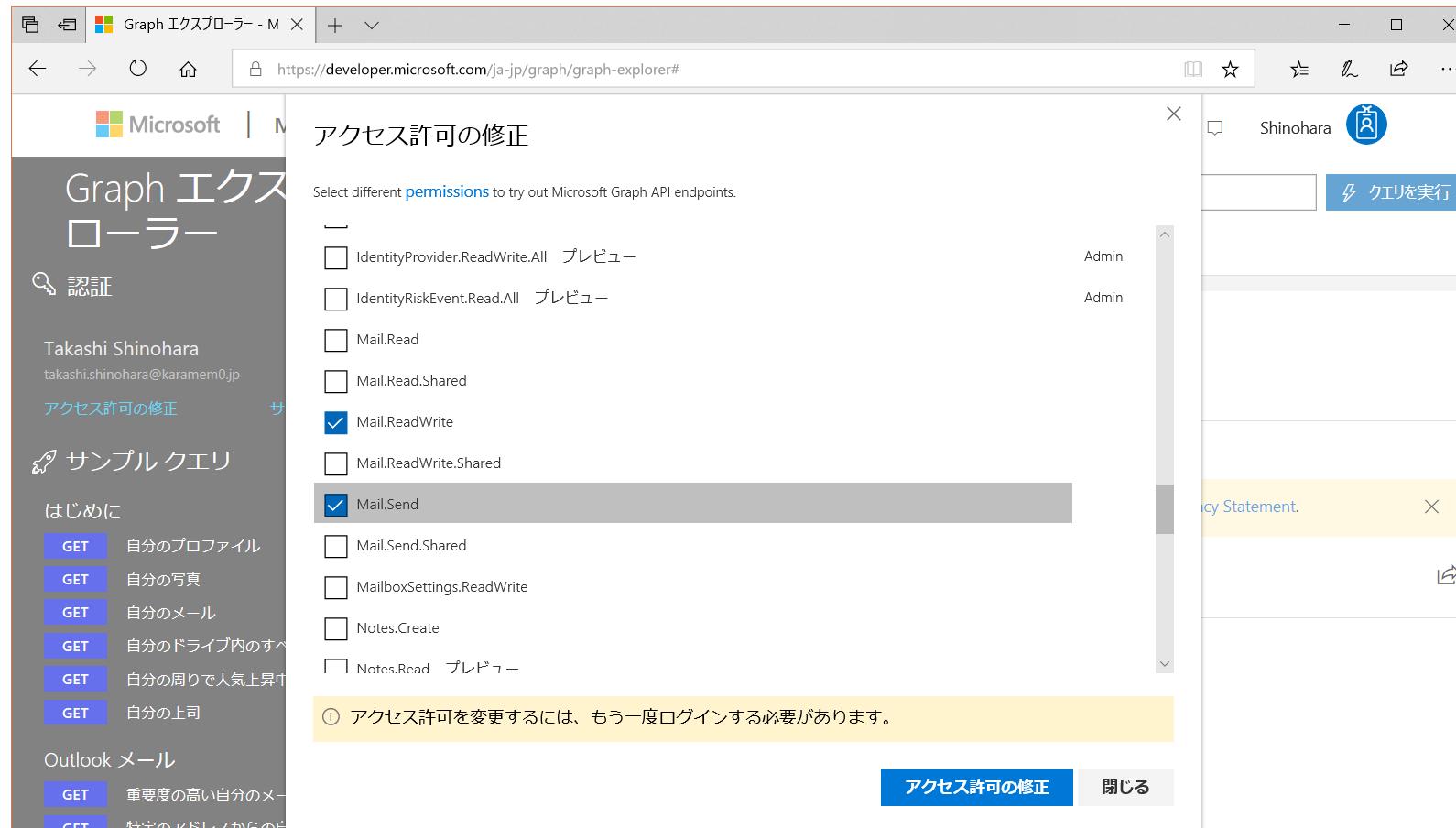
[アクセス許可の修正] をクリック



The screenshot shows the Microsoft Graph Explorer interface. In the top navigation bar, the URL is https://developer.microsoft.com/ja-jp/graph/graph-explorer. The main area displays a request to GET https://graph.microsoft.com/v1.0/me/. The request details section shows 'GET' selected under method, 'v1.0' under version, and the URL https://graph.microsoft.com/v1.0/me/. A large button labeled 'クエリを実行' (Execute Query) is visible. On the left sidebar, there are sections for '認証' (Authentication), 'Takashi Shinohara' (with email takashi.shinohara@karamem0.jp), 'アクセス許可の修正' (Update Access Permission), and 'サインアウト' (Sign Out). Below these, under 'サンプル クエリ' (Sample Query), there is a 'はじめに' (Getting Started) section with several GET requests listed, such as '自分のプロファイル', '自分の写真', etc. At the bottom of the main window, a yellow banner states: 'When you use the Microsoft Graph API, you agree to the [Microsoft Graph Terms of Use](#) and the [Microsoft Privacy Statement](#)'. There is also a link to '応答のプレビュー' (Response Preview) and '応答ヘッダー' (Response Headers).

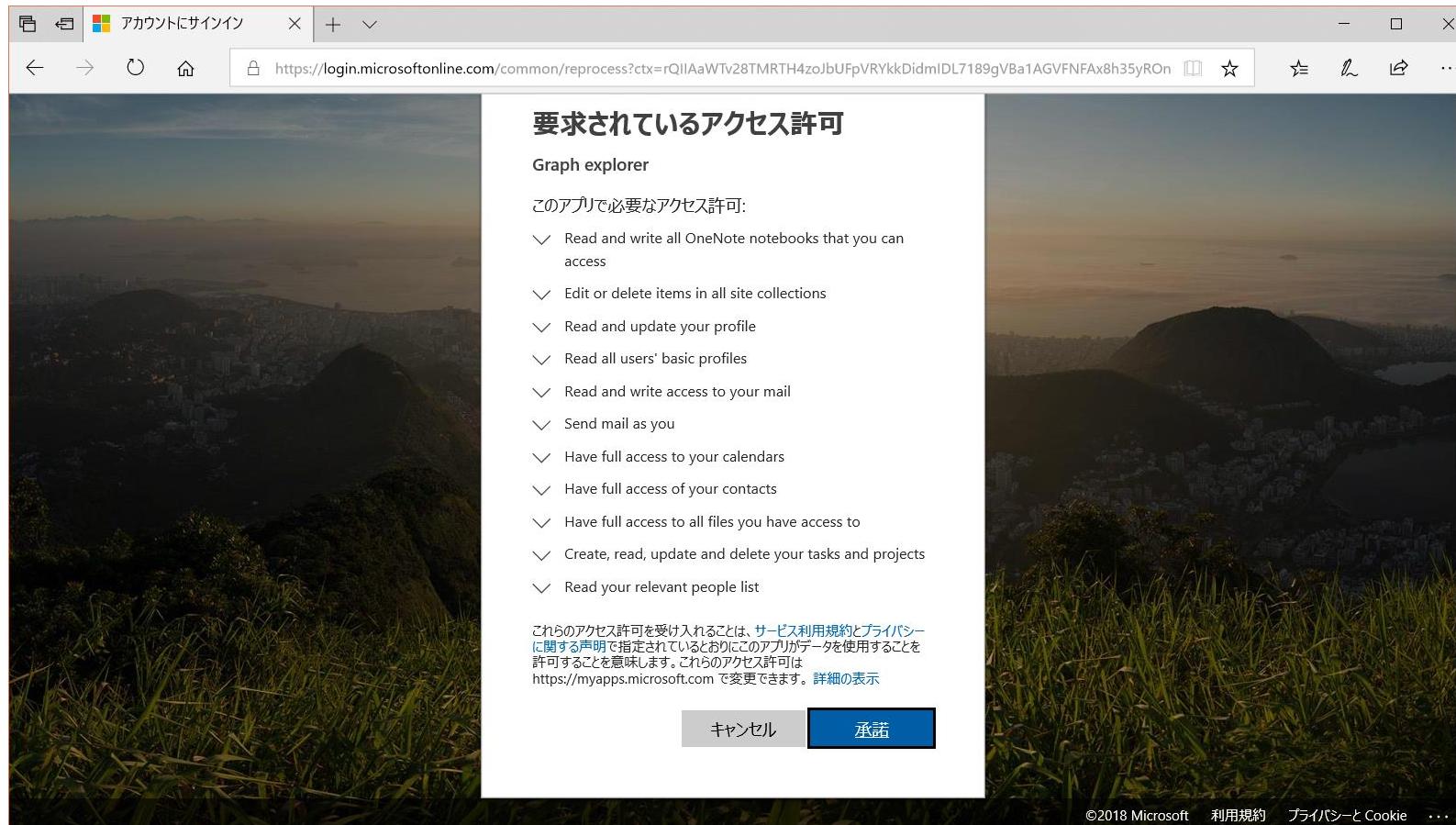
# メールの送信: アクセス許可の選択

[Mail.Send] を選択して [アクセス許可の修正] をクリック



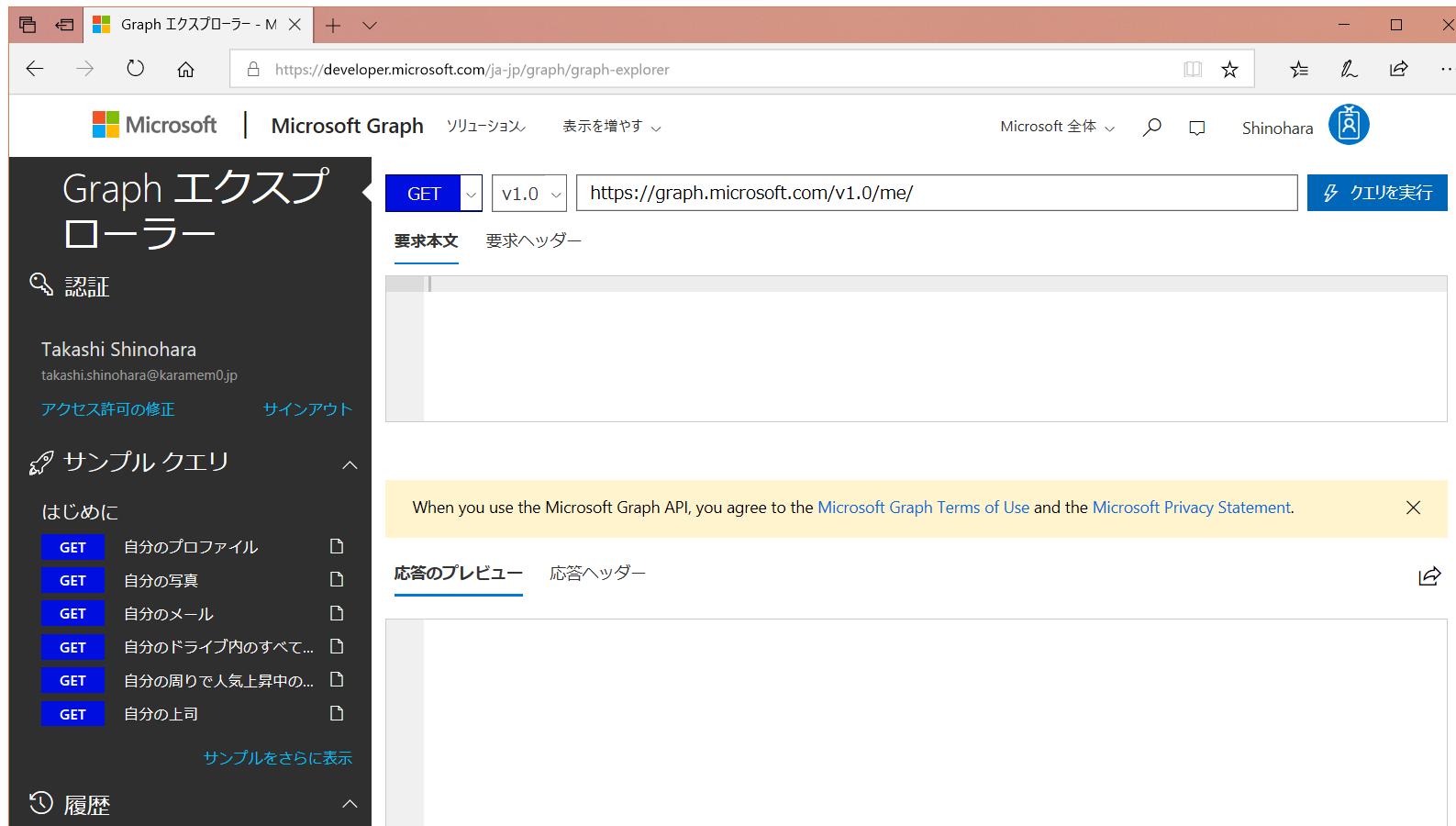
# メールの送信:アクセス許可の承認

[承諾] をクリック



# メールの送信: サンプルをさらに表示

[サンプルをさらに表示] をクリック



# メールの送信: サンプル カテゴリの選択

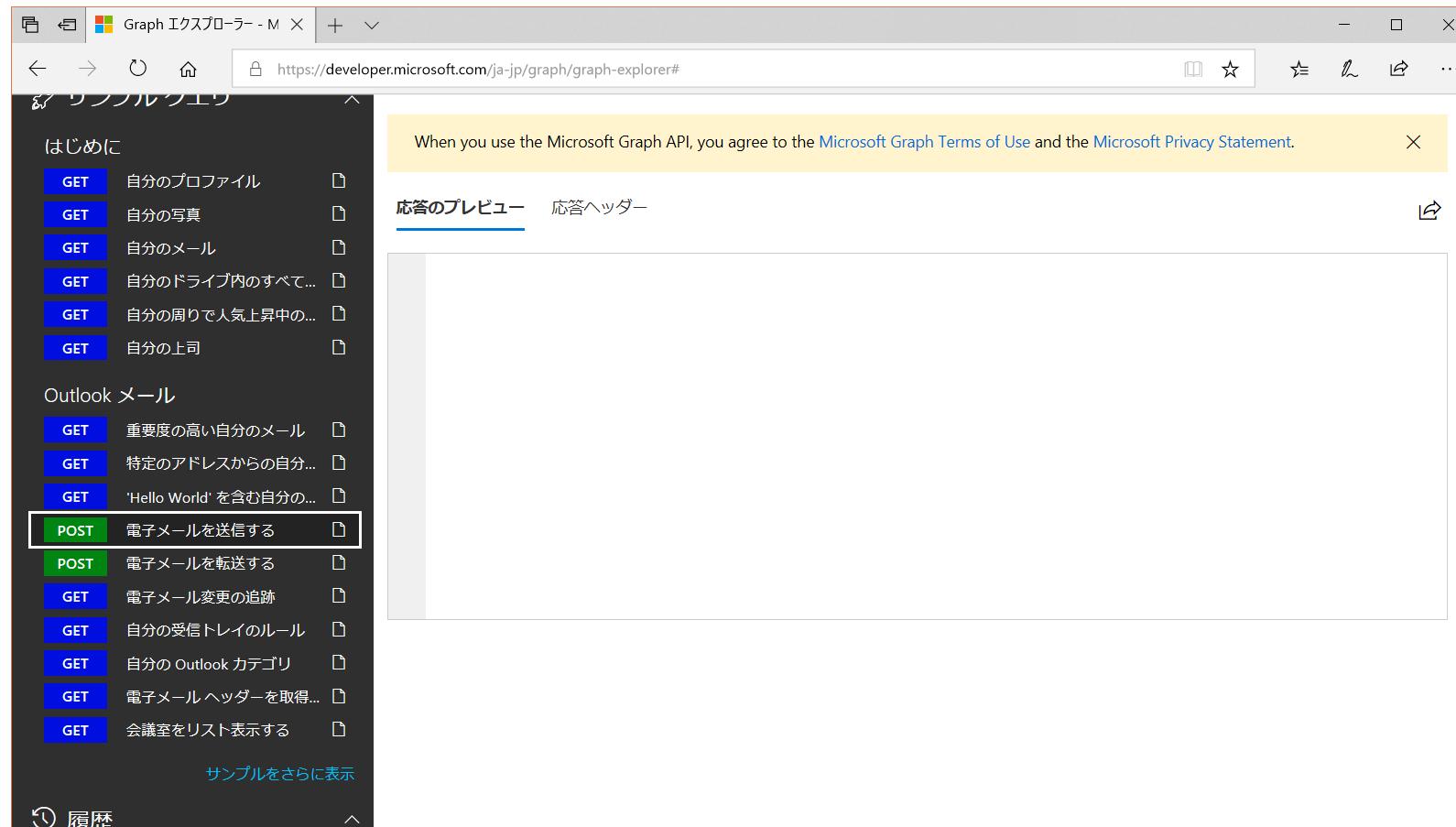
## [Outlook メール] をオンに設定

The screenshot shows the Microsoft Graph Explorer interface. On the left, there's a sidebar with sections like 'サンプル クエリ' and 'Outlook メール'. The main area has a 'GET' button, a dropdown for 'v1.0', and a URL field. A message box says 'When you use the Microsoft Graph API, you agree to the Microsoft Terms of Use'. On the right, a sidebar titled 'サンプル カテゴリ' lists various categories with toggle switches. The 'Outlook メール' category is selected, indicated by a blue outline around its row.

サンプル カテゴリ	状態
はじめに (6)	オン
ユーザー (8)	オフ
グループ (8)	オフ
Outlook メール (10)	オン
Outlook メール (ベータ版) (1)	オフ
Outlook カレンダー (6)	オフ
個人用連絡先 (2)	オフ
OneDrive (5)	オフ
Excel (7)	オフ
Planner (13)	オフ
Insights (1)	オフ
Insights (ベータ版) (3)	オフ
連絡先 (3)	オフ
拡張機能 (7)	オフ
OneNote (6)	オフ
SharePoint サイト (7)	オフ

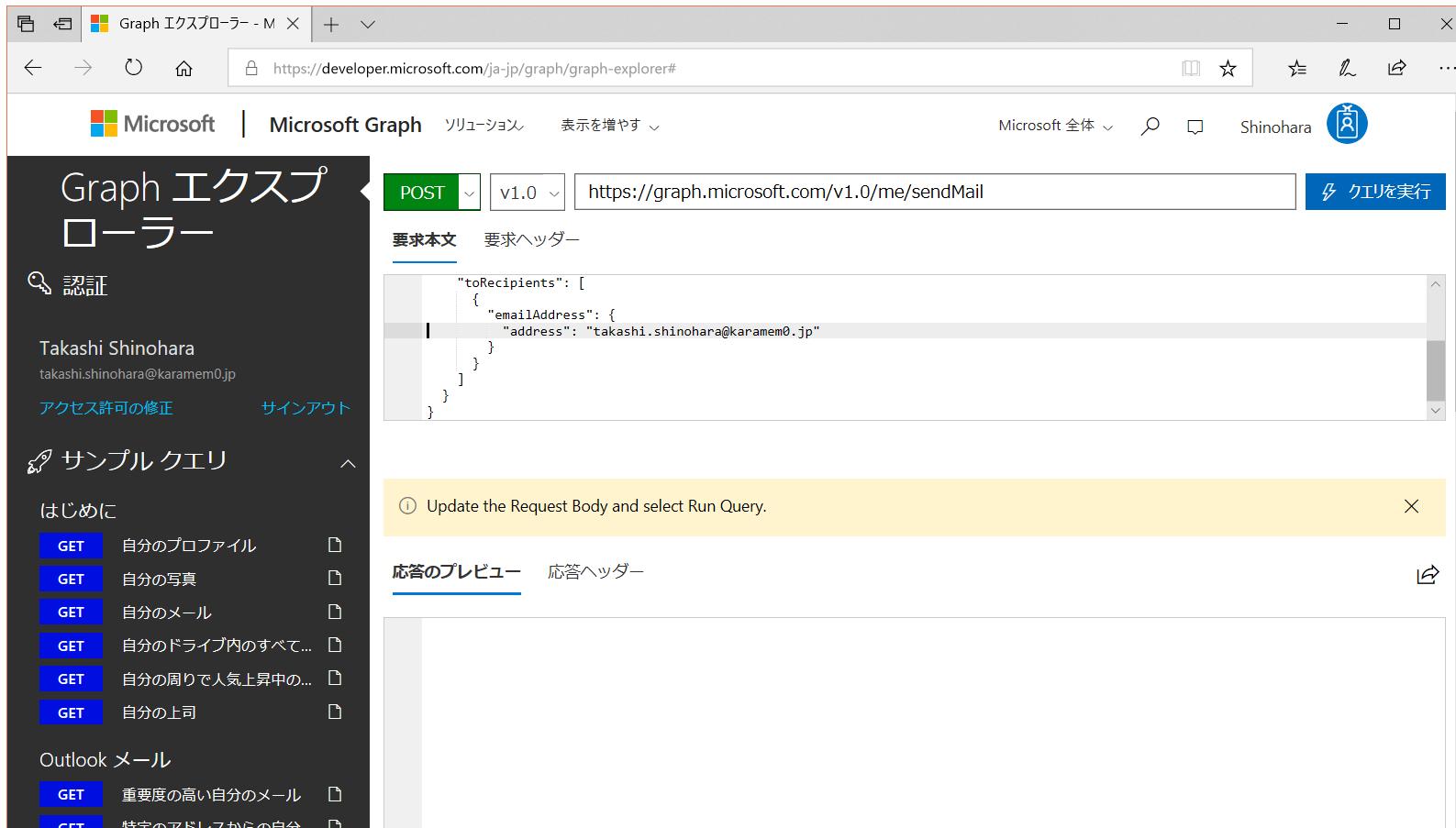
# メールの送信: サンプルの選択

[電子メールを送信する] を選択



# メールの送信: 宛先の変更

宛先を自分のメール アドレスに変更して実行



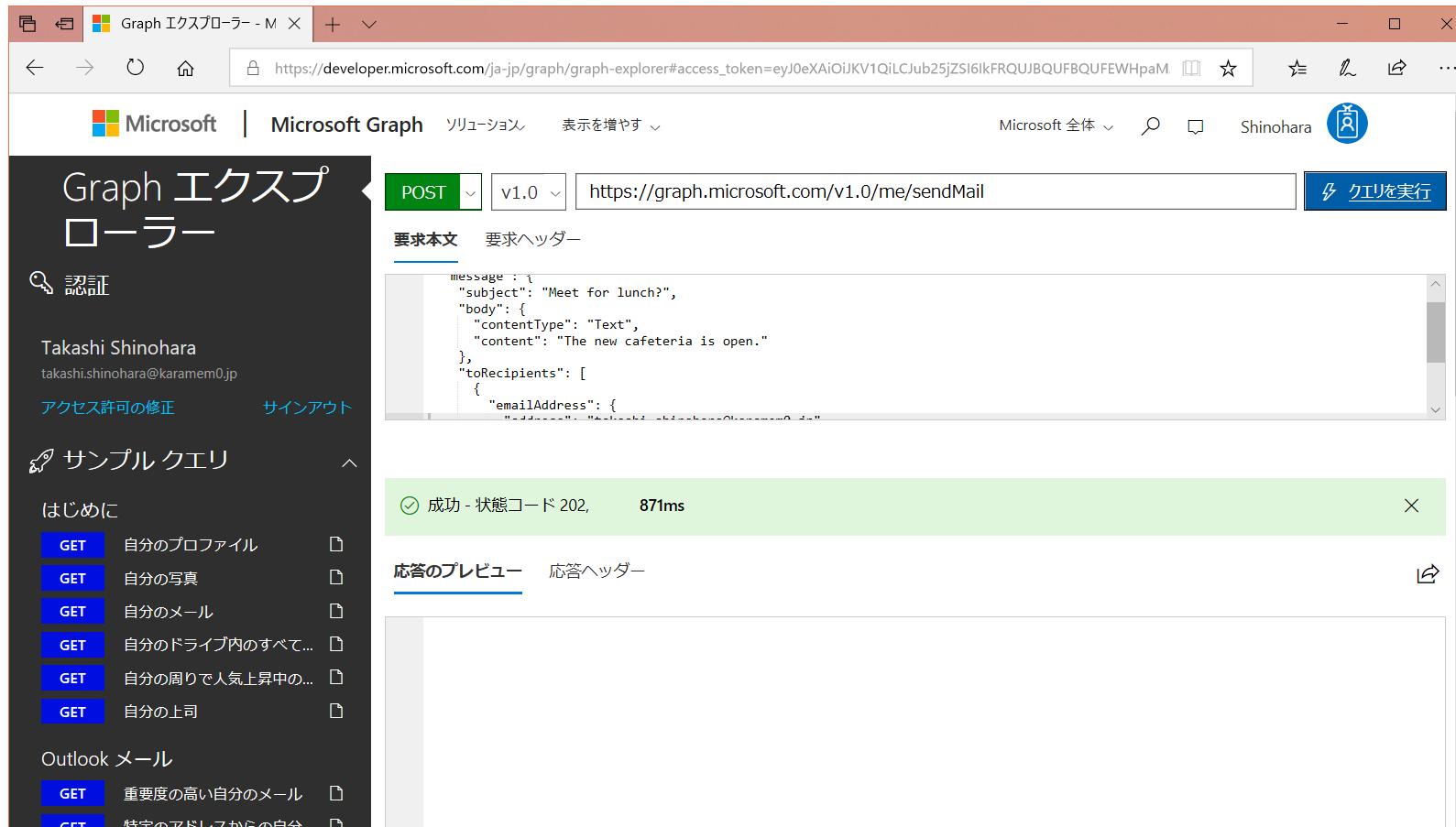
The screenshot shows the Microsoft Graph Explorer interface. The top navigation bar includes the Microsoft logo, 'Microsoft Graph' and 'ソリューション' (Solution) buttons, a search bar, and a user profile for 'Shinohara'. The main area is titled 'Graph エクスプローラー' (Graph Explorer). A 'POST' button is selected, and the URL 'https://graph.microsoft.com/v1.0/me/sendMail' is entered in the address bar. Below the URL, there are tabs for '要求本文' (Request Body) and '要求ヘッダー' (Request Headers). The '要求本文' tab contains the following JSON code:

```
POST v1.0 https://graph.microsoft.com/v1.0/me/sendMail
{
  "toRecipients": [
    {
      "emailAddress": {
        "address": "takashi.shinohara@karamem0.jp"
      }
    }
  ]
}
```

A yellow banner at the bottom of the request body editor says 'Update the Request Body and select Run Query.' To the right of the request body editor, there are tabs for '応答のプレビュー' (Response Preview) and '応答ヘッダー' (Response Headers), both of which are currently empty.

# メールの送信: 実行結果の確認

## 応答の状態コードの確認



The screenshot shows the Microsoft Graph Explorer interface. The top navigation bar includes the Microsoft logo, 'Microsoft Graph' title, and a search bar. The main area has a 'POST' dropdown set to 'v1.0' and a URL input field containing 'https://graph.microsoft.com/v1.0/me/sendMail'. Below this, there are two tabs: '要求本文' (Request Body) and '要求ヘッダー' (Request Headers). The '要求本文' tab contains a JSON payload for sending an email:

```
message : {
  "subject": "Meet for lunch?",
  "body": {
    "contentType": "Text",
    "content": "The new cafeteria is open."
  },
  "toRecipients": [
    {
      "emailAddress": {
        "address": "jdoe@contoso.com"
      }
    }
  ]
}
```

At the bottom, a green success message box displays: '成功 - 状態コード 202, 871ms'. The '応答のプレビュー' (Response Preview) section is currently empty.

# メールの送信: Graph での受信メールの確認

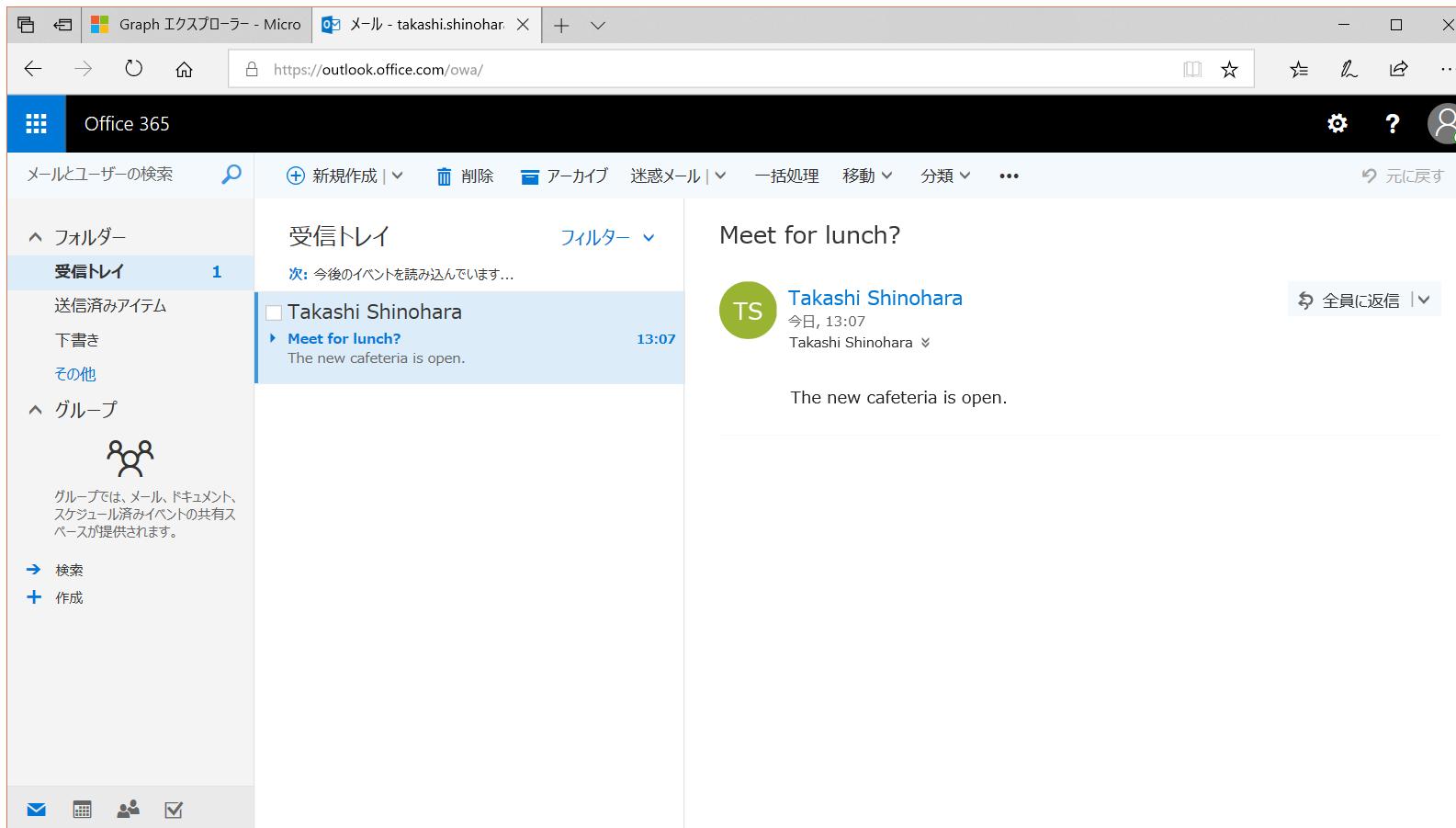
/v1.0/me/messages?\$filter=isRead eq false を実行

The screenshot shows the Microsoft Graph Explorer interface. In the top navigation bar, the URL is https://developer.microsoft.com/ja-jp/graph/graph-explorer#. The main area displays a GET request to https://graph.microsoft.com/v1.0/me/messages?\$filter=isRead eq false. The response status is "成功 - 状態コード 200, 175ms". The response body is a JSON object representing an unread message:

```
    "@odata.etag": "W/\"CQAAABYAAACXgqpKgFnEQ6wt17NYvwRHAAu9nqT\\\"",  
    "id": "AAMkAGE10GQ00GQ1LTNhOGQtNDhmNC1iYTmLTeZGI3YW4ZDcxYwBGAAAAADgX3wOGbUhQou9aMhiUGLzBwCXgqpKgFnEQ6wt17NYvwRHAAUAA",  
    "createdDateTime": "2018-09-13T04:07:35Z",  
    "lastModifiedDateTime": "2018-09-13T04:07:36Z",  
    "changeKey": "CQAAABYAAACXgqpKgFnEQ6wt17NYvwRHAAu9nqT",  
    "categories": [],  
    "receivedDateTime": "2018-09-13T04:07:35Z",  
    "sentDateTime": "2018-09-13T04:07:34Z",  
    "hasAttachments": false,  
    "internetMessageId": "<052PR01MB1131DDF7F85302DC3C9406E6A81A0@052PR01MB1131.jpnprd01.prod.outlook.com>",  
    "subject": "Meet for lunch?",  
    "bodyPreview": "The new cafeteria is open."
```

# メールの送信: Outlook での受信メールの確認

## Outlook から受信メールを確認



# Graph Explorer

サンプル データを表示する

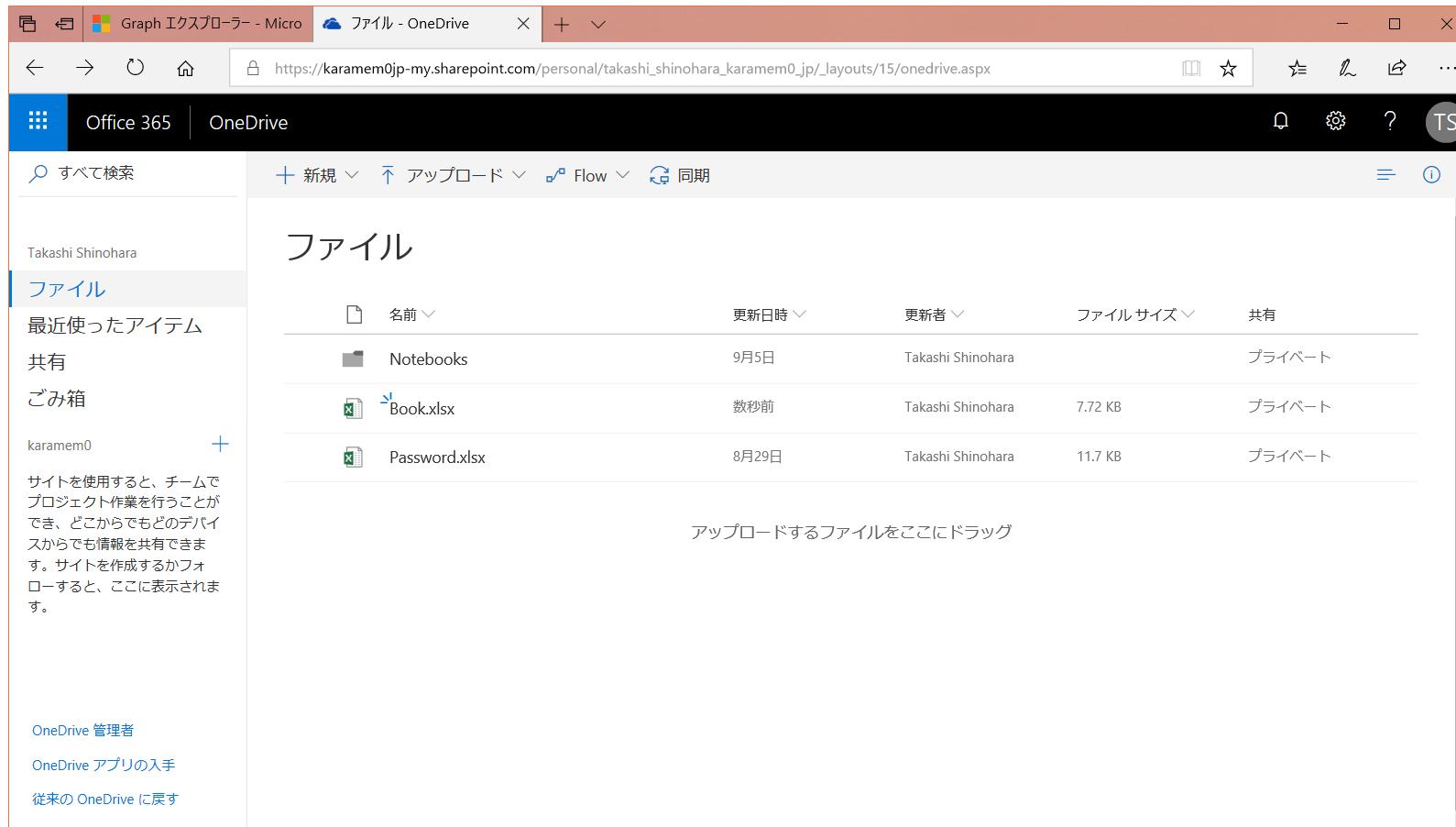
自分のユーザー情報を取得する

メールを送信する

OneDrive の Excel ファイルに値を書き込む

# Excel 書き込み: ファイルの準備

## OneDrive for Business に Excel ファイルを作成



The screenshot shows the OneDrive for Business web interface. The left sidebar includes links for 'すべて検索', '新規', 'アップロード', 'Flow', '同期', 'ファイル' (selected), '最近使ったアイテム', '共有', 'ごみ箱', and 'karamemo'. The main area displays a list of files:

名前	更新日時	更新者	ファイル サイズ	共有
Notebooks	9月5日	Takashi Shinohara		プライベート
Book.xlsx	数秒前	Takashi Shinohara	7.72 KB	プライベート
Password.xlsx	8月29日	Takashi Shinohara	11.7 KB	プライベート

A message at the bottom says 'アップロードするファイルをここにドラッグ'.

# Excel 書き込み: ファイルの検索

/v1.0/me/drive/root/search(q='{query}') を実行し id を保管

The screenshot shows the Microsoft Graph Explorer interface. In the top navigation bar, the URL is https://developer.microsoft.com/ja-jp/graph/graph-explorer#. The main area displays a GET request to https://graph.microsoft.com/v1.0/me/drive/root/search(q='Book.xlsx'). The response status is '成功 - 状態コード 200, 1906ms'. The response body is a JSON object representing a collection of drive items, with one item matching the query 'Book.xlsx'.

```
{
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#Collection(driveItem)",
  "value": [
    {
      "@odata.type": "#microsoft.graph.driveItem",
      "createdDateTime": "2018-09-13T04:22:09Z",
      "id": "01HZPNDZ3FPYXSSIO4BDLWECL2OABLMD",
      "lastModifiedDateTime": "2018-09-13T04:22:14Z",
      "name": "Book.xlsx",
      "webUrl": "https://karamem0jp-my.sharepoint.com/personal/takashi_shinohara_karamem0_jp/_layouts/15/Doc.aspx?sourcedoc=%7B01HZPNDZ3FPYXSSIO4BDLWECL2OABLMD%7D&action=Edit",
      "size": 7904,
      "createdBy": {
        "name": "takashi.shinohara@karamem0.jp"
      }
    }
  ]
}
```

# Excel 書き込み: セッションの開始

/v1.0/me/drive/items/{itemid}/workbook/createSession を実行

The screenshot shows the Microsoft Graph Explorer interface. The URL in the address bar is `https://developer.microsoft.com/ja-jp/graph/graph-explorer#`. The request method is set to `POST`, the version to `v1.0`, and the endpoint to `/items/01HZPWDZ73FPYXS5IO4BDLWECL2OABLMD/workbook/createSession`. A large green success message at the bottom indicates a `成功 - 状態コード 201, 2878ms`. The response body shows the generated session ID and other metadata.

```
{ "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#microsoft.graph.workbookSessionInfo", "id": "cluster=JP28session=12.c3033077c1081.A268.1.E185.http%3a%2f%2ftier0%3fid%3dhttps%253a%252f%252Fkaramem0jp%252Dmy%252Esharepoint%252Ecom", "persistChanges": true }
```



# Excel 書き込み: セッション情報の設定

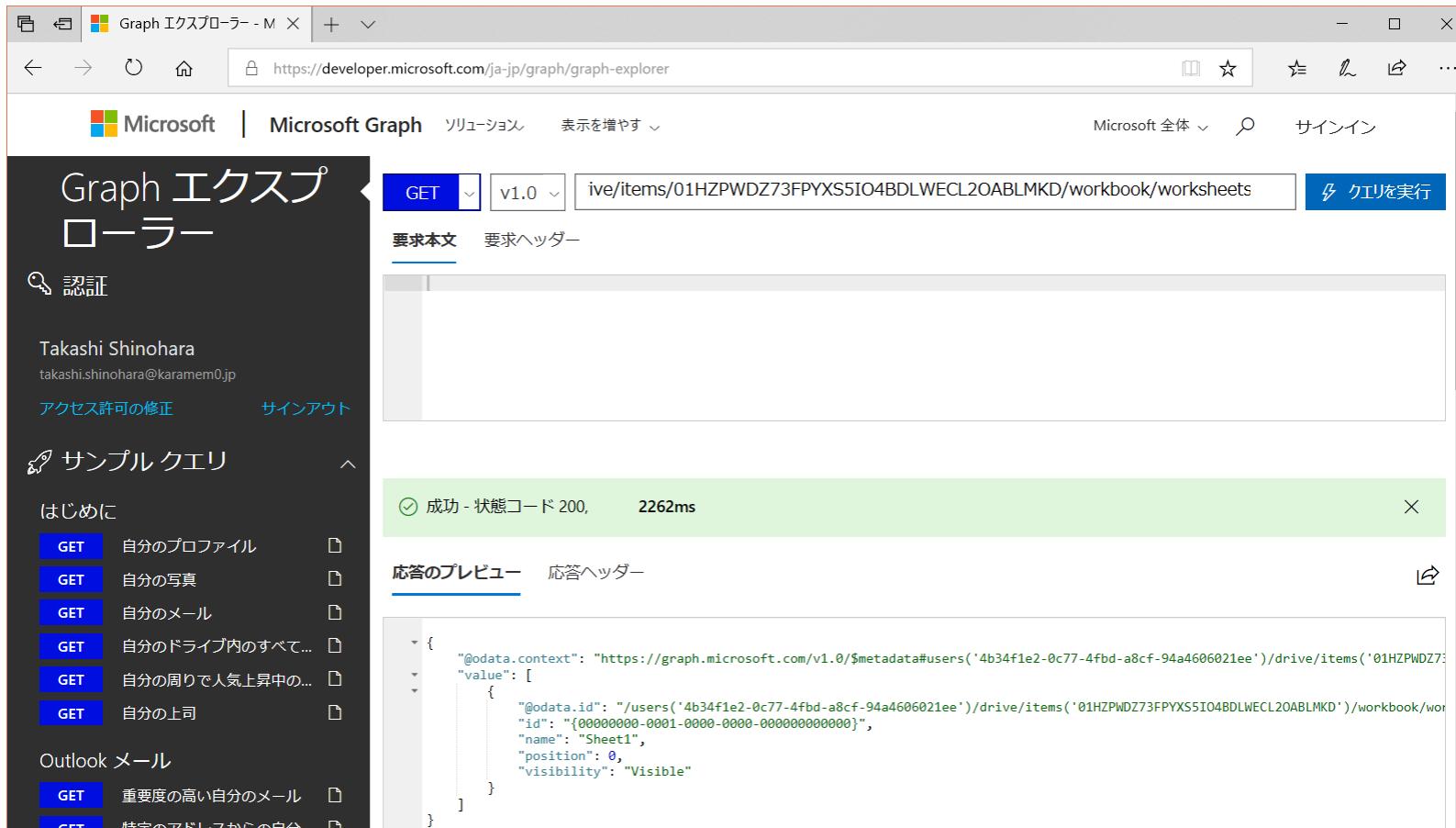
要求ヘッダーに workbook-session-id を設定

The screenshot shows the Microsoft Graph Explorer interface. The URL is `https://developer.microsoft.com/ja-jp/graph/graph-explorer`. The method is set to `POST` and the endpoint is `https://graph.microsoft.com/v1.0/me/drive/items/01HZPWDZ73FPYXS5IO4BD`. The `Content-type` header is set to `application/json`. The `workbook-session-id` header is set to `cluster=JP2&session=12.21e29decfd851.A268.1.E185.http%3a%`. The response shows a success status code 201 with a response time of 2559ms. The response body is a JSON object containing the session information.

```
{ "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#microsoft.graph.workbookSessionInfo", "id": "cluster=JP2&session=12.21e29decfd851.A268.1.E185.http%3a%2f%2ftier0%3fid%3dhttps%253A%252F%252Fkaramem0jp%252Dmy%252Esharepoint%252Ecom", "persistChanges": true }
```

# Excel 書き込み: ワークシートの取得

/v1.0/me/drive/items/{itemid}/workbook/worksheets を実行



The screenshot shows the Microsoft Graph Explorer interface. The URL in the address bar is `https://developer.microsoft.com/ja-jp/graph/graph-explorer`. The request method is set to `GET` and the version is `v1.0`. The endpoint is `me/drive/items/01HZPWDZ73FPYXS5IO4BDLWECL2OABLMD/workbook/worksheets`. A green success message at the bottom indicates a status code of 200 and a response time of 2262ms. The response body is displayed in JSON format:

```
{  
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users('4b34f1e2-0c77-4fb1-a8cf-94a4606021ee')/drive/items('01HZPWDZ73FPYXS5IO4BDLWECL2OABLMD')/workbook/worksheets",  
  "value": [  
    {  
      "@odata.id": "/users('4b34f1e2-0c77-4fb1-a8cf-94a4606021ee')/drive/items('01HZPWDZ73FPYXS5IO4BDLWECL2OABLMD')/workbook/worksheets('00000000-0001-0000-0000-000000000000')",  
      "id": "00000000-0001-0000-0000-000000000000",  
      "name": "Sheet1",  
      "position": 0,  
      "visibility": "Visible"  
    }  
  ]  
}
```

# Excel 書き込み: セルに書き込み

/v1.0/me/drive/items/{itemid}/workbook/worksheets/{sheetname}/range(address='A1:A1') を実行

要求本文:

```
{  
  "values" : [ ["Hello Microsoft Graph!"] ],  
}
```

# Excel 書き込み: 実行結果の確認

## 応答の状態コードの確認

The screenshot shows the Microsoft Graph Explorer interface. In the top navigation bar, the URL is https://developer.microsoft.com/ja-jp/graph/graph-explorer#. The main area displays a PATCH request to the endpoint O4BDLWECL2OABLMKD/workbook/worksheets/Sheet1/range(address='A1:A1'). The request body contains the JSON payload:

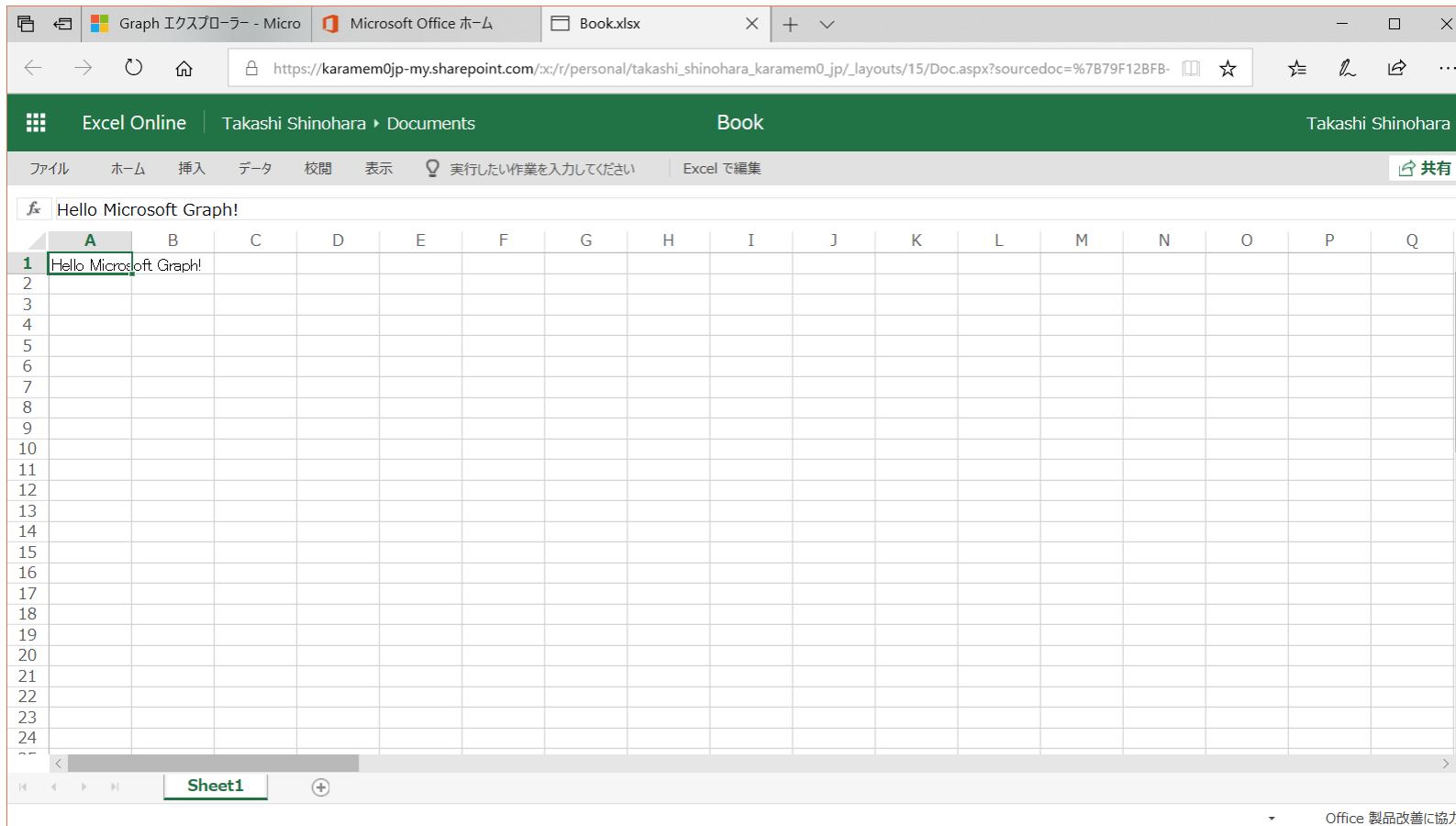
```
{
  "values": [["Hello Microsoft Graph!"]]
}
```

Below the request, a green success message indicates "成功 - 状態コード 200, 1156ms". The response preview pane shows the detailed JSON response from the API.

```
{
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#workbookRange",
  "@odata.type": "#microsoft.graph.workbookRange",
  "@odata.id": "/users('4b34f1e2-0c77-4fb1-a8cf-94a4606021ee')/drive/items('01HZPWDZ3FPYX55I04BDLWECL2OABLMKD')/workbook/worksheets('Sheet1')/range(address='A1:A1',addressLocal:'Sheet1!A1',cellCount:1,columnCount:1,columnHidden:false,columnIndex:0,formulas:[],isText:1,rowCount:1,selectedCellIndex:1,selectedCellAddress:'A1')",
  "id": "01HZPWDZ3FPYX55I04BDLWECL2OABLMKD!A1"
}
```

# Excel 書き込み: Excel ファイルの確認

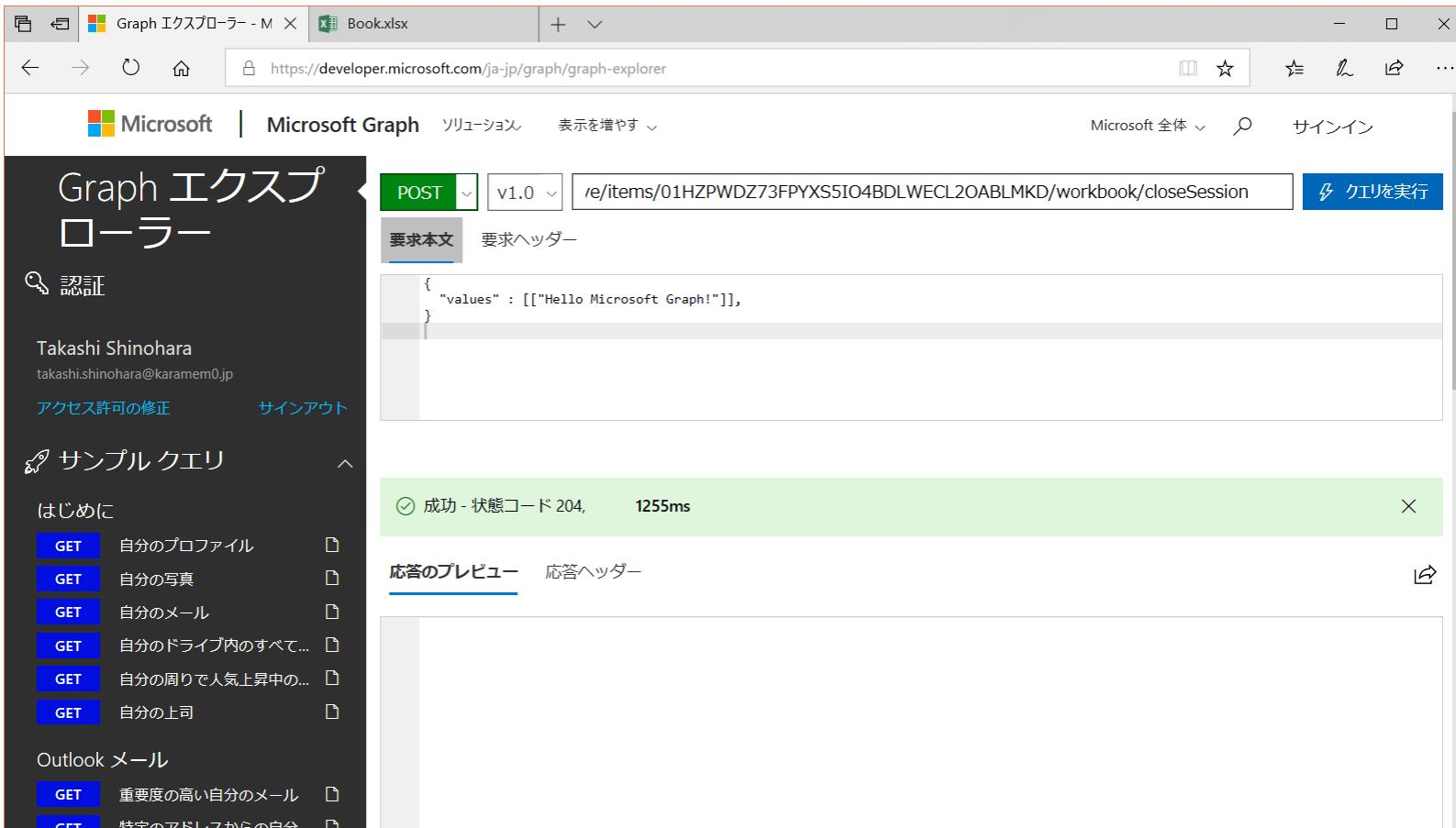
Excel ファイルに値が書き込まれていることを確認



The screenshot shows a Microsoft Excel Online spreadsheet titled "Book.xlsx". The URL in the browser address bar is [https://karamem0jp-my.sharepoint.com/:x/r/personal/takashi\\_shinohara\\_karamem0.jp/\\_layouts/15/Doc.aspx?sourcedoc=%7B79F12BFB-](https://karamem0jp-my.sharepoint.com/:x/r/personal/takashi_shinohara_karamem0.jp/_layouts/15/Doc.aspx?sourcedoc=%7B79F12BFB-). The spreadsheet has one sheet named "Sheet1". Cell A1 contains the text "Hello Microsoft Graph!". The rest of the cells in the first column are empty, with row numbers 2 through 24 visible.

# Excel 書き込み: セッションの終了

/v1.0/me/drive/items/{itemid}/workbook/closeSession を実行



The screenshot shows the Microsoft Graph Explorer interface. The URL in the address bar is `https://developer.microsoft.com/ja-jp/graph/graph-explorer`. The request method is set to `POST`, version `v1.0`, and the endpoint is `/e/items/01HZPWDZ73FPYXS5IO4BDLWECL2OABLMDK/workbook/closeSession`. The "Request Body" tab contains the following JSON:

```
{ "values" : [["Hello Microsoft Graph!"]], }
```

The "Response Preview" section shows a green success message: "成功 - 状態コード 204, 1255ms".

The left sidebar lists various sample queries under "サンプル クエリ", including "自分のプロファイル", "自分の写真", "自分のメール", etc.



OAuth とは

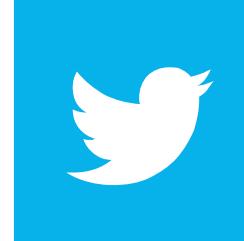
# OAuth

サード パーティー アプリケーションによる HTTP サービスへの限定期的なアクセスを可能にする認可フレームワーク

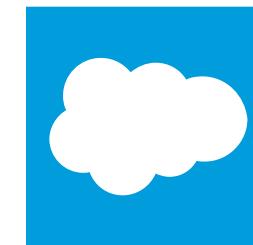
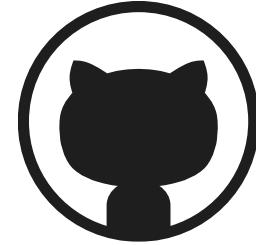
サード パーティー アプリケーションに対して資格情報 (ユーザー名とパスワードのペア) を共有することなく認可を行うことができる

# さまざまなサービスでの実装

OAuth 1.0



OAuth 2.0



# RFC

RFC 6749: The OAuth 2.0 Authorization Framework

RFC 6750: The OAuth 2.0 Authorization Framework: Bearer Token Usage

RFC 8252: OAuth 2.0 for Native Apps

`draft-ietf-oauth-device-flow`

`draft-ietf-oauth-jwt-bearer-12`

# ロールの種類

Resource Owner

Resource Server

Client

Authorization Server

# 認可フローの種類

Authorization Code Grant

Implicit Grant

Password Credentials Grant

Client Credentials Grant

Device Flow

On-Behalf-Of Flow

# Microsoft Graph における OAuth

# アクセス許可

	委任されたアクセス許可	アプリケーションのアクセス許可
認可フロー	Authorization Code Grant Implicit Grant Password Credentials Grant Device Flow On-Behalf-Of Flow	Client Credentials Grant
用途	対話型のアプリケーション	バックグランド アプリケーション
アクセス許可	Resource Owner (ユーザー) が持っている権限とアプリに付与された権限の最小範囲	完全なレベルの権限
/me の使用	可	不可



# エンドポイント

	v1.0	v2.0
URL	<a href="https://login.microsoftonline.com/oauth2">https://login.microsoftonline.com/oauth2</a>	<a href="https://login.microsoftonline.com/oauth2/v2.0">https://login.microsoftonline.com/oauth2/v2.0</a>
アカウント	組織アカウントのみ	組織アカウントおよびマイクロソフトアカウントの両方に対応
アプリの登録	Azure Portal	Application Registration Portal
動的なアクセス許可	不可	可
認可フロー	サポートされている	一部はサポートされていない
ライブラリ	ADAL	MSAL

# アクセストークン

認可が正常に行われたときに返される JSON Web Token (JWT) 形式の文字列

System.IdentityModel.Tokens.Jwt を使って解析できる  
有効期間が存在する (セキュリティ上短くするのが望ましい)

# リフレッシュトークン

有効期間が切れた場合はリフレッシュトークンを使ってアクセストークンを取得しなおす必要がある

認可フローによってはリフレッシュトークンを取得できない

有効期間が存在する (切れた場合は再認証)



# ハンズオン: Hello Microsoft Graph

# Hello Microsoft Graph

アプリの登録

委任されたアクセス許可によるアプリの実行

アプリケーションのアクセス許可によるアプリの実行

SDK を使ったアプリの実行

# Hello Microsoft Graph

## アプリの登録

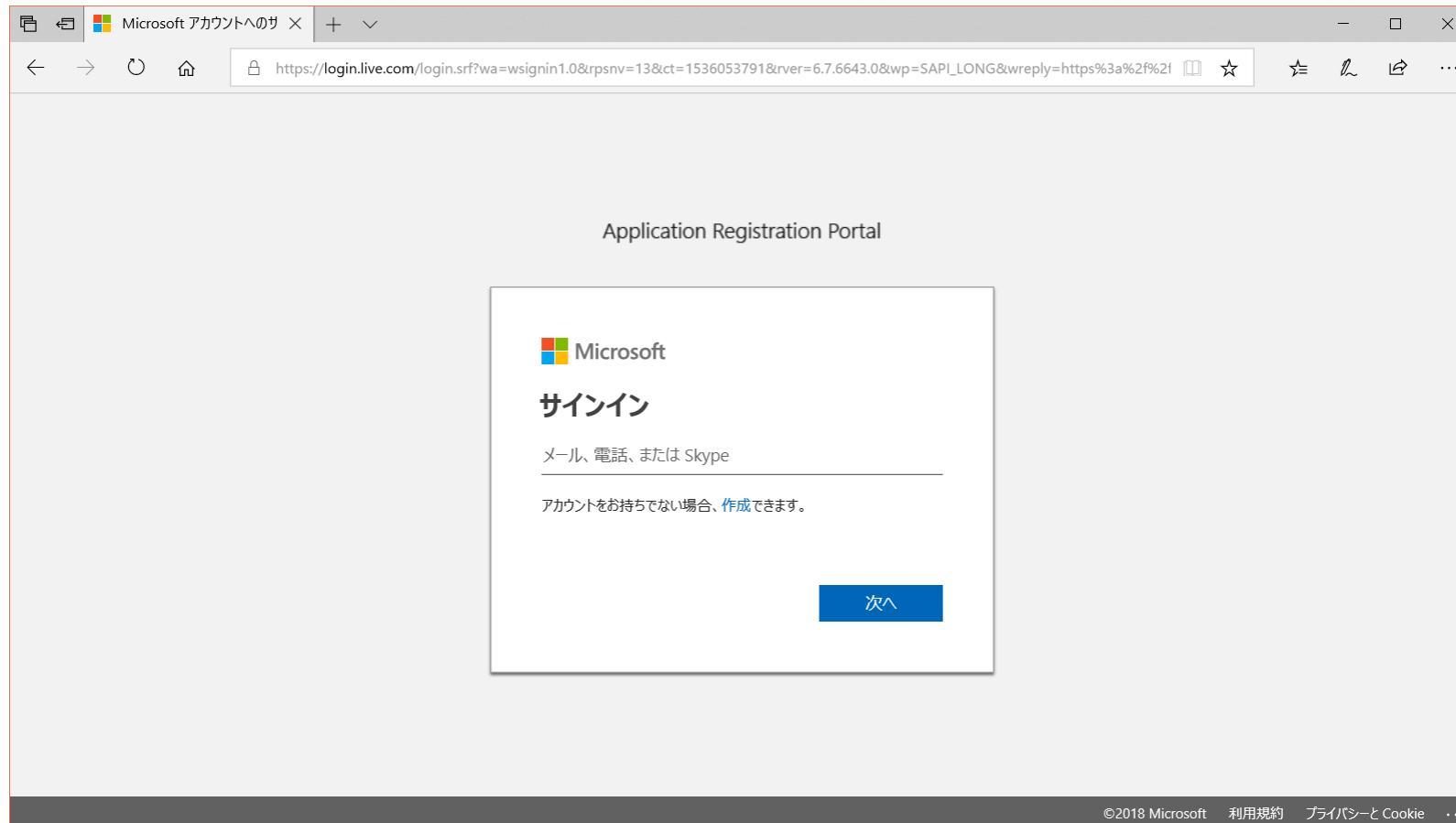
委任されたアクセス許可によるアプリの実行

アプリケーションのアクセス許可によるアプリの実行

SDK を使ったアプリの実行

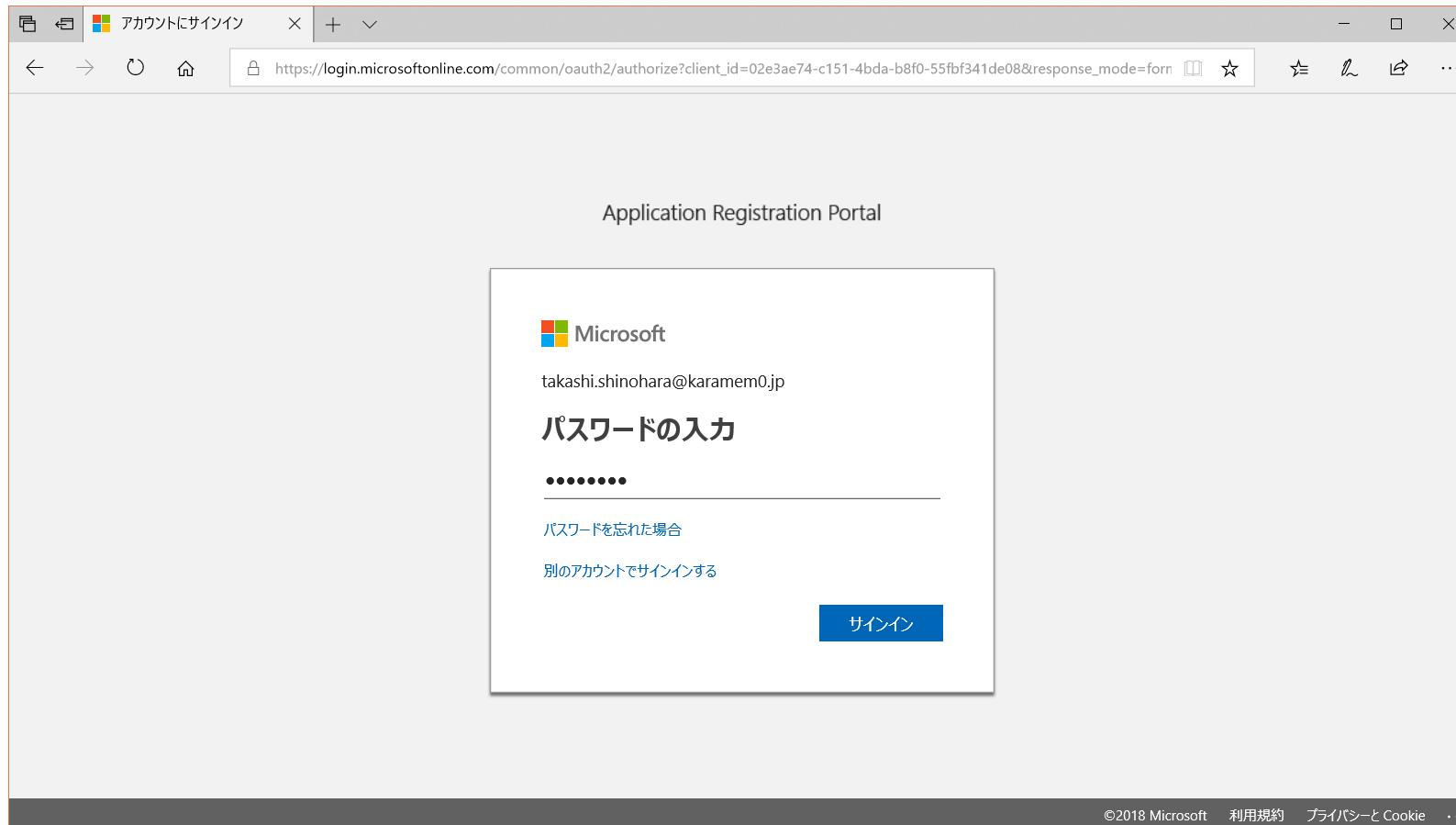
# アプリの登録: Application Registration Portal

<https://apps.dev.microsoft.com> にアクセス



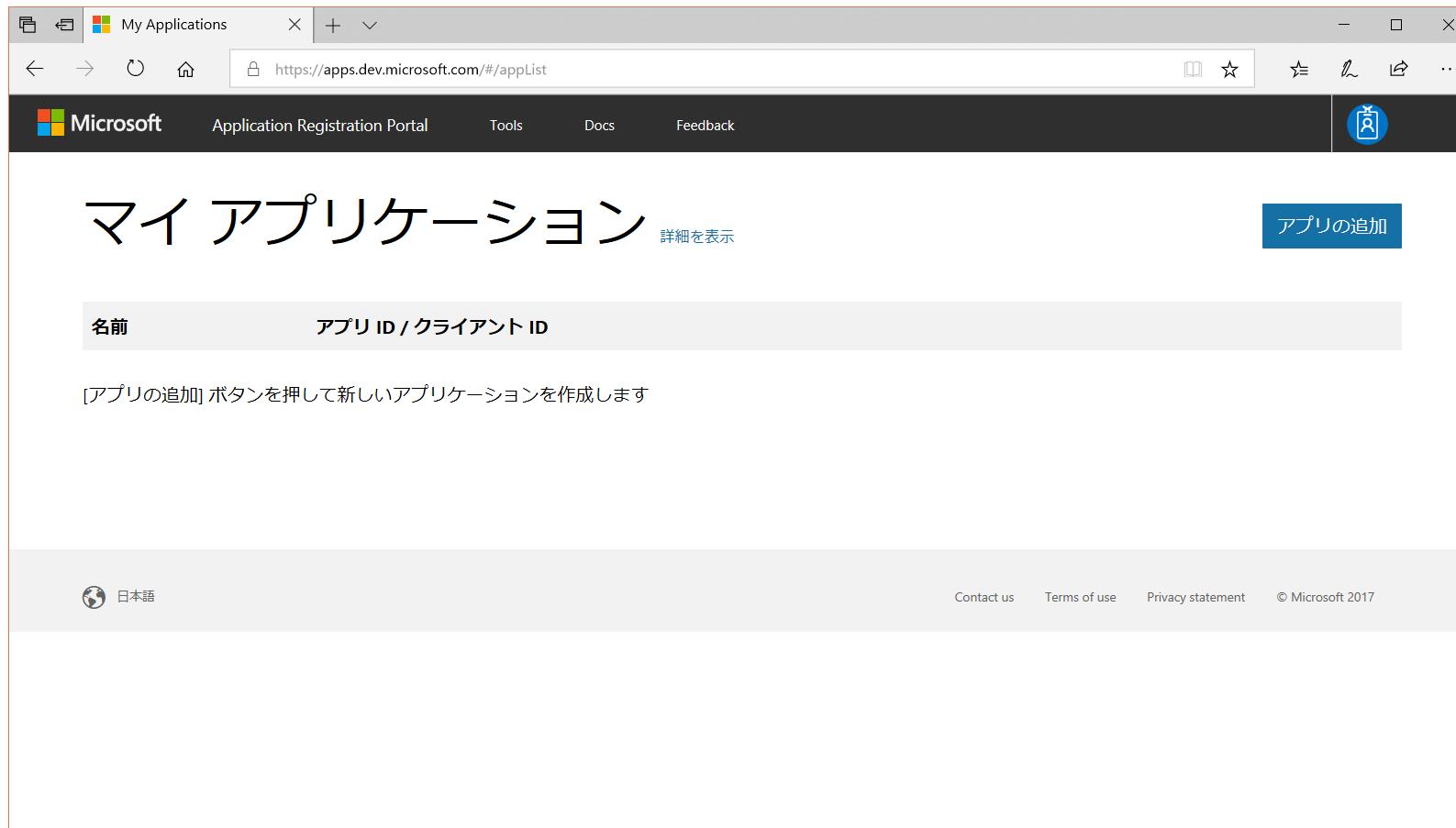
# アプリの登録: サインイン

## 組織アカウントでサインイン



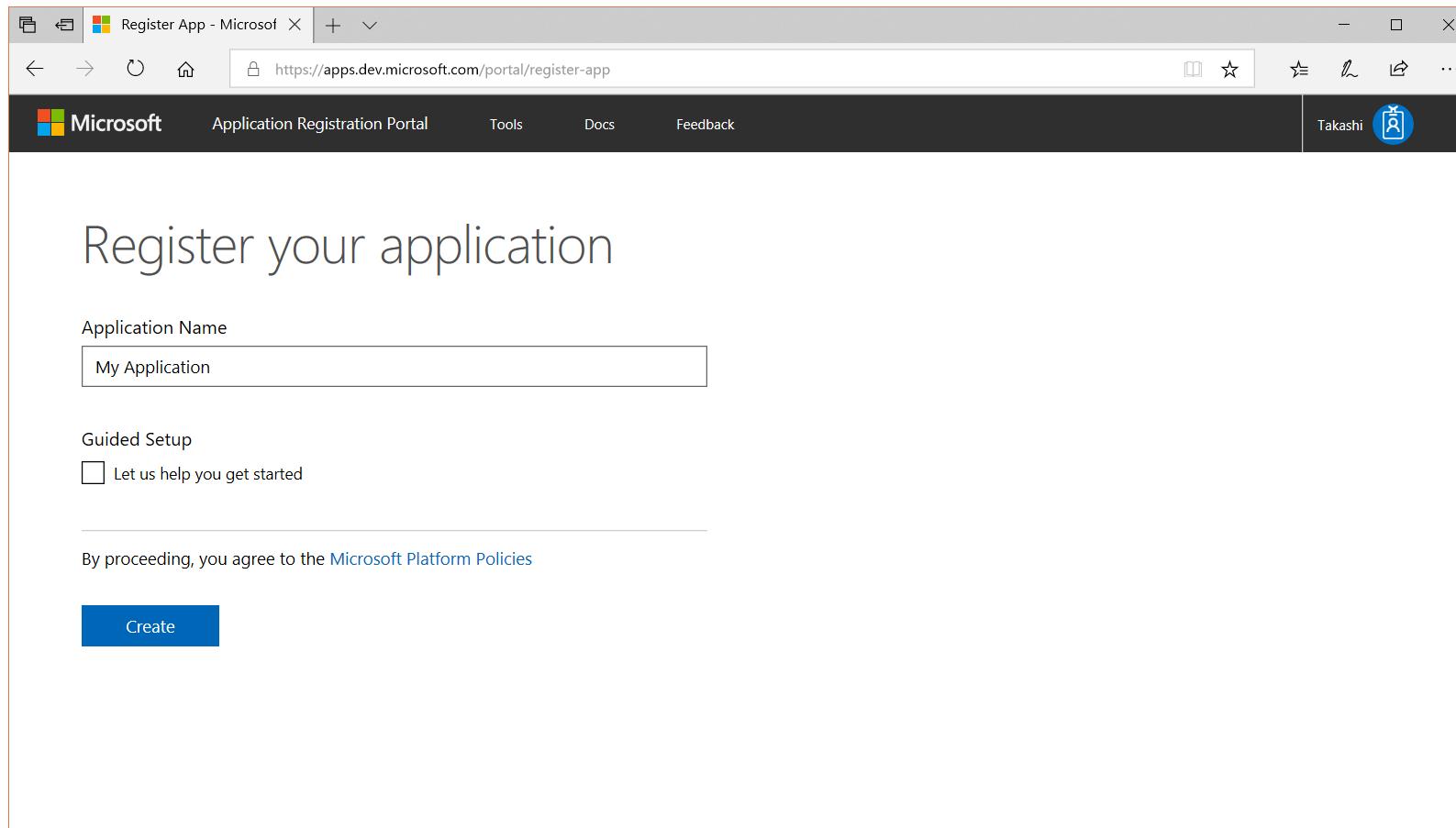
# アプリの登録: アプリの追加

[アプリの追加] をクリック



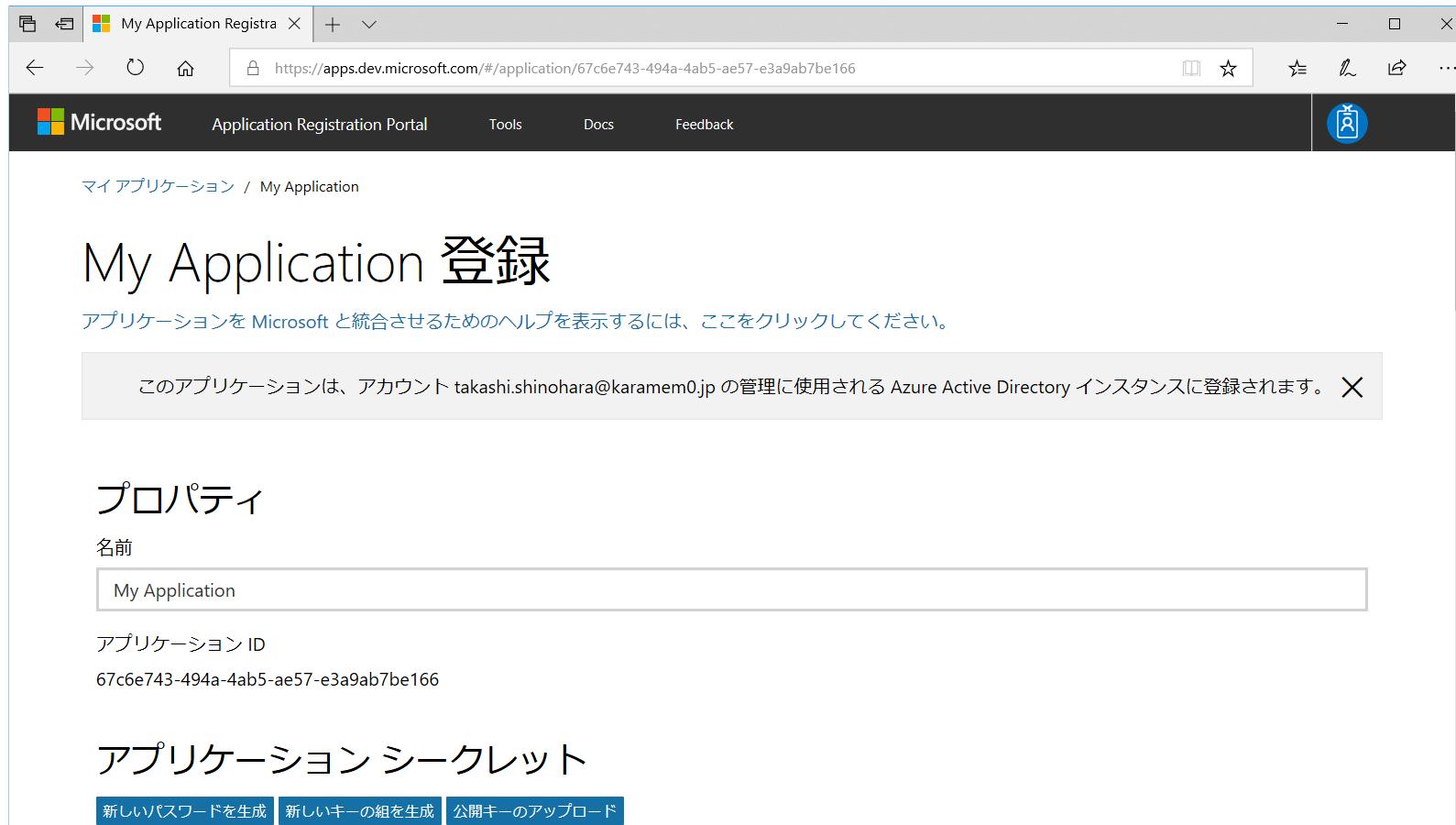
# アプリの登録: アプリケーション名の登録

[Application Name] を入力して [Create] をクリック



# アプリの登録: アプリケーション ID の確認

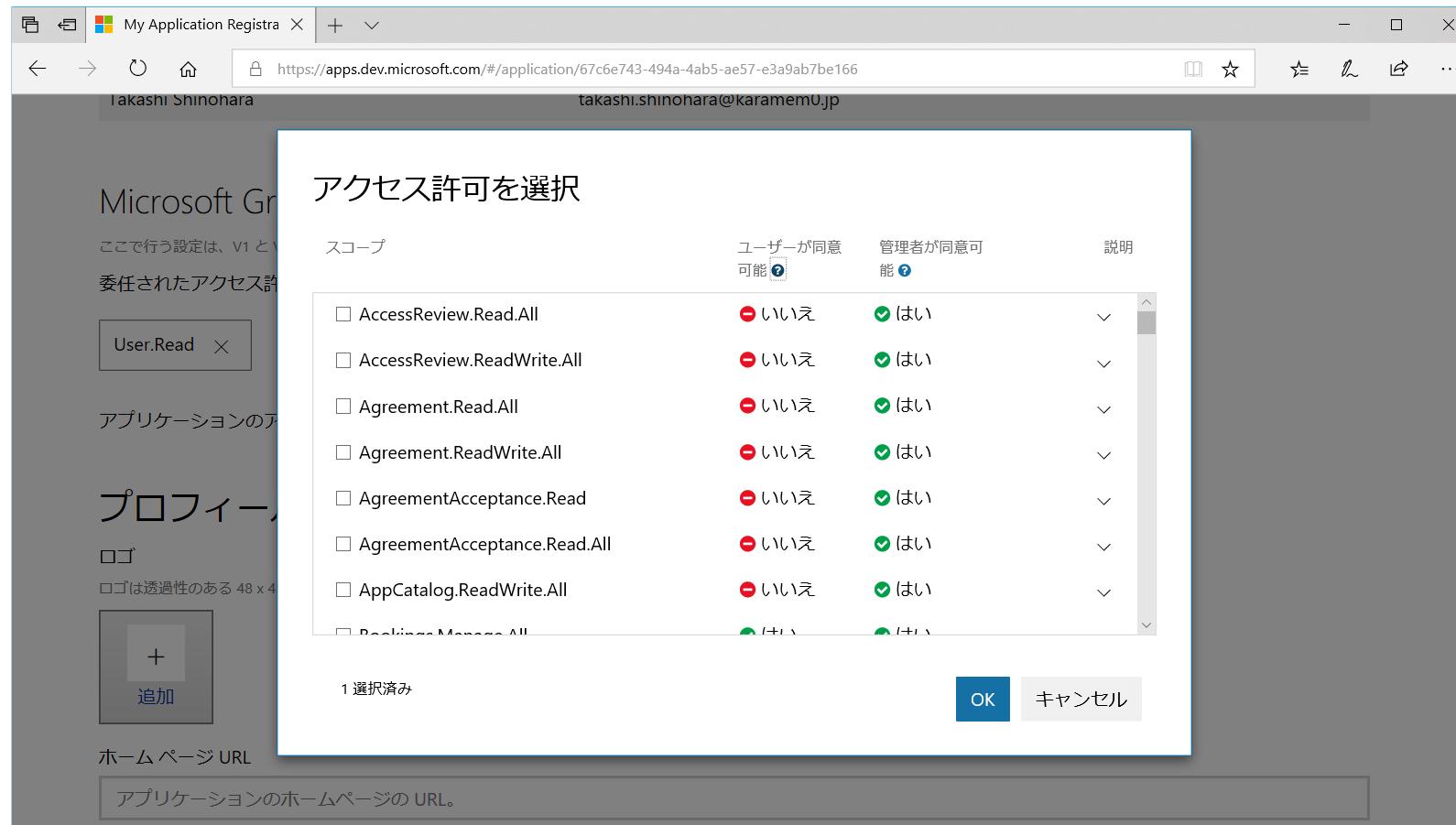
[アプリケーション ID] の値をコピーして保管



The screenshot shows a Microsoft Edge browser window displaying the Microsoft Application Registration Portal at <https://apps.dev.microsoft.com/#/application/67c6e743-494a-4ab5-ae57-e3a9ab7be166>. The page title is "My Application Registration". The main content area is titled "My Application 登録". A message box states: "このアプリケーションは、アカウント takashi.shinohara@karamem0.jp の管理に使用される Azure Active Directory インスタンスに登録されます。" (This application is registered with the Azure Active Directory instance managed by account takashi.shinohara@karamem0.jp). Below this, there is a "Properties" section with a "Name" field containing "My Application" and an "Application ID" field containing "67c6e743-494a-4ab5-ae57-e3a9ab7be166". At the bottom, there is a "Application Secret" section with three buttons: "新しいパスワードを生成" (Generate new password), "新しいキーの組を生成" (Generate new key pair), and "公開キーのアップロード" (Upload public key).

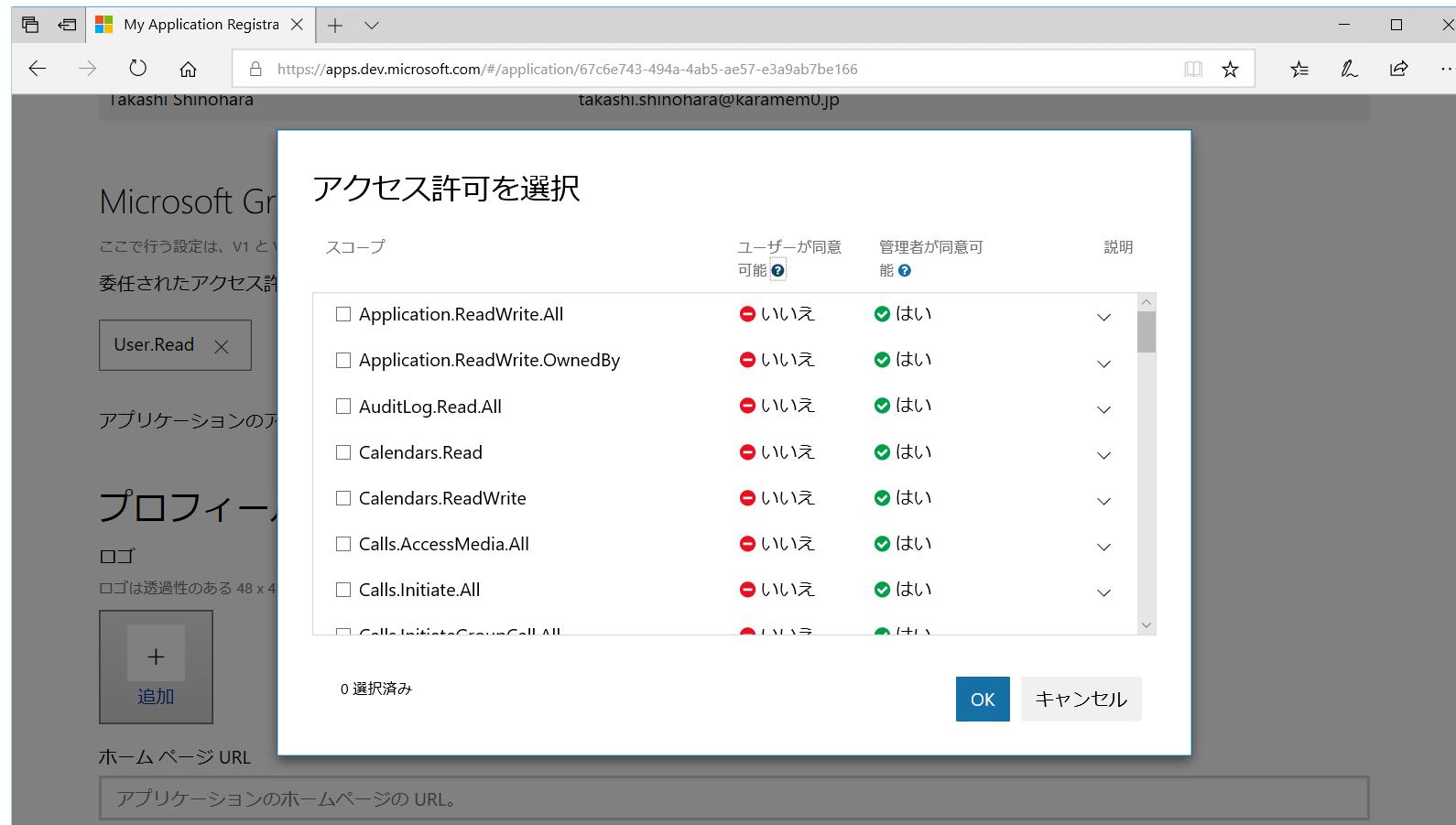
# アプリの登録: 委任されたアクセス許可の確認

[委任されたアクセス許可]-[追加] をクリック



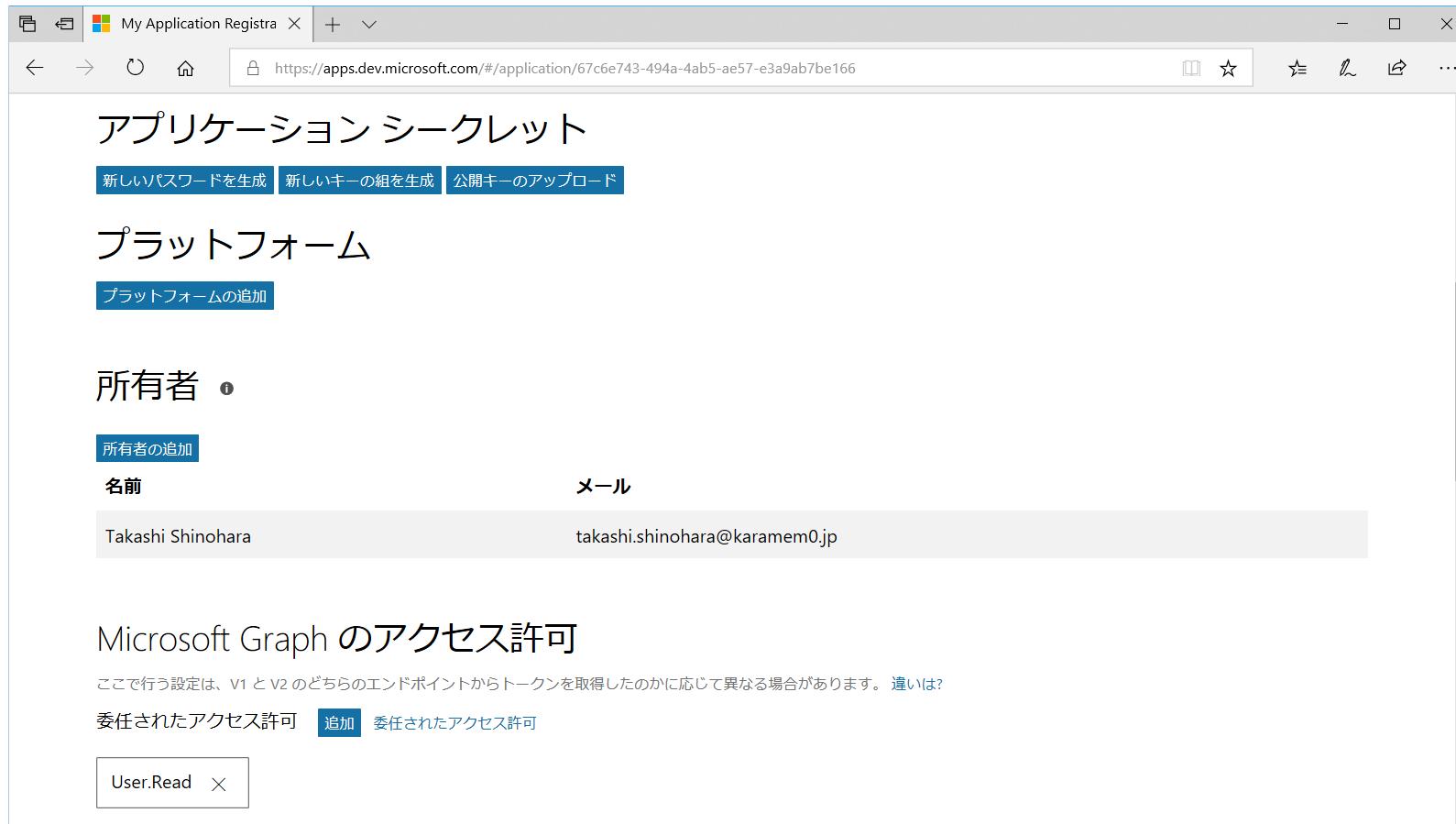
# アプリの登録: アプリケーションのアクセス許可の確認

[アプリケーションのアクセス許可]-[追加] をクリック



# アプリの登録: シークレットの生成

[新しいパスワードを生成] をクリック



My Application Registry

https://apps.dev.microsoft.com/#/application/67c6e743-494a-4ab5-ae57-e3a9ab7be166

## アプリケーションシークレット

新しいパスワードを生成 新しいキーの組を生成 公開キーのアップロード

### プラットフォーム

プラットフォームの追加

### 所有者

所有者の追加

名前	メール
Takashi Shinohara	takashi.shinohara@karamem0.jp

### Microsoft Graph のアクセス許可

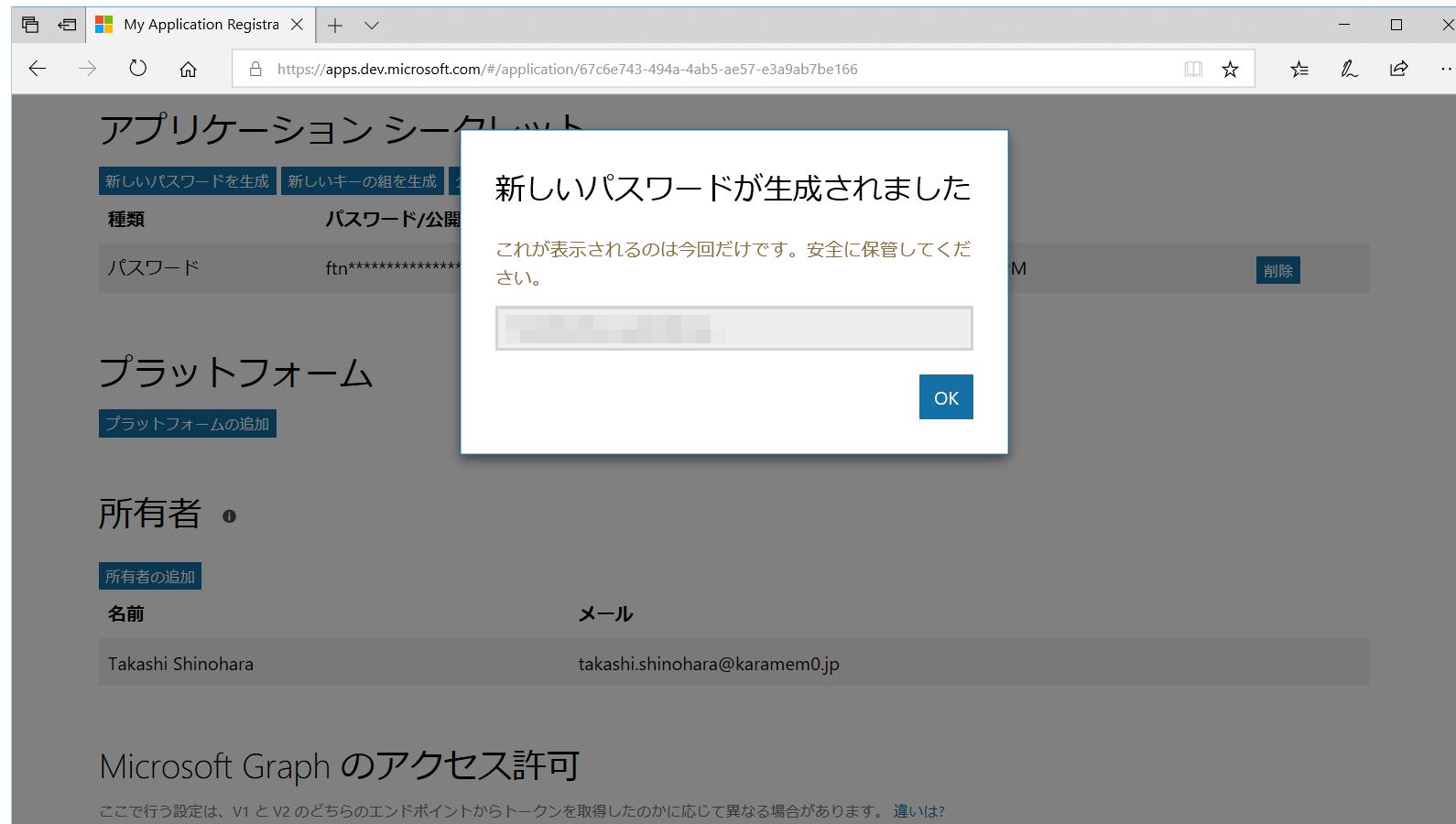
ここで行う設定は、V1 と V2 のどちらのエンドポイントからトークンを取得したのに応じて異なる場合があります。違いは?

委任されたアクセス許可 [追加](#) 委任されたアクセス許可

User.Read ×

# アプリの登録: シークレットの生成

生成されたパスワードをコピーして保管



# アプリの登録: プラットフォームの追加

[プラットフォームの追加] をクリック



The screenshot shows a Microsoft Edge browser window with the URL <https://apps.dev.microsoft.com/#/application/67c6e743-494a-4ab5-ae57-e3a9ab7be166>. The page title is "My Application Registry".

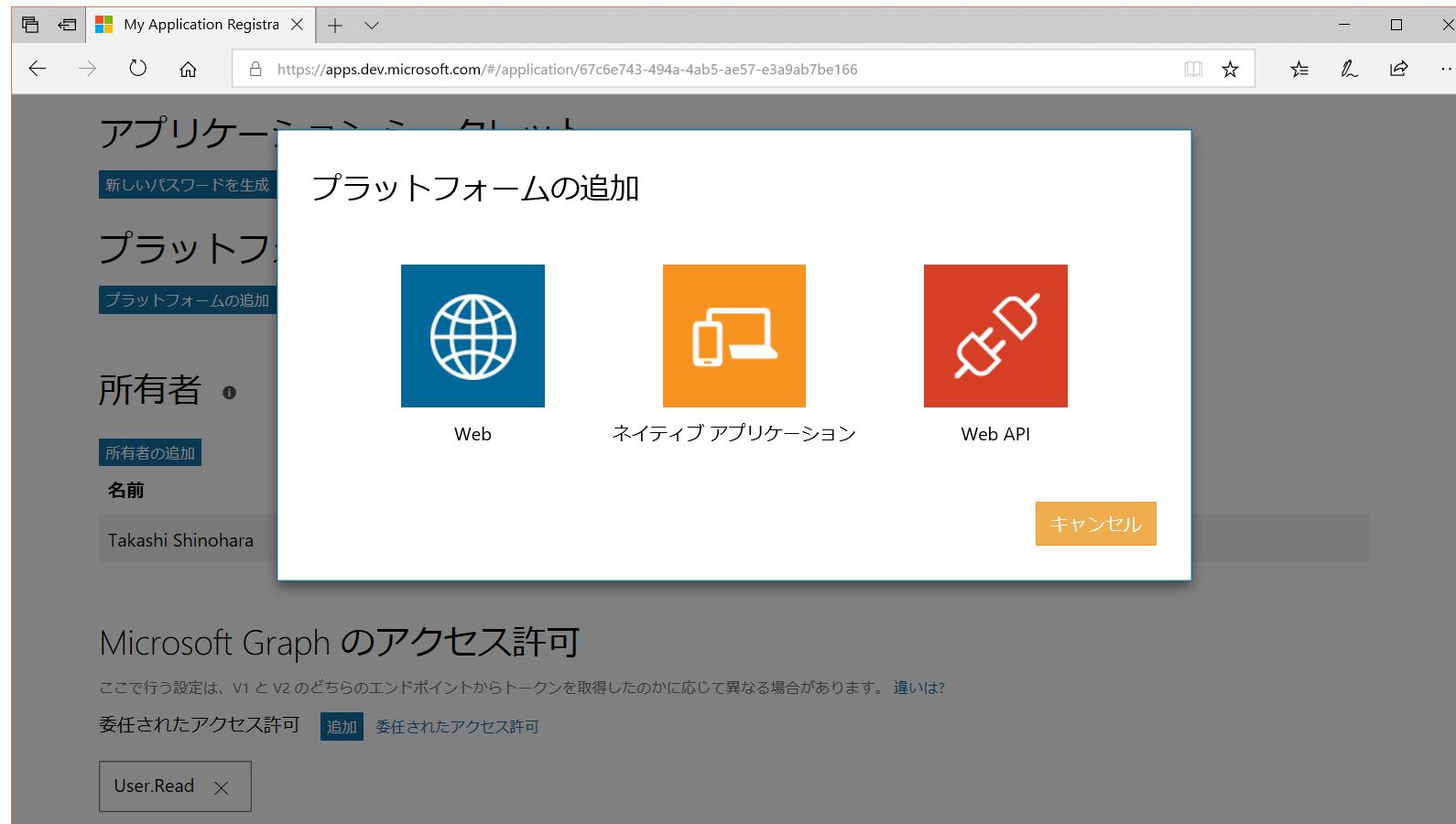
The main content area is titled "アプリケーション シークレット" (Application Secret) and includes three buttons: "新しいパスワードを生成" (Generate new password), "新しいキーの組を生成" (Generate new key pair), and "公開キーのアップロード" (Upload public key). Below this is a section titled "プラットフォーム" (Platform) with a button labeled "プラットフォームの追加" (Add platform).

Under the "所有者" (Owner) section, there is a button "所有者の追加" (Add owner). A table lists one owner: "名前" (Name) is "Takashi Shinohara" and "メール" (Email) is "takashi.shinohara@karamem0.jp".

The bottom section is titled "Microsoft Graph のアクセス許可" (Access permissions for Microsoft Graph) and contains a note: "ここで行う設定は、V1 と V2 のどちらのエンドポイントからトークンを取得したのかに応じて異なる場合があります。違いは?" (The settings here depend on whether tokens are obtained from the V1 or V2 endpoint. Differences may apply.). It shows two tabs: "委任されたアクセス許可" (Delegated permissions) and "委任されたアクセス許可" (Delegated permissions). Under "User.Read", there is a "追加" (Add) button.

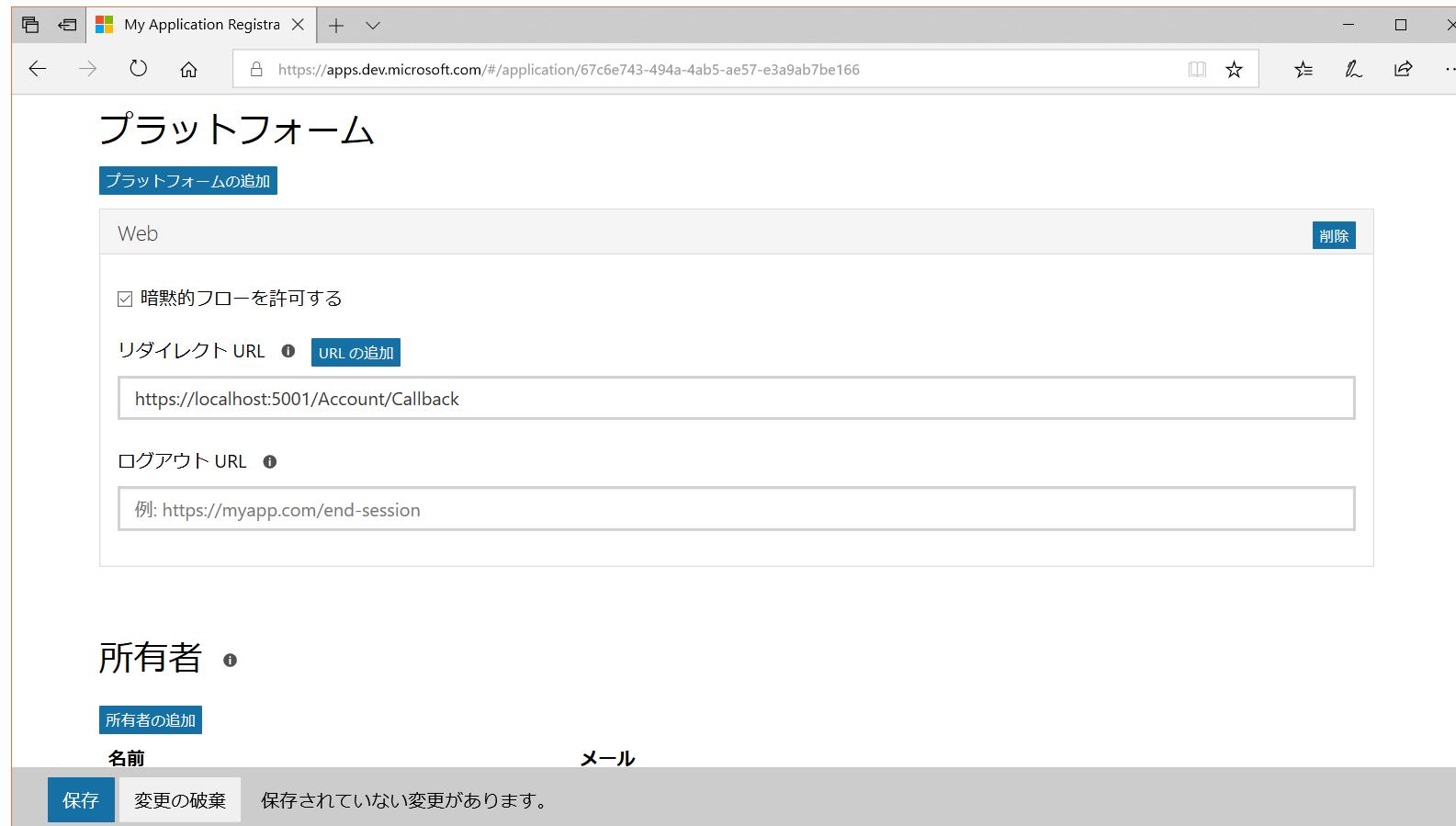
# アプリの登録: プラットフォームの選択

[Web] をクリック



# アプリの登録: リダイレクト URL の入力と保存

<https://localhost:5001/Account/Callback> と入力して保存



# Hello Microsoft Graph

アプリの登録

委任されたアクセス許可によるアプリの実行

アプリケーションのアクセス許可によるアプリの実行

SDK を使ったアプリの実行

# サンプル プロジェクト: AuthorizationCodeGrant

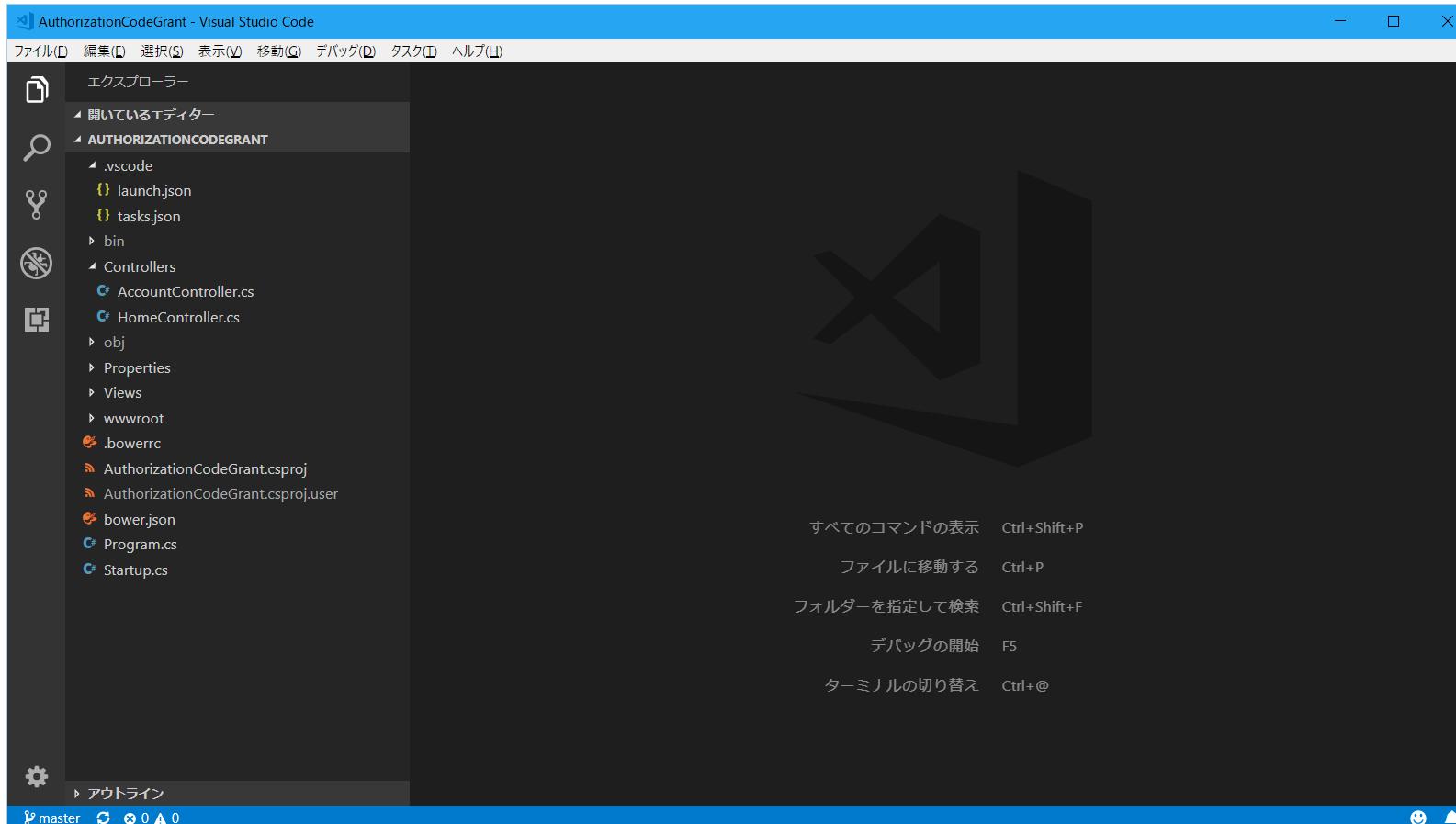
委任されたアクセス許可 (Authorization Code Grant による認可フロー) を体験するためのサンプル プロジェクト

ASP.NET Core MVC アプリケーション

サインインするとようこそメッセージを表示

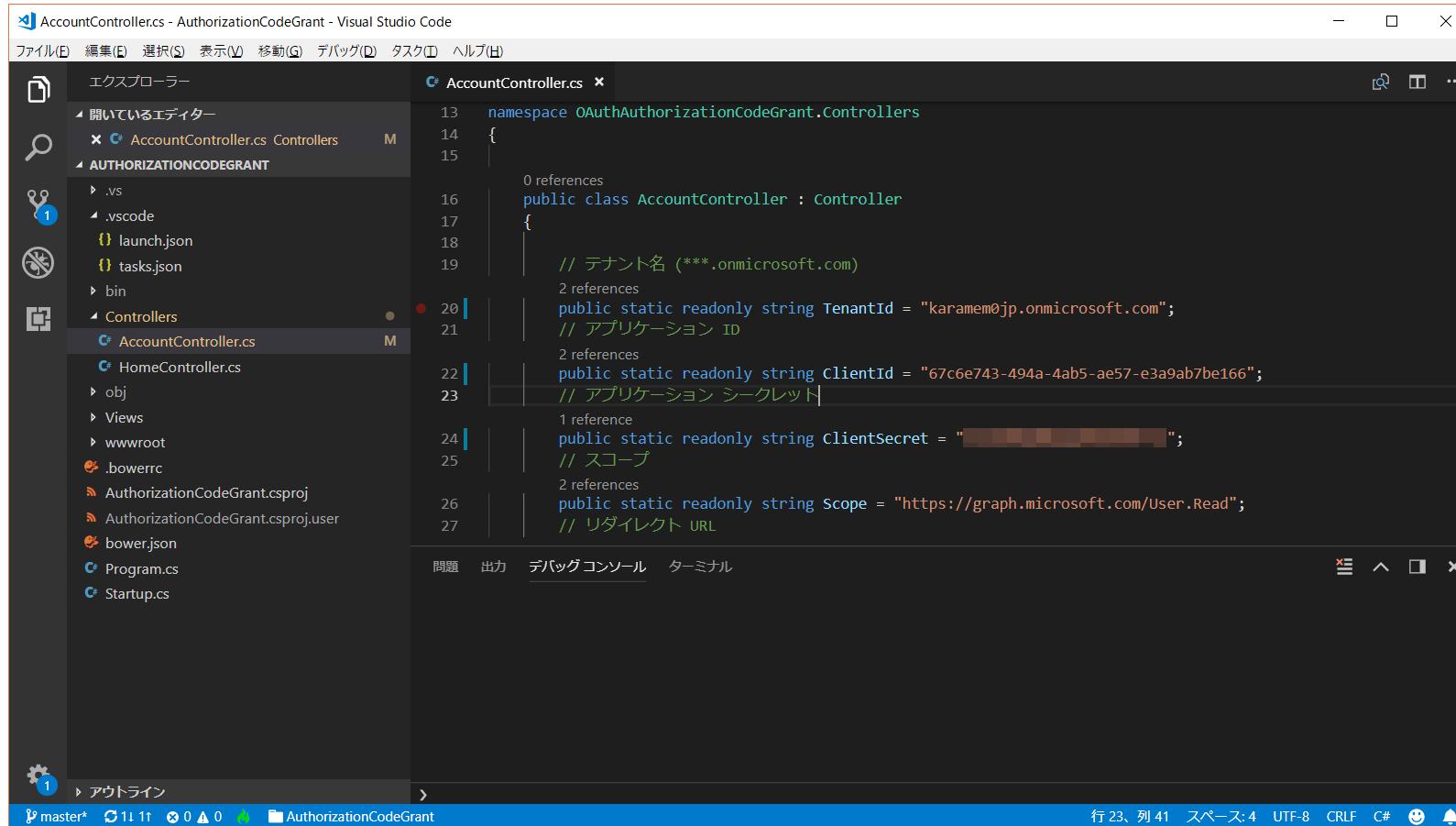
# 委任されたアクセス許可: VS Code で開く

## Visual Studio Code で [AuthorizationCodeGrant] を開く



# 委任されたアクセス許可: AccountController.cs の修正

TenantId, ClientId, ClientSecret を設定



The screenshot shows the Visual Studio Code interface with the AccountController.cs file open. The code defines a static class with properties for TenantId, ClientId, ClientSecret, and Scope.

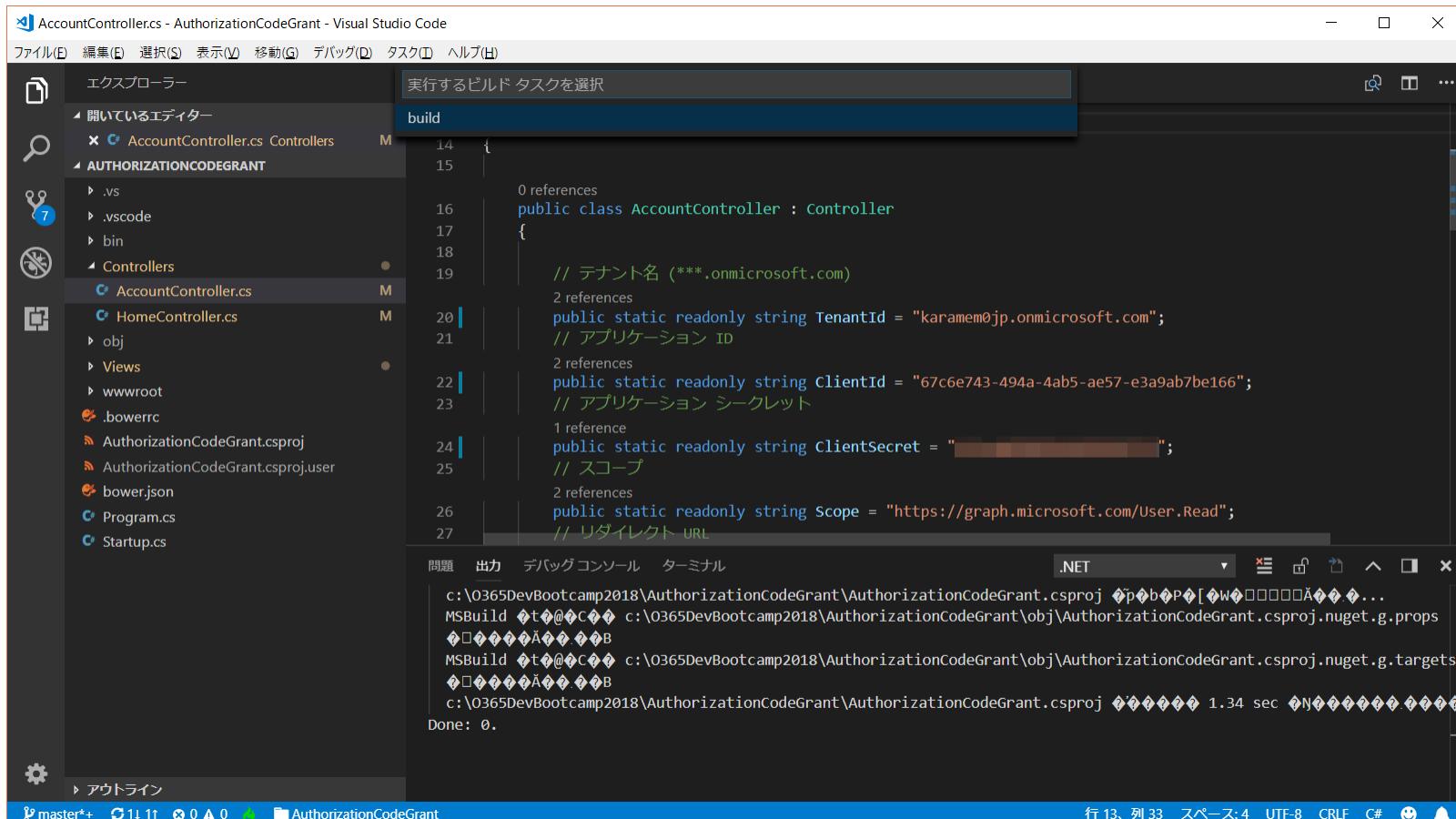
```
AccountController.cs - AuthorizationCodeGrant - Visual Studio Code
AccountController.cs - AuthorizationCodeGrant - Visual Studio Code

namespace OAuthAuthorizationCodeGrant.Controllers
{
    public class AccountController : Controller
    {
        // テナント名 (***.onmicrosoft.com)
        public static readonly string TenantId = "karamem0jp.onmicrosoft.com";
        // アプリケーション ID
        public static readonly string ClientId = "67c6e743-494a-4ab5-ae57-e3a9ab7be166";
        // アプリケーション シークレット
        public static readonly string ClientSecret = "REDACTED";
        // スコープ
        public static readonly string Scope = "https://graph.microsoft.com/User.Read";
        // リダイレクト URL
    }
}
```

The code editor shows syntax highlighting for C# and JSON files. The status bar at the bottom indicates the file is in master branch, has 1 change, and 0 errors. The bottom right corner shows the current line and column (行 23, 列 41).

# 委任されたアクセス許可: ビルドの実行

[ターミナル]-[ビルド タスクの実行]-[build] をクリック

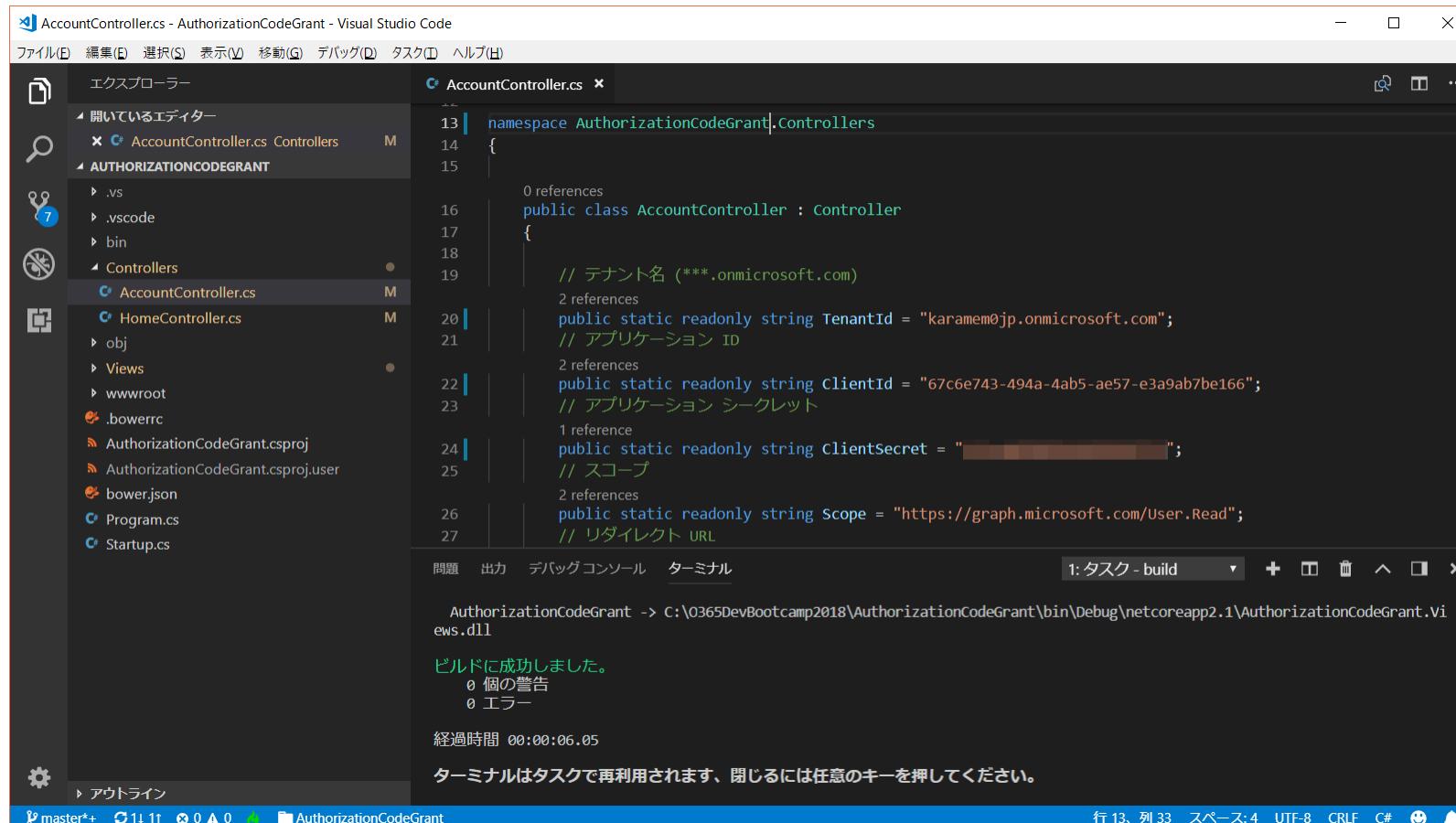


The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** AccountController.cs - AuthorizationCodeGrant - Visual Studio Code
- Menu Bar:** ファイル(E) 選択(S) 表示(V) 移動(G) デバッグ(D) タスク(T) ヘルプ(H)
- Sidebar:** エクスプローラー (File Explorer) showing the project structure:
  - 開いているエディター: AccountController.cs (Controllers)
  - AUTHORIZATIONCODEGRANT:
    - .vs
    - .vscode
    - bin
    - Controllers
      - AccountController.cs (selected)
      - HomeController.cs
    - obj
    - Views
    - wwwroot
    - .bowerrc
    - AuthorizationCodeGrant.csproj
    - AuthorizationCodeGrant.csproj.user
    - bower.json
    - Program.cs
    - Startup.cs
- Code Editor:** AccountController.cs code (highlighted line: build)
- Bottom Status Bar:** master\*+ 11 11 0 0 ▲ 0 AuthorizationCodeGrant 行 13, 列 33 スペース:4 UTF-8 CRLF C#
- Bottom Taskbar:** .NET
- Output Panel:** 显示了 MSBuild 执行日志，包括命令行参数、MSBuild 版本、目标文件路径以及完成状态 "Done: 0."

# 委任されたアクセス許可: ビルド結果の確認

[ターミナル] でビルドが成功したことを確認



The screenshot shows a Visual Studio Code window with the following details:

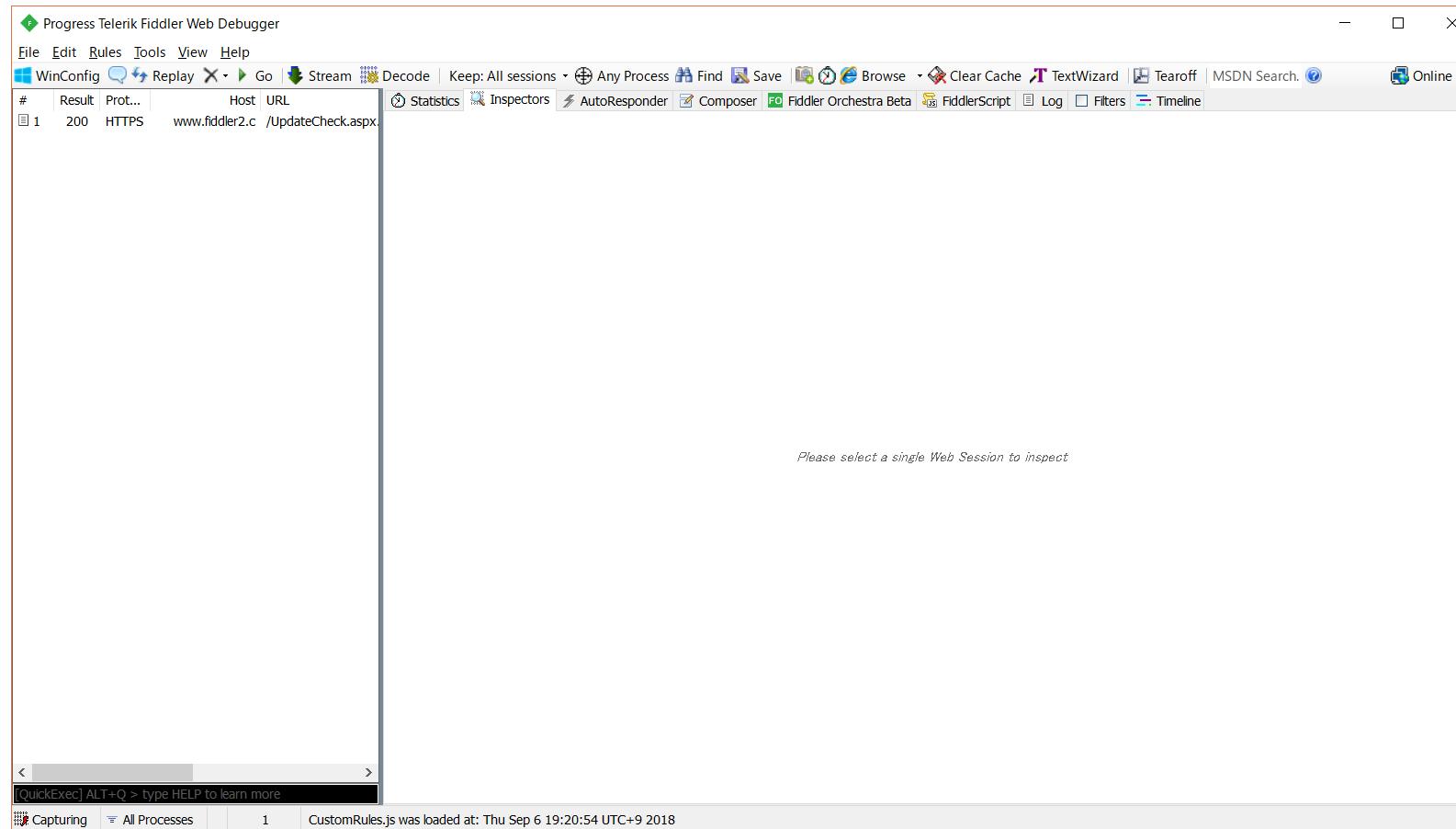
- Title Bar:** AccountController.cs - AuthorizationCodeGrant - Visual Studio Code
- File Explorer:** Shows the project structure:
  - Controllers: AccountController.cs (selected), HomeController.cs
  - Views
  - wwwroot
  - .bowerrc
  - AuthorizationCodeGrant.csproj
  - AuthorizationCodeGrant.csproj.user
  - bower.json
  - Program.cs
  - Startup.cs
- Code Editor:** AccountController.cs file content:

```
13 namespace AuthorizationCodeGrant.Controllers
14 {
15     0 references
16     public class AccountController : Controller
17     {
18         // テナント名 (***.onmicrosoft.com)
19         2 references
20         public static readonly string TenantId = "karamem0jp.onmicrosoft.com";
21         // アプリケーション ID
22         2 references
23         public static readonly string ClientId = "67c6e743-494a-4ab5-ae57-e3a9ab7be166";
24         // アプリケーション シークレット
25         1 reference
26         public static readonly string ClientSecret = "REDACTED";
27         // スコープ
28         2 references
29         public static readonly string Scope = "https://graph.microsoft.com/User.Read";
30         // リダイレクト URL
31     }
32 }
```
- Terminal:** Shows the build output:

```
AuthorizationCodeGrant -> C:\0365DevBootcamp2018\AuthorizationCodeGrant\bin\Debug\netcoreapp2.1\AuthorizationCodeGrant.dll
ビルドに成功しました。
0 個の警告
0 エラー
経過時間 00:00:06.05
ターミナルはタスクで再利用されます、閉じるには任意のキーを押してください。
```
- Status Bar:** master\*+ 11 11 0 0 ▲ 0 AuthorizationCodeGrant 行 13, 列 33 スペース:4 UTF-8 CRLF C# 🎙

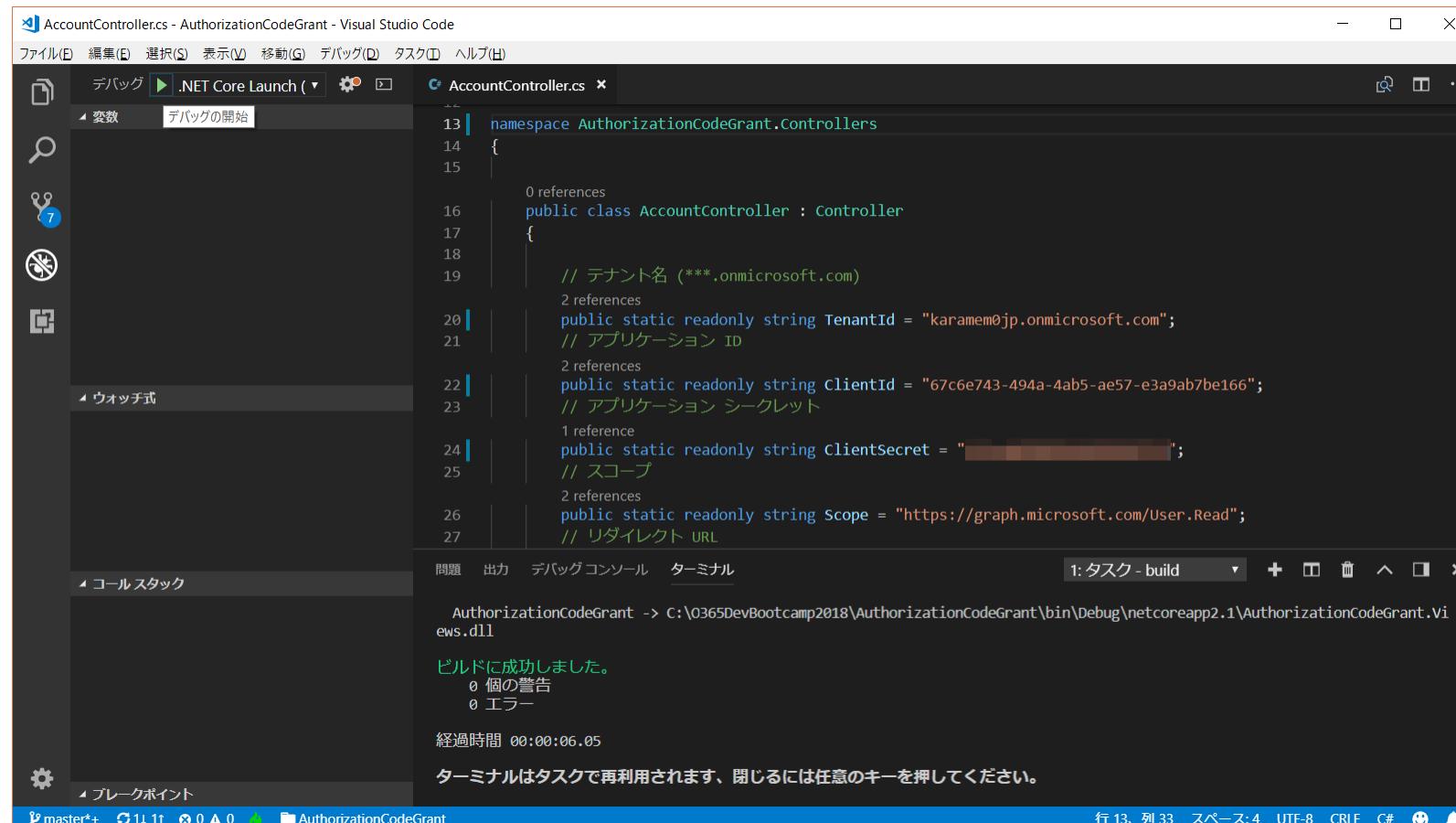
# 委任されたアクセス許可: Fiddler

## Fiddler を起動



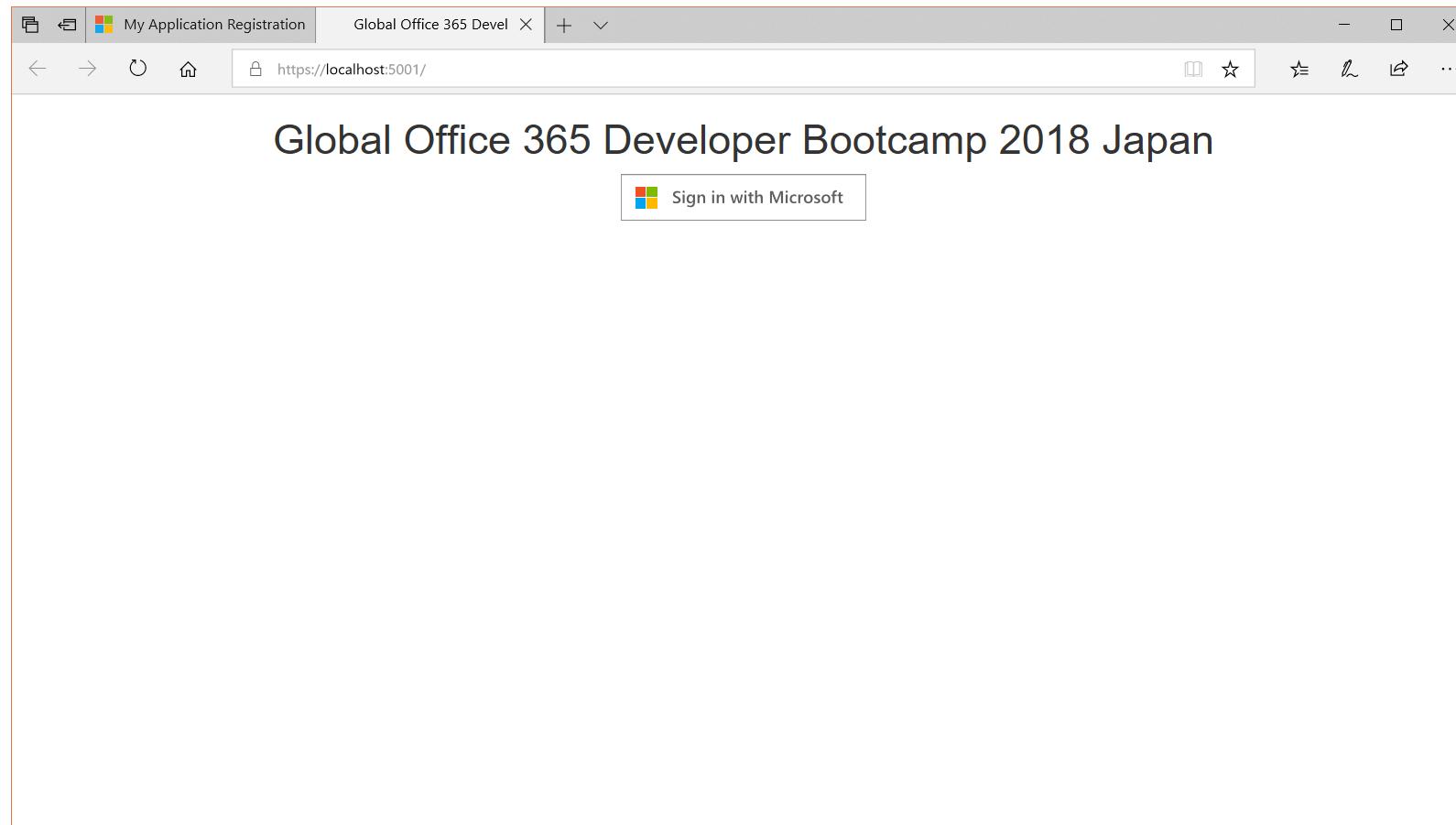
# 委任されたアクセス許可: デバッグの開始

[デバッグ]-[デバッグの開始] をクリック



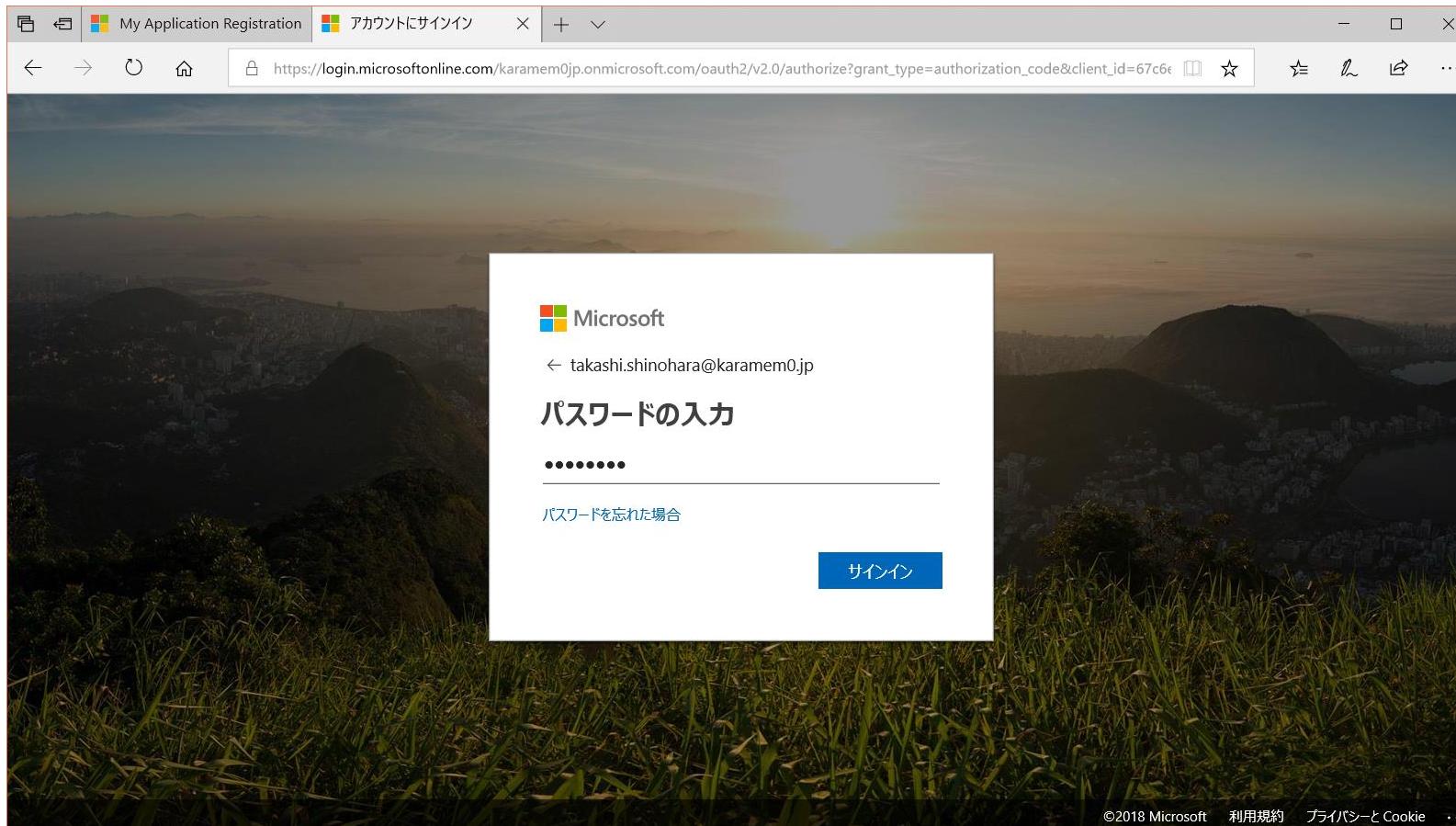
# 委任されたアクセス許可: Sign in with Microsoft

[Sign in with Microsoft] をクリック



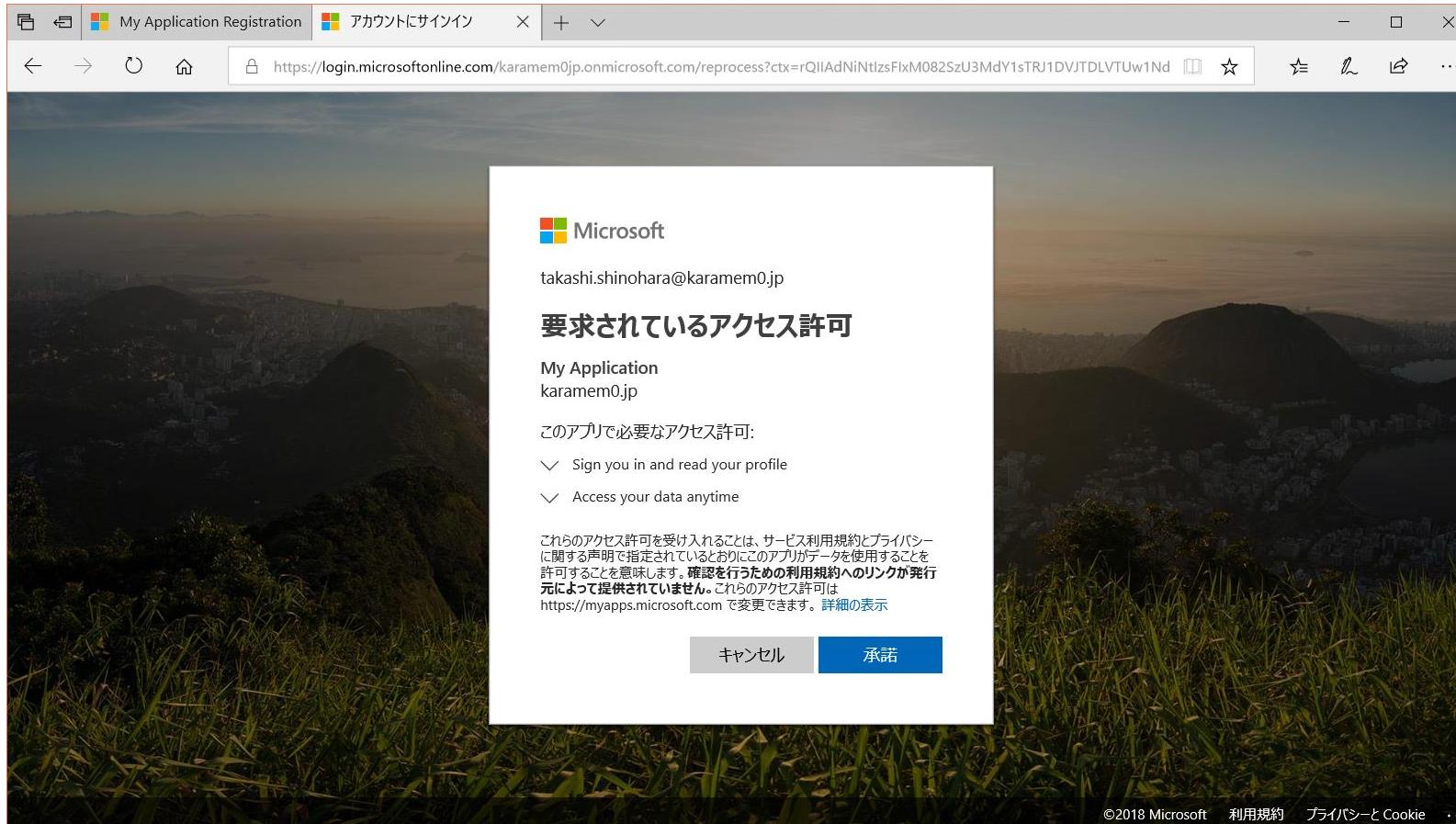
# 委任されたアクセス許可: サインイン

## 組織アカウントでサインイン



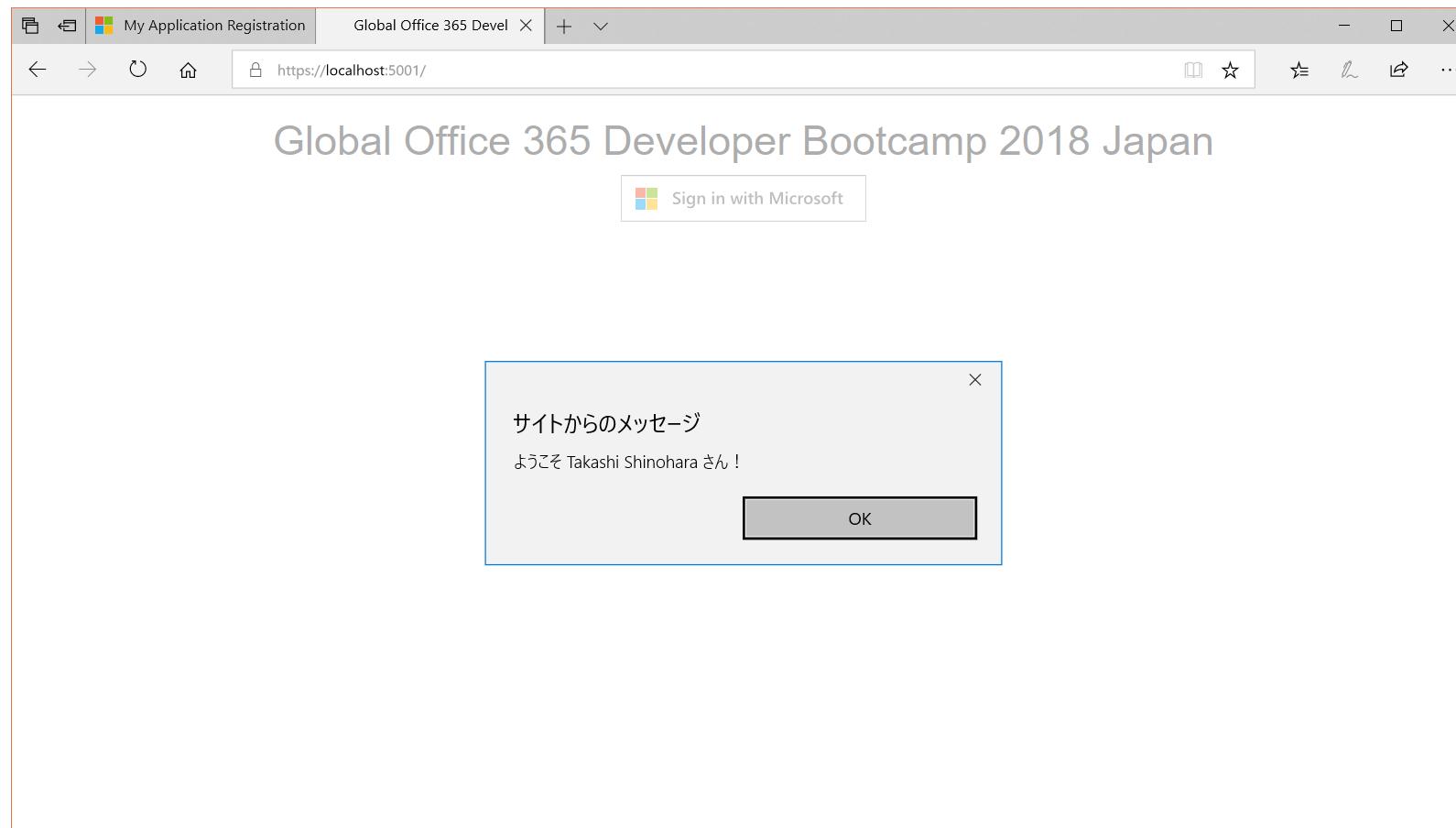
# 委任されたアクセス許可: アクセス許可の承認

[承諾] をクリック

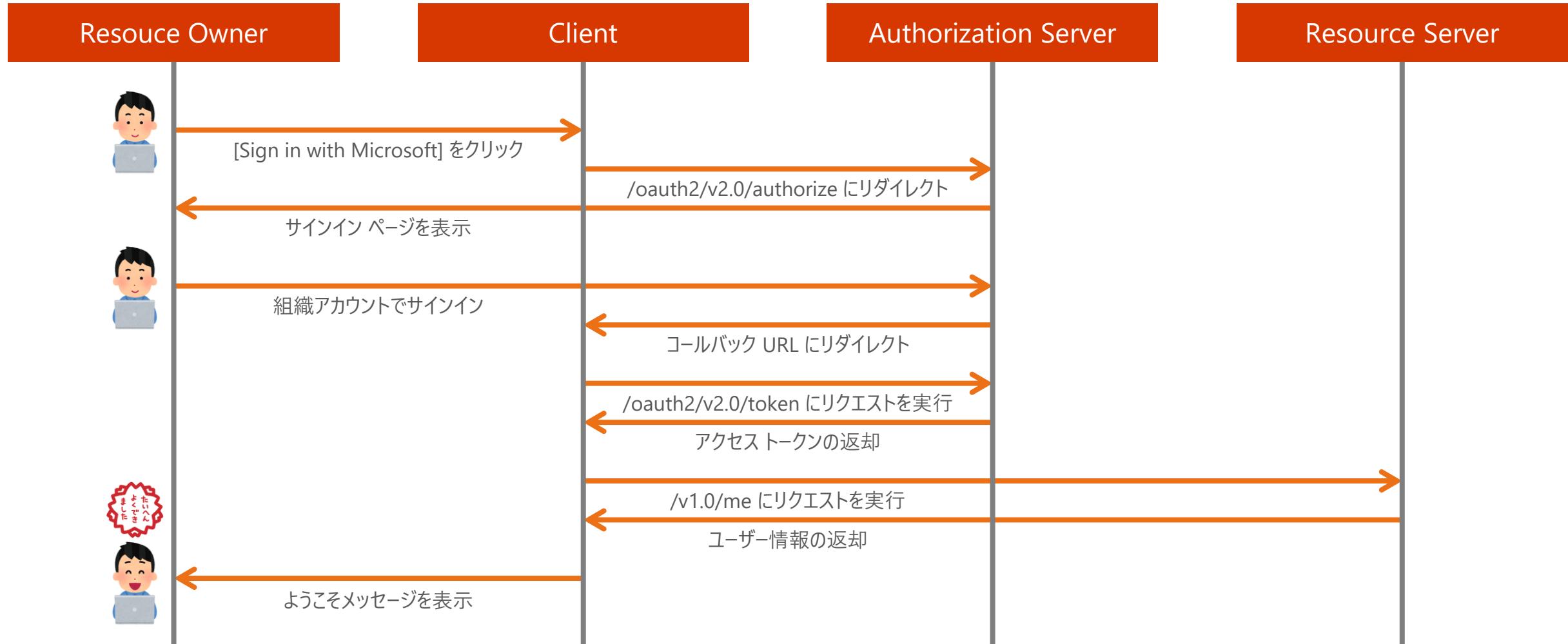


# 委任されたアクセス許可: ようこそ

ようこそメッセージが表示されることを確認



# 委任されたアクセス許可: 認可フロー



# 委任されたアクセス許可: /oauth2/v2.0/authorize

Fiddler で実行結果を確認

The screenshot shows the Fiddler Web Debugger interface during an OAuth2 authorization code grant process. The main pane displays a list of network requests and responses. Request 27 shows the authorization URL with parameters:

Name	Value
grant_type	authorization_code
client_id	67c6e743-494a-4ab5-ae57-e3a9ab7be166
scope	https://graph.microsoft.com/User.Read
response_type	code
redirect_url	https://localhost:5001/Account/Callback

The response body for Request 40 indicates that the response is encoded and can be decoded.

At the bottom, the status bar shows the URL: [https://login.microsoftonline.com/karamem0jp.onmicrosoft.com/oauth2/v2.0/authorize?grant\\_type=authorization\\_code&client\\_id=67c6e743-494a-4ab5-ae57-e3a9ab7be166](https://login.microsoftonline.com/karamem0jp.onmicrosoft.com/oauth2/v2.0/authorize?grant_type=authorization_code&client_id=67c6e743-494a-4ab5-ae57-e3a9ab7be166).

# 委任されたアクセス許可: /oauth2/v2.0/token

Fiddler で実行結果を確認

The screenshot shows the Fiddler Web Debugger interface with the following details:

**Session List:** Shows multiple requests and responses, with session 42 highlighted. Session 42 details are as follows:

Name	Value
grant_type	authorization_code
code	OAQABAAIAAADXzZ3fr-GRbDT45zNSEFMn8Hw_w-Qv55BJWxTy87lgRhVm1UI5KE2S-6Is-Sx067c6e743-494a-4ab5-ae57-e3a9ab7be166
client_id	67c6e743-494a-4ab5-ae57-e3a9ab7be166
client_secret	[redacted]
scope	https://graph.microsoft.com/User.Read

**Transformer:** The JSON response is displayed in the Transformer pane:

```
access_token=eyJ0eXAiOiJKV1QiLCJub25jZSI6IkFRQUJBQUFBQUFEWHPaM2lmc1HUmJEVDQ1ek5TRUZFSFdXNTktdnJlZTROrmtMXzNFUWlxevFBaXrRvEU2WXIWmRuZU5xV19RbHzt
expires_in=3599
ext_expires_in=0
scope=profile openid email https://graph.microsoft.com/User.Read
token_type=Bearer
```

**Status Bar:** Shows the URL <https://login.microsoftonline.com/karamem0jp.onmicrosoft.com/oauth2/v2.0/token>.

# 委任されたアクセス許可: /v1.0/me

## Fiddler で実行結果を確認

The screenshot shows the Fiddler Web Debugger interface with the following details:

- Request Headers:** GET /v1.0/me HTTP/1.1  
Client: Accept: application/json  
Security: Authorization: Bearer eyJ0eXAiOiJKV1QiLCJub25jZSI6IkFRQUJBQUFBQUEWHPaM2Imc1HUmJEVDQ1ek5TRUZFSFdNTktdnJZTRORmtMXzNFUWixeVFbAXRrVEU2WxWWmRuZU5xV19RbI  
Transport: Host: graph.microsoft.com
- Response Body (JSON Format):**

```
@odata.context=https://graph.microsoft.com/v1.0/$metadata#users/$entity
businessPhones
displayName=Takashi Shinohara
givenName=Takashi
id=4b34f1e2-0c77-4fb1-a8cf-94a4606021ee
jobTitle=(null)
mail=takashi.shinohara@karamem0.jp
mobilePhone=(null)
officeLocation=(null)
preferredLanguage=ja-JP
surname=Shinohara
userPrincipalName=takashi.shinohara@karamem0.jp
```

# Hello Microsoft Graph

アプリの登録

委任されたアクセス許可によるアプリの実行

アプリケーションのアクセス許可によるアプリの実行

SDK を使ったアプリの実行

# サンプル プロジェクト: ClientCredentialsGrant

アプリケーションのアクセス許可 (Client Credentials Grant による認可フロー) を体験するためのサンプル プロジェクト

.NET Core コンソール アプリケーション

実行すると組織内のすべてのユーザーの情報を表示

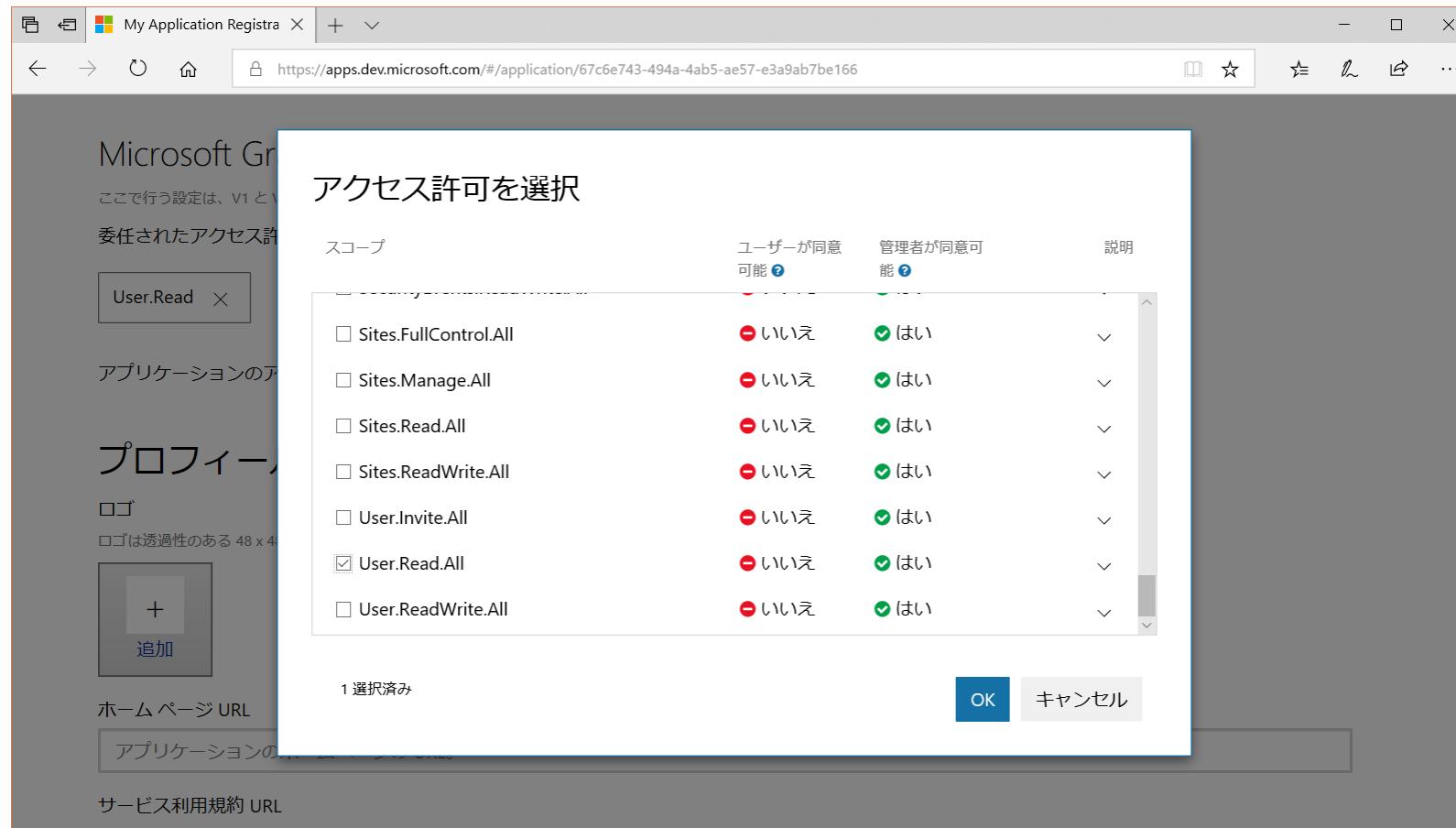
# アプリケーションのアクセス許可: アクセス許可の追加

[アプリケーションのアクセス許可]-[追加] をクリック



# アプリケーションのアクセス許可: アクセス許可の選択

[User.Read.All] を選択して [OK] をクリック



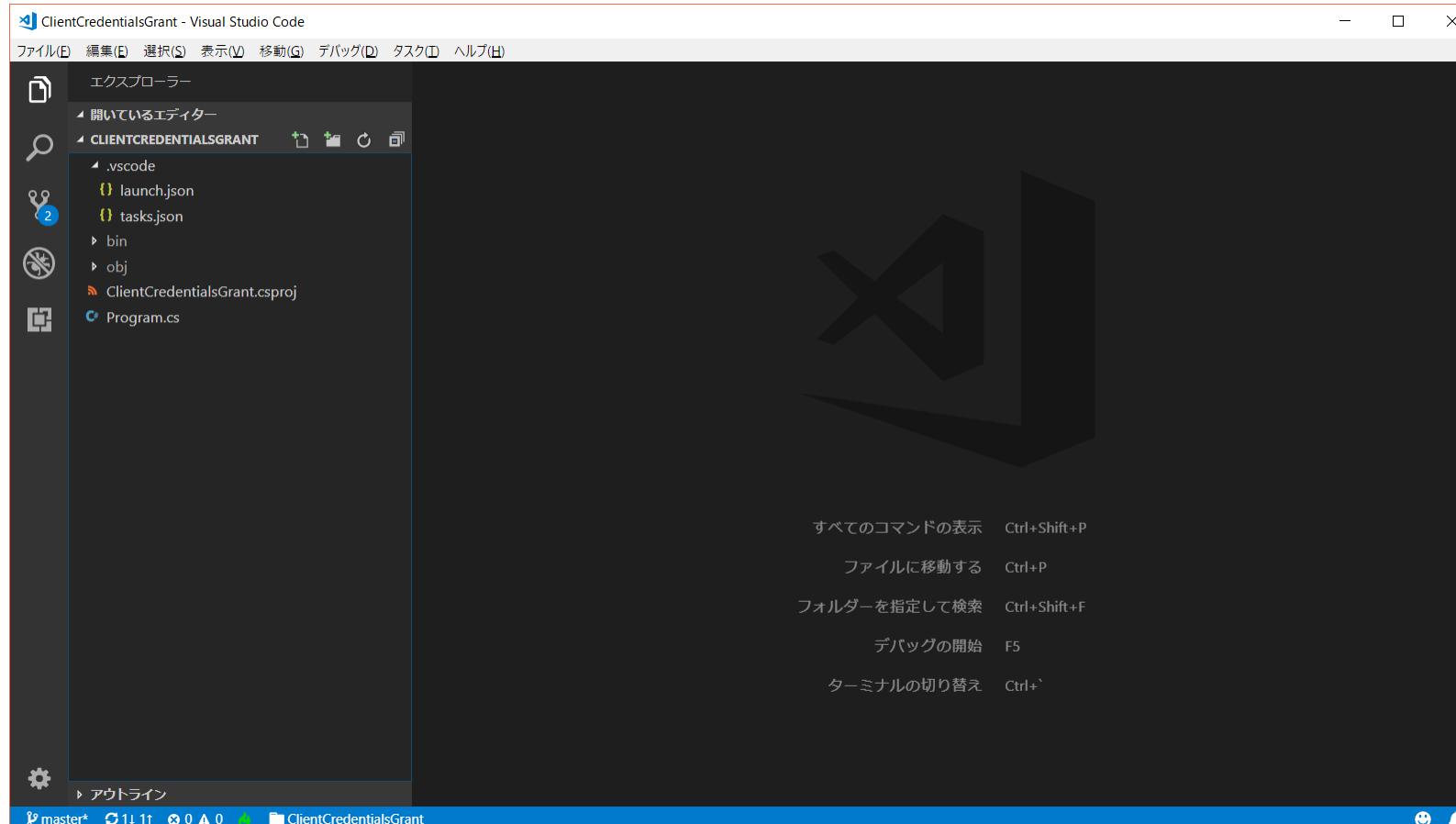
# アプリケーションのアクセス許可: 変更の保存

[保存] をクリック



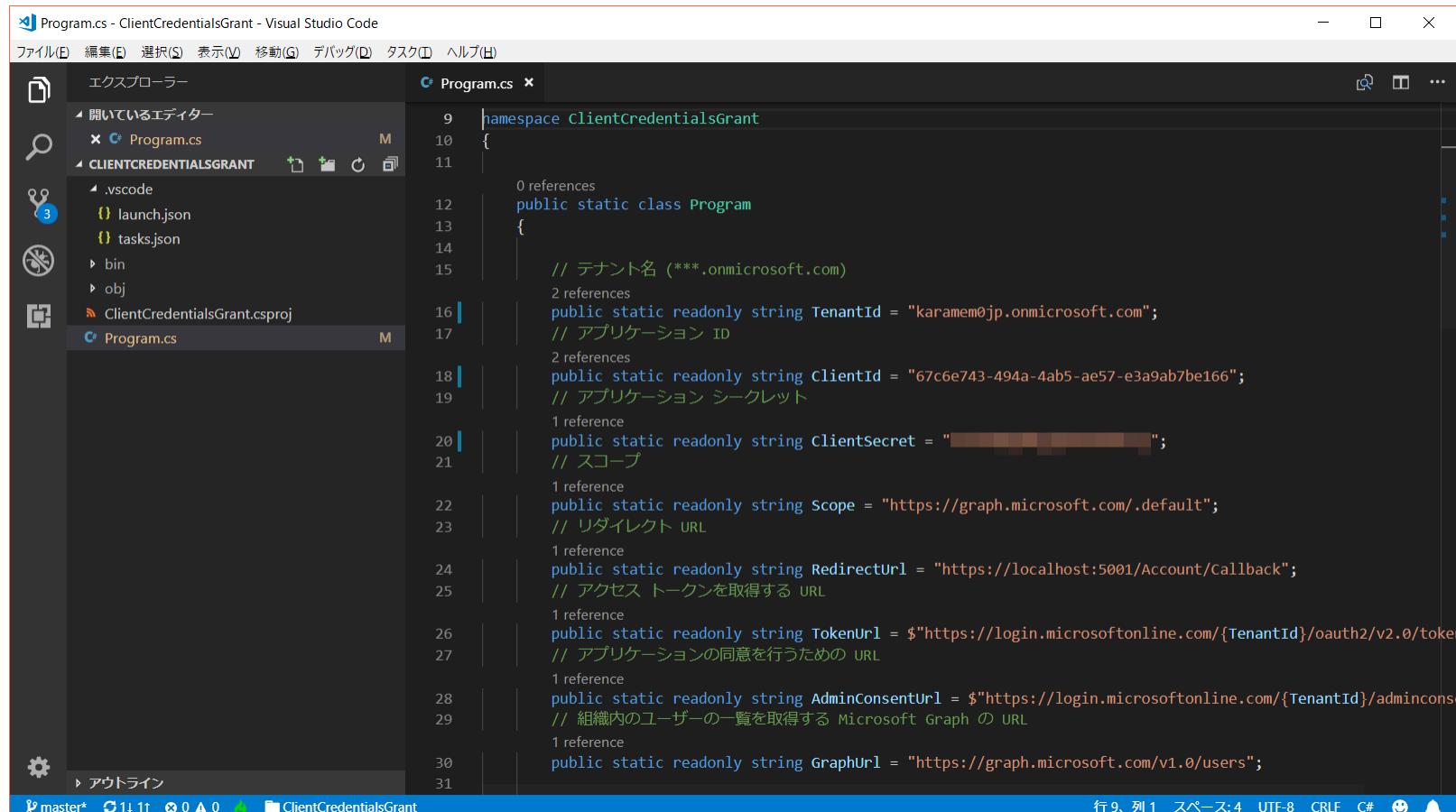
# アプリケーションのアクセス許可: VS Code で開く

Visual Studio Code で [ClientCredentialsGrant] を開く



# アプリケーションのアクセス許可: Program.cs の修正

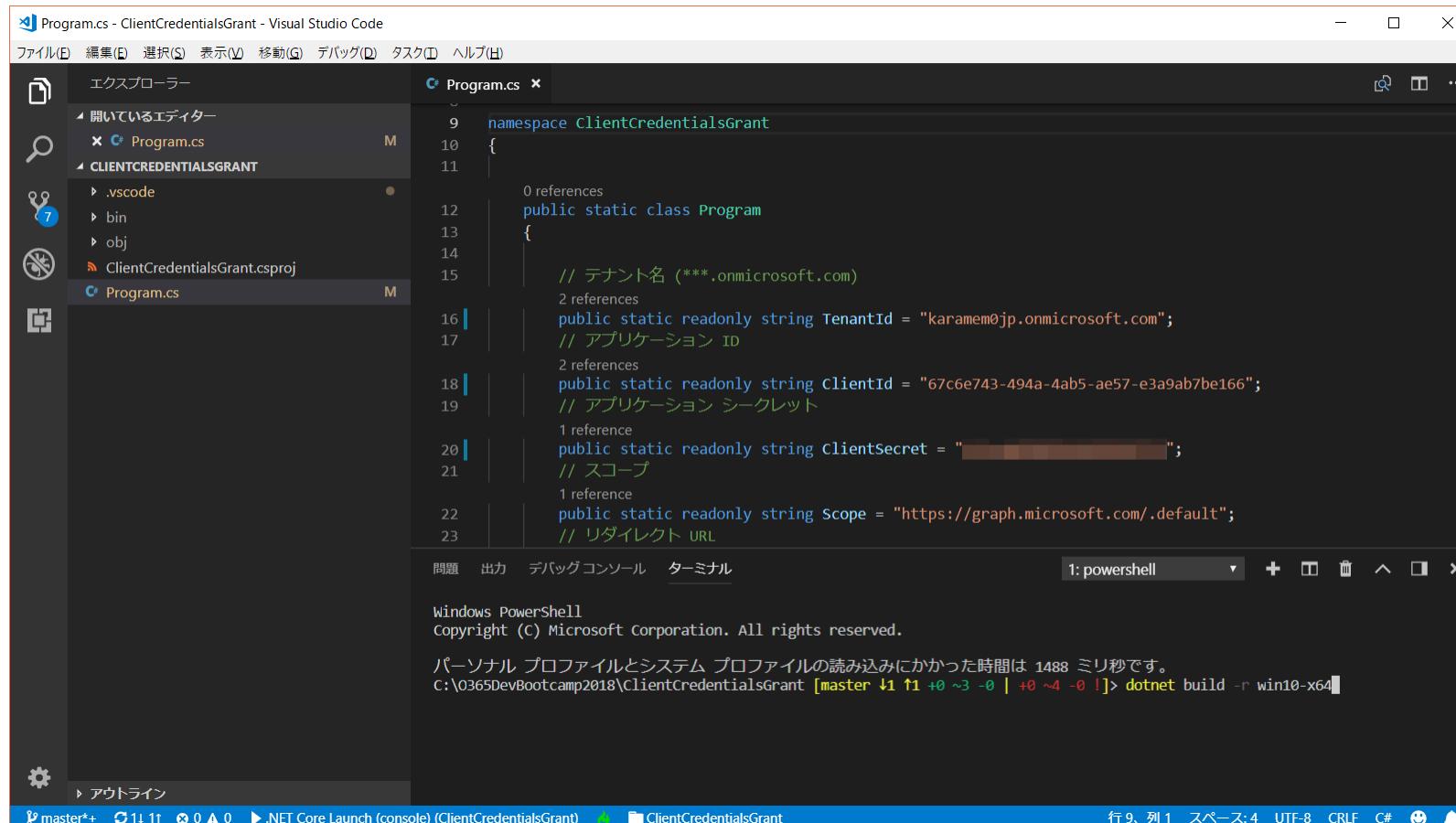
TenantId, ClientId, ClientSecret を設定



```
Program.cs - ClientCredentialsGrant - Visual Studio Code
Program.cs
namespace ClientCredentialsGrant
{
    public static class Program
    {
        // テナント名 (***.onmicrosoft.com)
        public static readonly string TenantId = "karamem0jp.onmicrosoft.com";
        // アプリケーション ID
        public static readonly string ClientId = "67c6e743-494a-4ab5-ae57-e3a9ab7be166";
        // アプリケーション シークレット
        public static readonly string ClientSecret = "XXXXXXXXXXXXXX";
        // スコープ
        public static readonly string Scope = "https://graph.microsoft.com/.default";
        // リダイレクト URL
        public static readonly string RedirectUrl = "https://localhost:5001/Account/Callback";
        // アクセス トークンを取得する URL
        public static readonly string TokenUrl = $"https://login.microsoftonline.com/{TenantId}/oauth2/v2.0/token";
        // アプリケーションの同意を行うための URL
        public static readonly string AdminConsentUrl = $"https://login.microsoftonline.com/{TenantId}/adminconsent";
        // 組織内のユーザーの一覧を取得する Microsoft Graph の URL
        public static readonly string GraphUrl = "https://graph.microsoft.com/v1.0/users";
    }
}
```

# アプリケーションのアクセス許可: ビルドの実行

[ターミナル] で *dotnet build -r <RID>* と入力して実行



The screenshot shows the Visual Studio Code interface. The left sidebar displays a file tree with 'Program.cs' selected. The main editor window shows the code for 'Program.cs':

```
namespace ClientCredentialsGrant
{
    public static class Program
    {
        // テナント名 (***.onmicrosoft.com)
        public static readonly string TenantId = "karamem0jp.onmicrosoft.com";
        // アプリケーション ID
        public static readonly string ClientId = "67c6e743-494a-4ab5-ae57-e3a9ab7be166";
        // アプリケーション シークレット
        public static readonly string ClientSecret = "REDACTED";
        // スコープ
        public static readonly string Scope = "https://graph.microsoft.com/.default";
        // リダイレクト URL
    }
}
```

Below the editor is a terminal window titled 'powershell'. It shows the command being run:

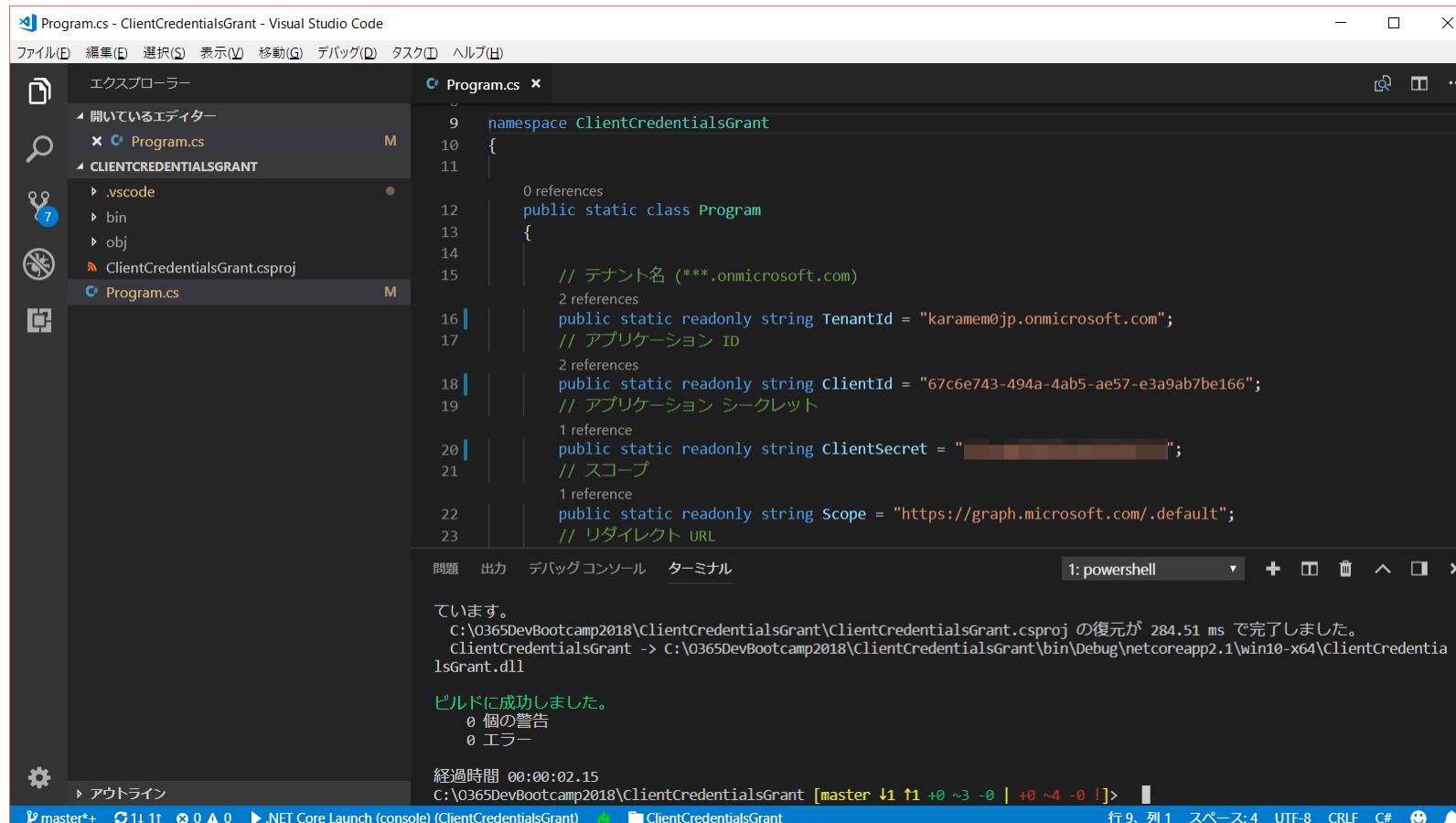
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

パーソナル プロファイルとシステム プロファイルの読み込みにかかった時間は 1488 ミリ秒です。
C:\0365DevBootcamp2018\ClientCredentialsGrant [master ✘ 11 +1 ~3 -0 | +0 ~4 -0 !] dotnet build -r win10-x64
```

The status bar at the bottom indicates the file is 'ClientCredentialsGrant' and the current line is '行 9、列 1'.

# アプリケーションのアクセス許可: ビルド結果の確認

[ターミナル] でビルドが成功したことを確認

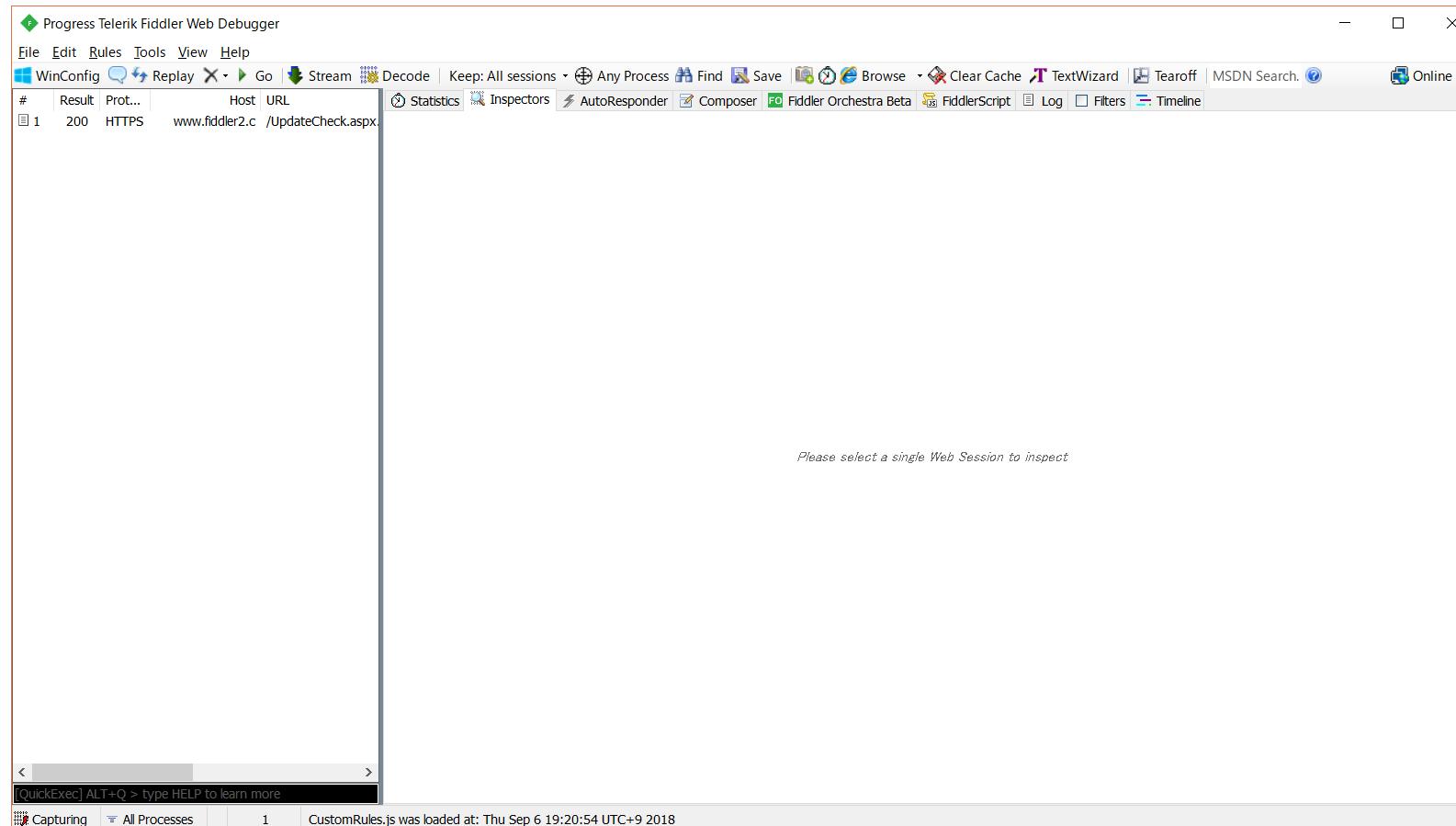


The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Program.cs - ClientCredentialsGrant - Visual Studio Code
- File Explorer:** Shows the project structure: .vscode, bin, obj, ClientCredentialsGrant.csproj, and Program.cs.
- Code Editor:** Displays the content of Program.cs, which defines a static class Program with tenant ID, client ID, client secret, and scope variables.
- Terminal:** Shows the build output:
  - Build completed successfully: "Build succeeded."
  - Build time: "Elapsed time: 00:00:02.15"
  - Build path: "C:\0365DevBootcamp2018\ClientCredentialsGrant\bin\Debug\netcoreapp2.1\win10-x64\ClientCredentialsGrant.dll"
  - Success message: "Build successful."
  - 0 warnings and 0 errors.
- Status Bar:** Shows the current branch is master, and the file is ClientCredentialsGrant.cs.

# アプリケーションのアクセス許可: Fiddler

## Fiddler を起動



# アプリケーションのアクセス許可: プログラムの実行

[ターミナル] で *dotnet run -r <RID>* と入力して実行

The screenshot shows the Visual Studio Code interface. The left sidebar displays a file tree with files like .vscode, bin, obj, and ClientCredentialsGrant.csproj. The main editor window shows the code for Program.cs:

```
namespace ClientCredentialsGrant
{
    public static class Program
    {
        // テナント名 (***.onmicrosoft.com)
        public static readonly string TenantId = "karamem0jp.onmicrosoft.com";
        // アプリケーション ID
        public static readonly string ClientId = "67c6e743-494a-4ab5-ae57-e3a9ab7be166";
        // アプリケーション シークレット
        public static readonly string ClientSecret = "REDACTED";
        // スコープ
        public static readonly string Scope = "https://graph.microsoft.com/.default";
        // リダイレクト URL
    }
}
```

Below the editor, a terminal window is open with the command `dotnet run -r win10-x64`. The output shows the build process completed successfully:

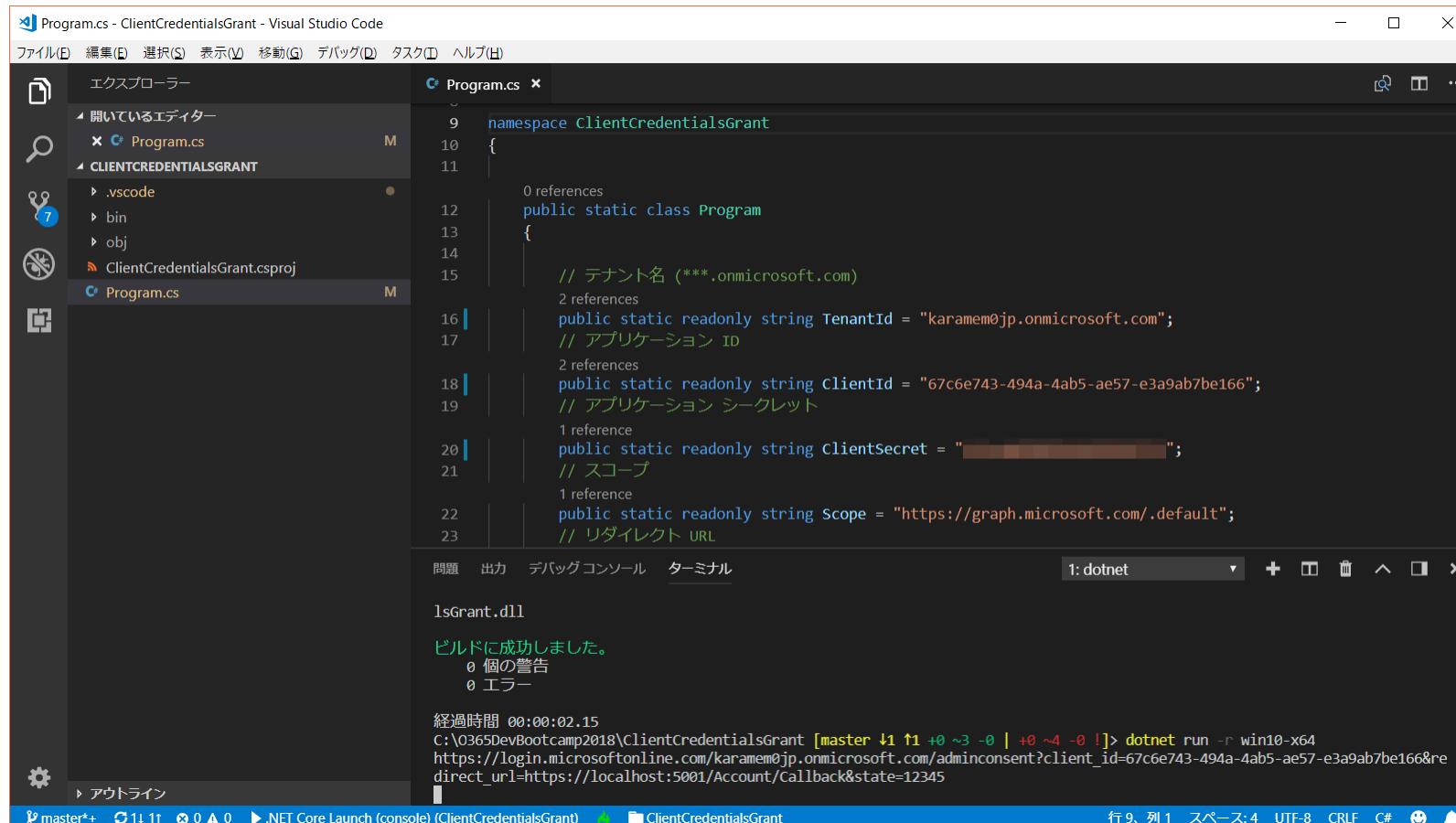
```
1: powershell
C:\0365DevBootcamp2018\ClientCredentialsGrant\ClientCredentialsGrant.csproj の復元が 284.51 ms で完了しました。
ClientCredentialsGrant -> C:\0365DevBootcamp2018\ClientCredentialsGrant\bin\Debug\netcoreapp2.1\win10-x64\ClientCredentialsGrant.dll

ビルドに成功しました。
0 個の警告
0 エラー

経過時間 00:00:02.15
C:\0365DevBootcamp2018\ClientCredentialsGrant [master ↴ 1 ↑ 1 +0 ~3 -0 | +0 ~4 -0 !]> dotnet run -r win10-x64
```

# アプリケーションのアクセス許可: Admin Consent

[ターミナル] に表示される URL をコピーしてアクセス



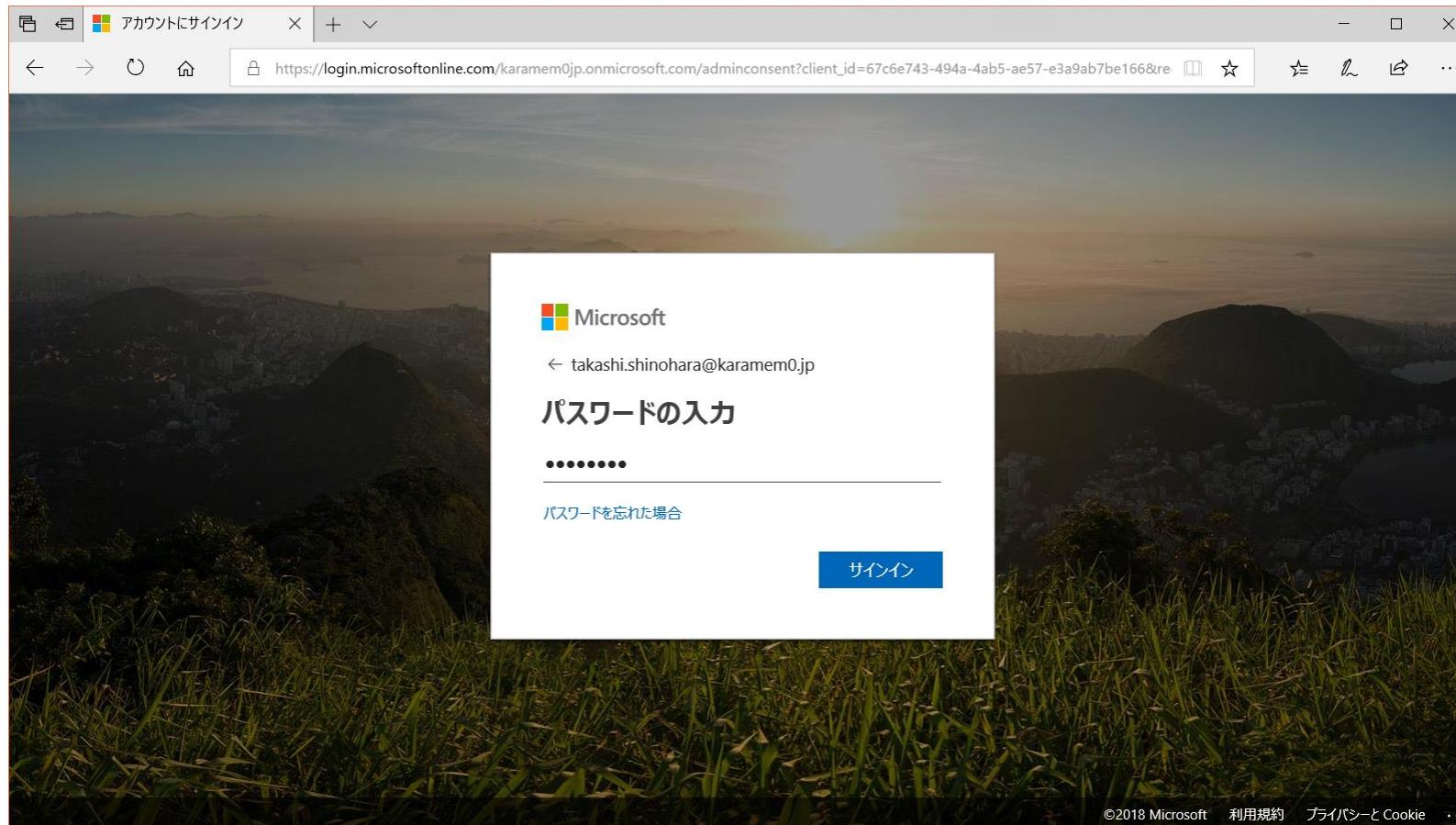
```
Program.cs - ClientCredentialsGrant - Visual Studio Code
ファイル(E) 選択(S) 表示(V) 移動(G) デバッグ(D) タスク(I) ヘルプ(H)
エクスプローラー
  開いているエディター
    Program.cs
  CLIENTCREDENTIALSGRANT
    .vscode
    bin
    obj
    ClientCredentialsGrant.csproj
    Program.cs
Program.cs
namespace ClientCredentialsGrant
{
    public static class Program
    {
        // テナント名 (***.onmicrosoft.com)
        public static readonly string TenantId = "karamem0jp.onmicrosoft.com";
        // アプリケーション ID
        public static readonly string ClientId = "67c6e743-494a-4ab5-ae57-e3a9ab7be166";
        // アプリケーション シークレット
        public static readonly string ClientSecret = "XXXXXXXXXXXXXX";
        // スコープ
        public static readonly string Scope = "https://graph.microsoft.com/.default";
        // リダイレクト URL
    }
}

lsGrant.dll
ビルトに成功しました。
0 個の警告
0 エラー

経過時間 00:00:02.15
C:\0365DevBootcamp2018\ClientCredentialsGrant [master ✚ 1 1 +0 ~3 -0 | +0 ~4 -0 !] > dotnet run -r win10-x64
https://login.microsoftonline.com/karamem0jp.onmicrosoft.com/adminconsent?client_id=67c6e743-494a-4ab5-ae57-e3a9ab7be166&re
direct_url=https://localhost:5001/Account/Callback&state=12345
```

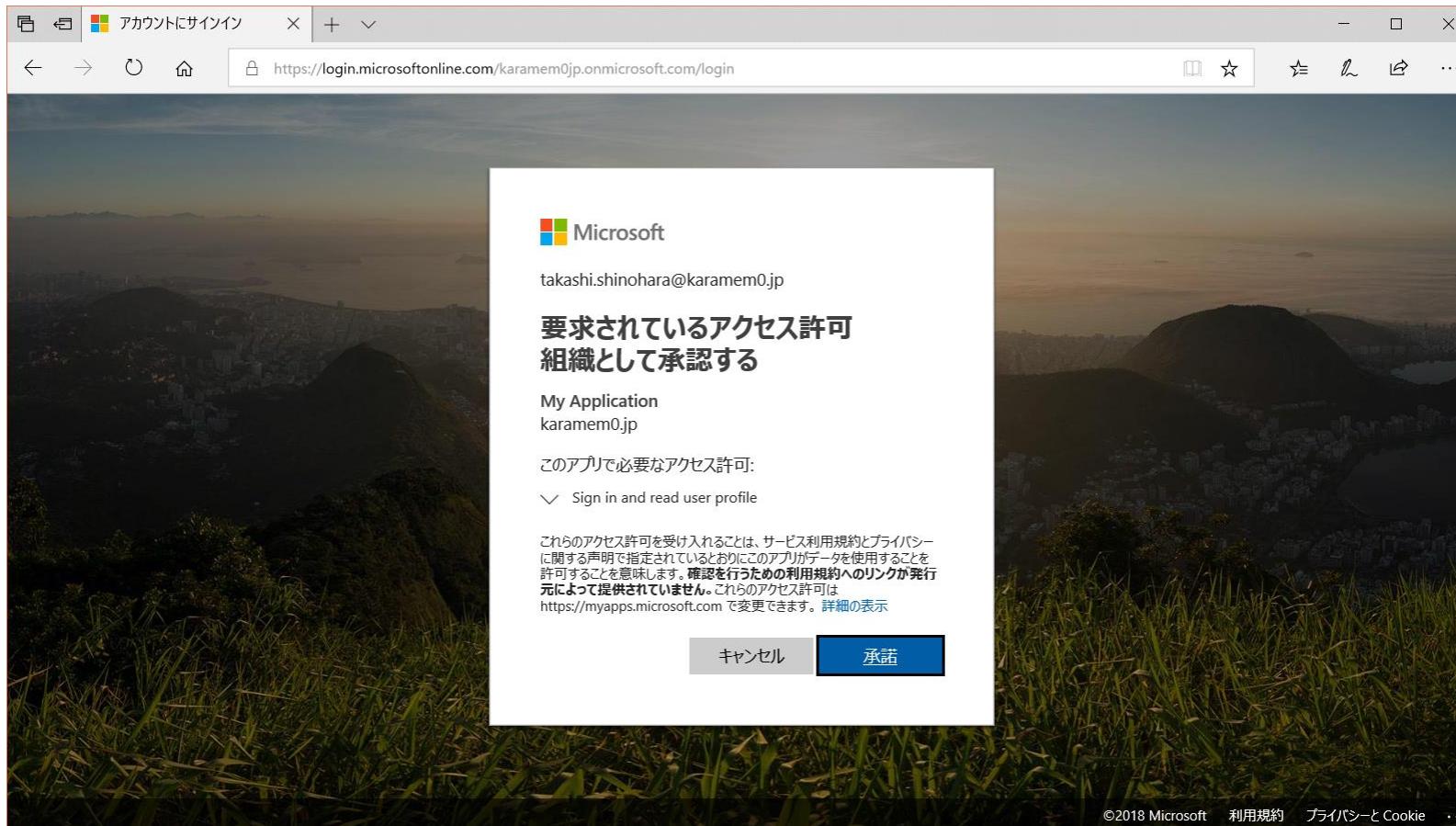
# アプリケーションのアクセス許可: サインイン

## 組織アカウントでサインイン



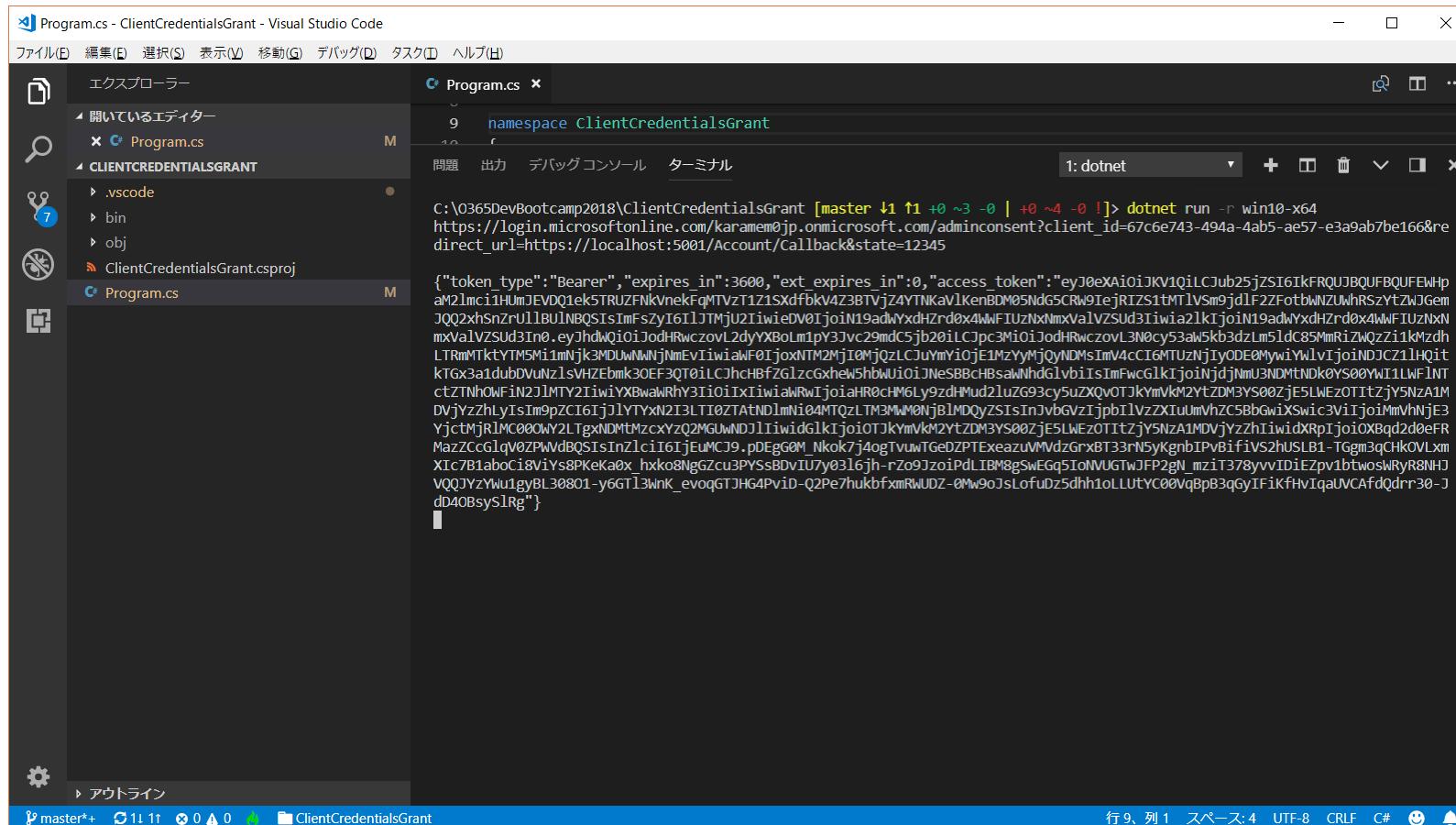
# アプリケーションのアクセス許可: アクセス許可の承認

[承諾] をクリック



# アプリケーションのアクセス許可: アクセストークンの表示

[ターミナル] で Enter キーを押すとアクセストークンを表示



The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** Program.cs - ClientCredentialsGrant - Visual Studio Code
- File Menu:** ファイル(E) 編集(F) 選択(S) 表示(V) 移動(G) デバッグ(D) タスク(I) ヘルプ(H)
- Left Sidebar:** エクスプローラー (File Explorer) showing the project structure:
  - 開いているエディター: Program.cs
  - CLIENTCREDENTIALSGRANT folder
    - .vscode
    - bin
    - obj
    - ClientCredentialsGrant.csproj
  - Program.cs
- Center Area:** Program.cs editor tab, showing the code:

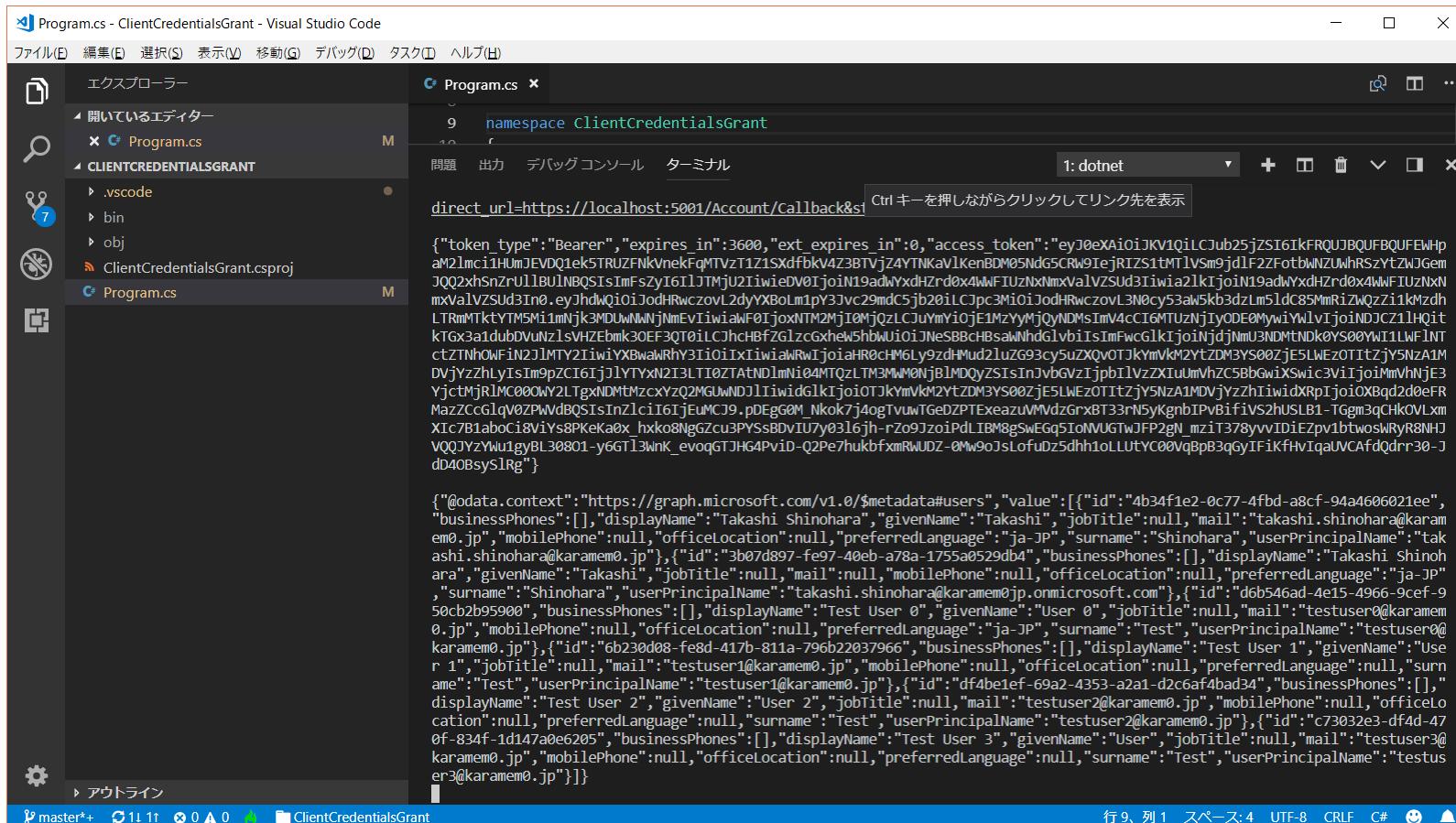
```
9 namespace ClientCredentialsGrant
```
- Terminal Tab:** ターミナル (Terminal) tab, showing the command and its output:

```
C:\0365DevBootcamp2018\ClientCredentialsGrant [master 11 11 +0 ~3 -0 | +0 ~4 -0 !] > dotnet run -r win10-x64
https://login.microsoftonline.com/karamem@jp.onmicrosoft.com/adminconsent?client_id=67c6e743-494a-4ab5-ae57-e3a9ab7be166&re
direct_url=https://localhost:5001/Account/Callback&state=12345
```

```
{"token_type": "Bearer", "expires_in": 3600, "ext_expires_in": 0, "access_token": "eyJ0eXAiOiJKV1QiLCJub25jZSI6IkFRQUJBQUFBQUEWHP
ah21mciiHUmJEVDQ1ek5TRUZFNVnekFqMTVzt1Z1sXdfbV4Z3BTvJz4YTNKaVLkenBDM05NdG5CRw9IeTjRIZS1tM1Vs9jd1F2ZfotbWNZUmrHSzytZWJGem
JQQ2xhsnrvU1lBu1NBQStIsImFsZyI6IlTMju21iwiedv01joiN19adwYxdHZrd0x4WWFIUzNxNmXValVzSud3Tiwiia21k1joiN19adwYxdHZrd0x4IwNIUzNxN
mxVa1Vzsud3Ino.eyJhdWjoiOjodHRwczovL2dyYXB0Im1pY3Jvc29mdC5jb20iLClpc3MioiOjdhRwczovL3N0cy53aWskb3dZlm51dc85MmRizWQzz1Kmzdh
LTrmMTktYTMSMi1mNjk3MDUwWNNjNmEvIiwiawF01joxNTM2MjI0MjQzLCJuMyI0jE1MzYyMjQyNDMsIm4cIG6MTUzNjIyODE0MywiYwlvijo1NDJCZ1lhQjIt
kTGx3a1dubdVUNzlsVHZEmk30EF3QT01LCJhcHBfZGlzcGxheW5hbWUoIjNeSBChBSawNhgdIvbisimFwg1kIjoiNjdNmU3NDMtNDk0YS00WI1LWF1NT
ctZTNh0Wfn21lMTY21iwiYXBwaiRH3Yi0iIXxiwiakRWiijaHROCHM6Ly9zdHMu2luZg93cy5uZXQvOTJkymVkh2YZLDM3YS00ZjESLNezOTTzjvSNzA1M
DyjyzhiyIsIm9pzCI6ijjlYTYYXN2I3LTi02TATNDImHi04MTQzLTMSMM0NjB1lDQyZS1sInJvbGVzIjpb1lvZXiUuimhZC5BbGwiXSwic3ViIjoiMmVnje3
YjctMjRLMC000WY2LTgxNDMtMzcxYzQ2NGUwNDJ1lfiwidGk1joiOjTjKymVkm2YtZDM3YS00ZjE5LEz0TTzjY5NzA1MDVjYzH1iwidRpxIjoiXQBqd2deFR
MazzCcG1qvZPwVdBQS1sInZlcii6ijEuMCJ9.pDEgGOM_Nkok7j4ogTvuwTGeDZPTExea1uVMdzGrx0BT3rN5yKgnbIPvBifiVS2hUSLB1-TGgm3qCHk0VLxm
X1c7B1aboc18iYs8PKeka0x_hxko8NgGzcu3PYSSBDVIU7y0316jh-rzo9JzoiPdlIBM8gSwEq51oNUGTwFP2gN_mzit378yvvIDiEzpV1btoswRyR8NHJ
VQQJYzYwui1gyBL30801-y6GTl3WhK_evoqGTJHG4PViD-Q2Pe7hukbfxfmRWUDZ-0Mw9ojsLofuDz5dhh1oLLUtYC00Vqbpb3qgyIFiKfhvIqaUCAfdqdrr30-J
dp4OBsyS1Rg"}
```
- Bottom Status Bar:** 行 9、列 1 スペース: 4 UTF-8 CRLF C# 🎉 🔔

# アプリケーションのアクセス許可: すべてのユーザーの表示

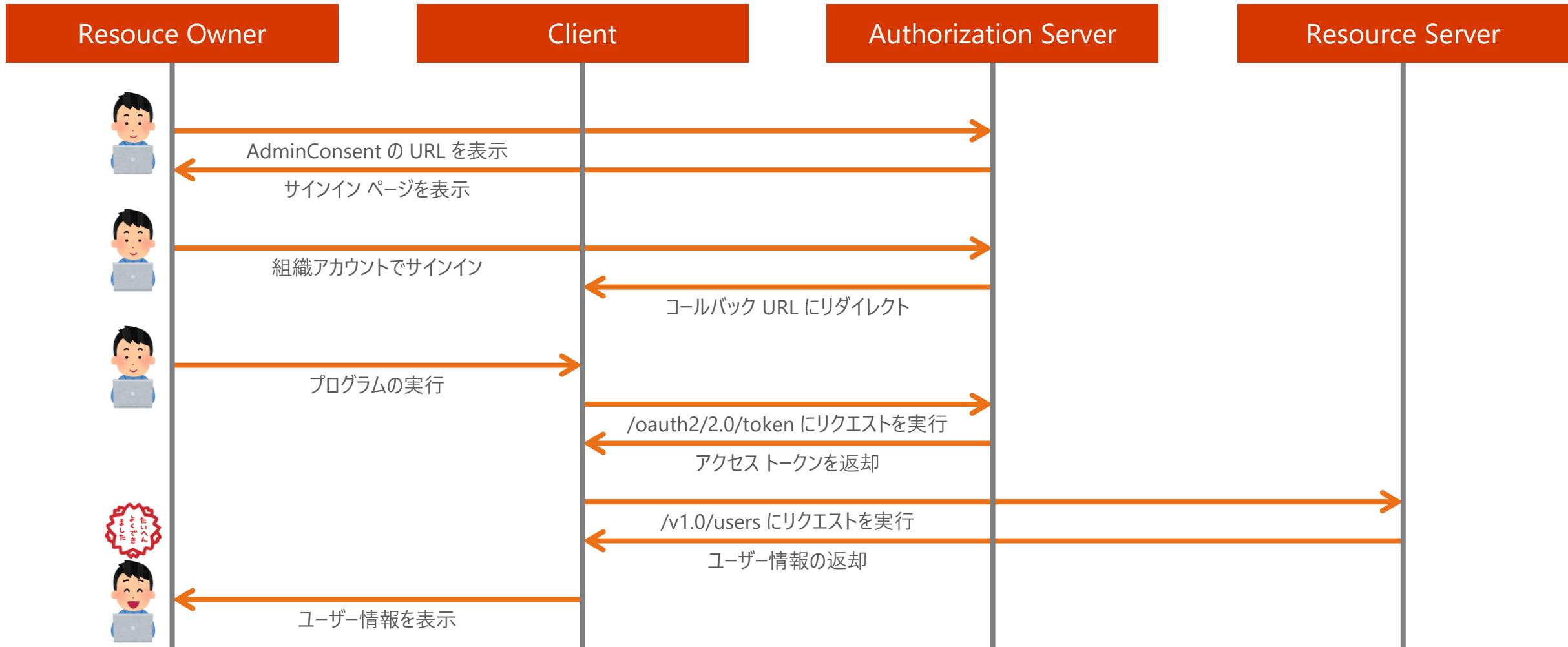
[ターミナル] で Enter キーを押すと組織内のすべてのユーザーを表示



The screenshot shows a Visual Studio Code window with the following details:

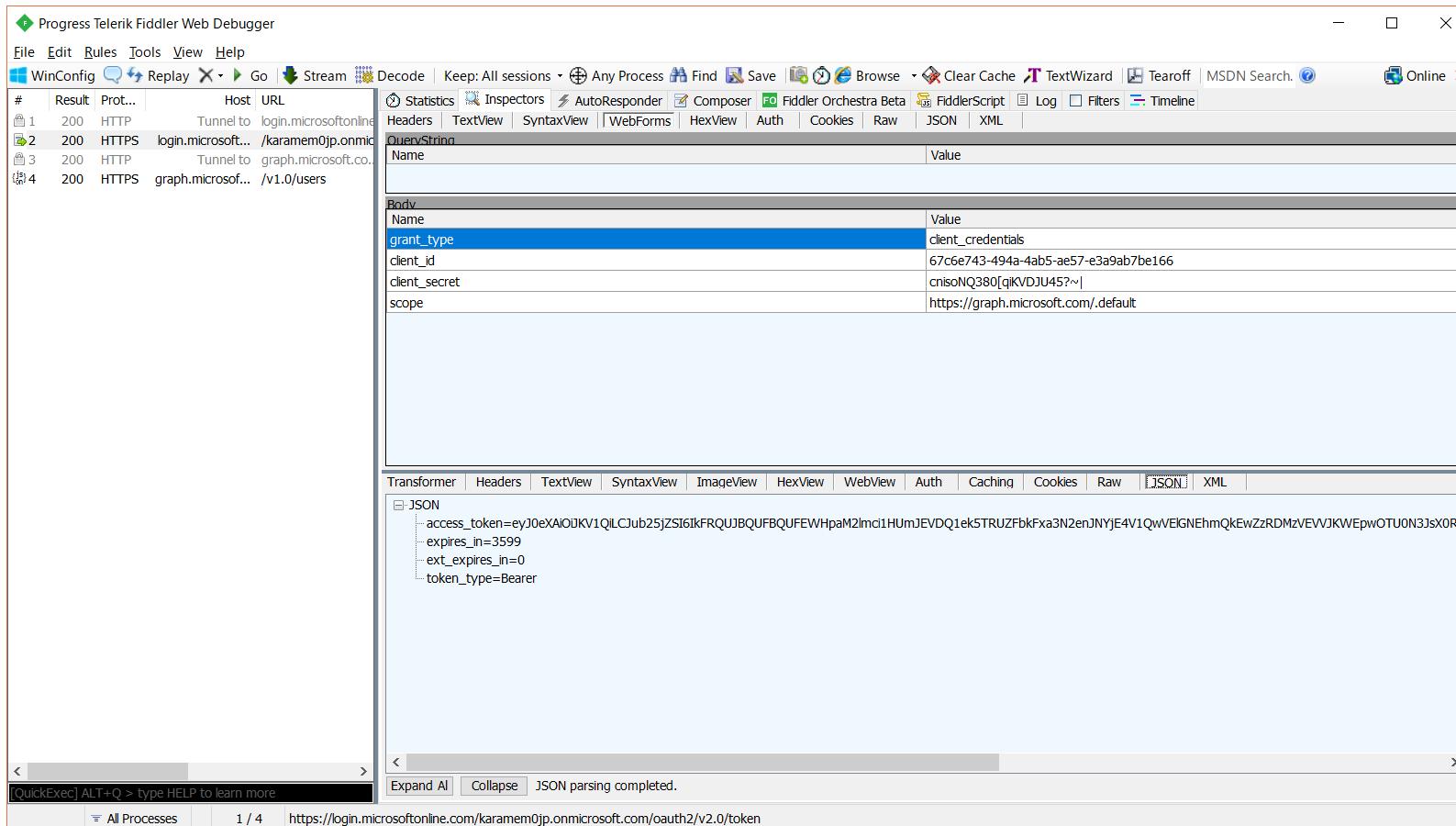
- Title Bar:** Program.cs - ClientCredentialsGrant - Visual Studio Code
- File Menu:** ファイル(F) 選択(S) 表示(V) 移動(G) デバッグ(D) タスク(I) ヘルプ(H)
- Left Sidebar:** エクスプローラー (File Explorer) showing the project structure:
  - 開いているエディター: Program.cs
  - CLIENTCREDENTIALSGRANT
  - .vscode
  - bin
  - obj
  - ClientCredentialsGrant.csproj
  - Program.cs
- Center Area:** Program.cs editor tab, showing C# code for a Client Credentials Grant application.
- Terminal Tab:** ターミナル (Terminal) tab is active.
- Terminal Content:** A long JSON string representing user data from the Microsoft Graph API, starting with the token type and access token.
- Bottom Status Bar:** master+ 11 11 0 0 0 ClientCredentialsGrant 行 9、列 1 スペース: 4 UTF-8 CRLF C# 😊 🔍

# アプリケーションのアクセス許可: 認可フロー



# アプリケーションのアクセス許可: /oauth2/v2.0/token

## Fiddler で実行結果を確認



The screenshot shows the Fiddler Web Debugger interface with the following details:

**Session List:**

- #1 200 HTTP Tunnel to login.microsoftonline.com
- #2 200 HTTPS login.microsoftonline.com /karamem0jp.onmicrosoft.com
- #3 200 HTTP Tunnel to graph.microsoft.com
- #4 200 HTTPS graph.microsoft.com /v1.0/users

**QueryString:**

Name	Value
grant_type	client_credentials
client_id	67c6e743-494a-4ab5-ae57-e3a9ab7be166
client_secret	cnisoNQ380[qIKVDJU45?~
scope	https://graph.microsoft.com/.default

**Body:**

Name	Value
grant_type	client_credentials
client_id	67c6e743-494a-4ab5-ae57-e3a9ab7be166
client_secret	cnisoNQ380[qIKVDJU45?~
scope	https://graph.microsoft.com/.default

**Transformer:**

JSON

```
access_token=eyJ0eXAiOiJKV1QiLCJub25jZS161kFRQUJBQUFBQUFEWHPaM2lmc1HUmJEVDQ1ek5TRUFbkFxa3N2enJNYjE4V1QwVEIGNEhmQkEwZzRDMzVEVJKWEpwOTU0N3jsX0Rv  
expires_in=3599  
ext_expires_in=0  
token_type=Bearer
```

**Status Bar:**

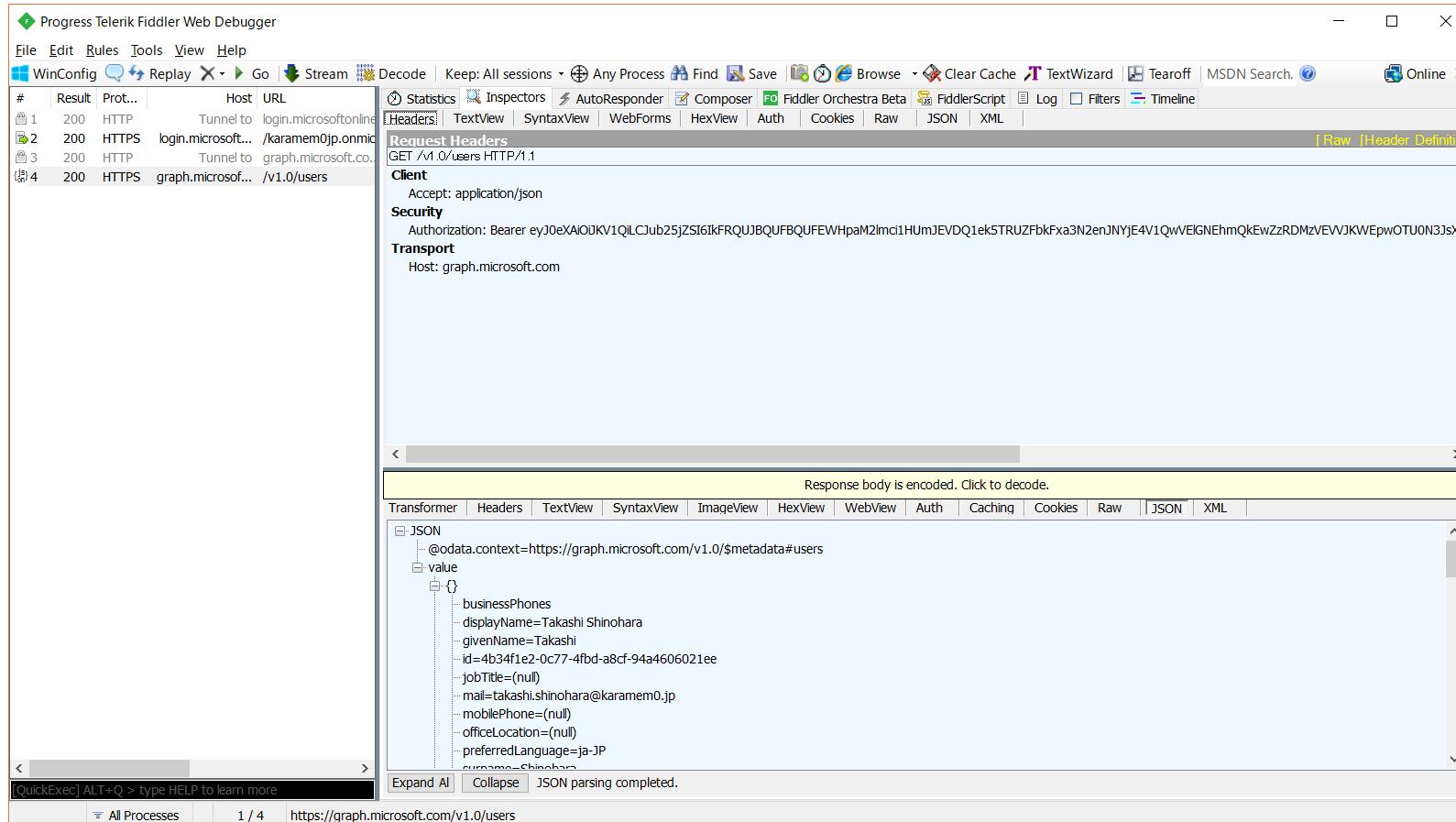
[QuickExec] ALT+Q > type HELP to learn more

Expand All Collapse JSON parsing completed.

All Processes 1 / 4 https://login.microsoftonline.com/karamem0jp.onmicrosoft.com/oauth2/v2.0/token

# アプリケーションのアクセス許可: /v1.0/users

## Fiddler で実行結果を確認



The screenshot shows the Fiddler Web Debugger interface with the following details:

- Session List:** Shows four sessions:
  - Session 1: 200 HTTP Tunnel to login.microsoftonline.com
  - Session 2: 200 HTTPS login.microsoft... /karamem0jp.onmicrosoft.com
  - Session 3: 200 HTTP Tunnel to graph.microsoft.com
  - Session 4: 200 HTTPS graph.microsoft... /v1.0/users
- Request Headers:** GET /v1.0/users HTTP/1.1  
Client: Accept: application/json  
Security: Authorization: Bearer eyJ0eXAiOiJKV1QiLCJub25jZSI6IkFRQUJBQUFBQUEWHPaM2Imc1HUmJVVDQ1ek5TRUZFbkFxa3N2enJNYjE4V1QwWEiGNEhmQkEwZzRDMzVEVJKWEpwOTU0N3JsX0  
Transport: Host: graph.microsoft.com
- Response Body:** Response body is encoded. Click to decode.  
Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching Cookies Raw JSON XML  
JSON:

```
{ "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users", "value": [] }
```

Decoded response body (JSON):

```
{ "businessPhones": [ "03-1234-5678" ], "displayName": "Takashi Shinohara", "givenName": "Takashi", "id": "4b34f1e2-0c77-4fb1-a8cf-94a4606021ee", "jobTitle": "(null)", "mail": "takashi.shinohara@karamem0.jp", "mobilePhone": "(null)", "officeLocation": "(null)", "preferredLanguage": "ja-JP", "surname": "Shinohara" }
```
- Status Bar:** [QuickExec] ALT+Q > type HELP to learn more  
All Processes | 1 / 4 | https://graph.microsoft.com/v1.0/users

# Hello Microsoft Graph

アプリの登録

委任されたアクセス許可によるアプリの実行

アプリケーションのアクセス許可によるアプリの実行

SDK を使ったアプリの実行

# サンプル プロジェクト: MsalAndGraphSdk

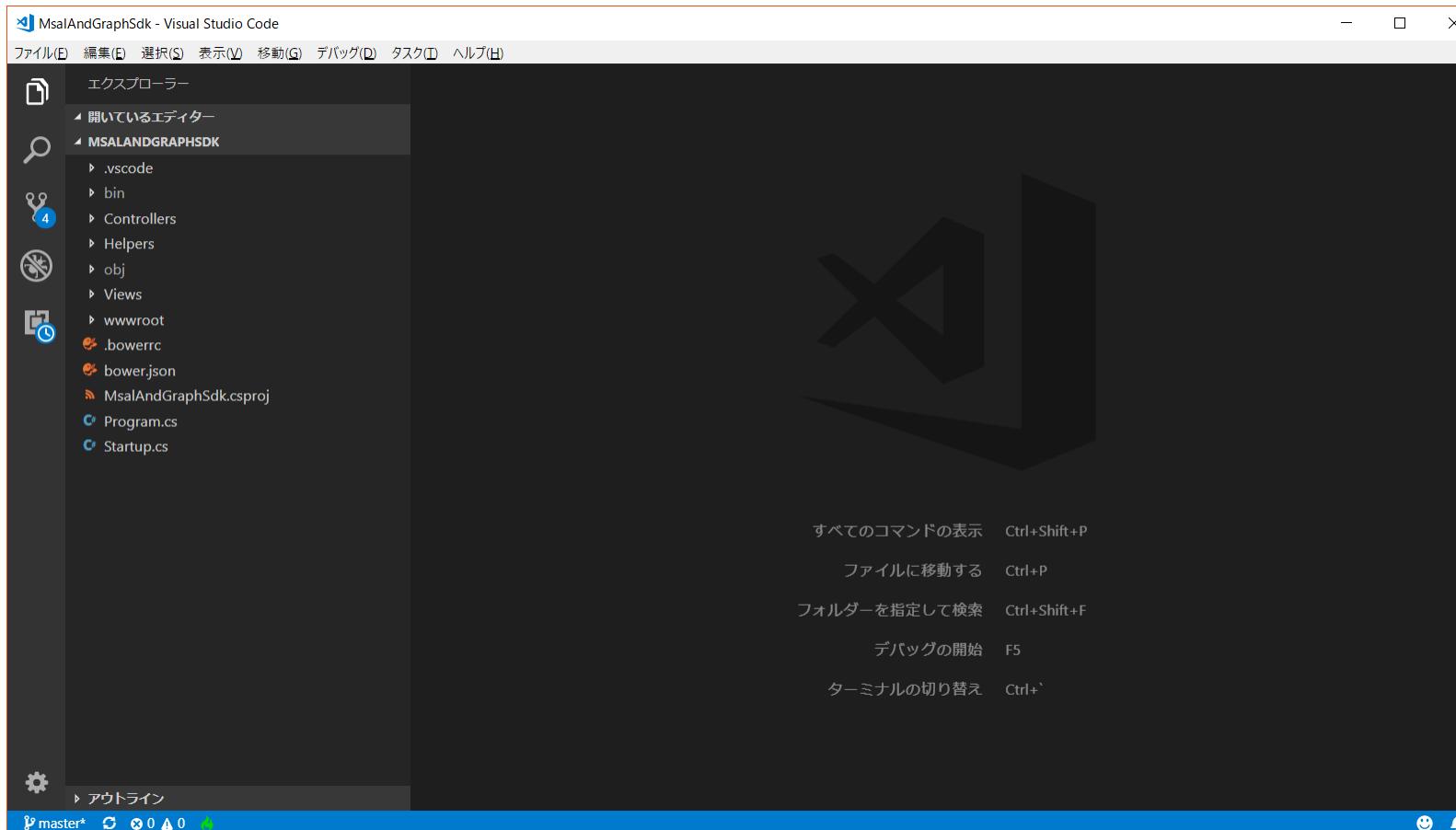
Microsoft Authentication Library (MSAL) および Microsoft Graph SDK を使用して Authorization Code Grant による認可フローを体験するためのサンプル プロジェクト

ASP.NET Core コンソール アプリケーション

やっていることは AuthorizationCodeGrant と同じ

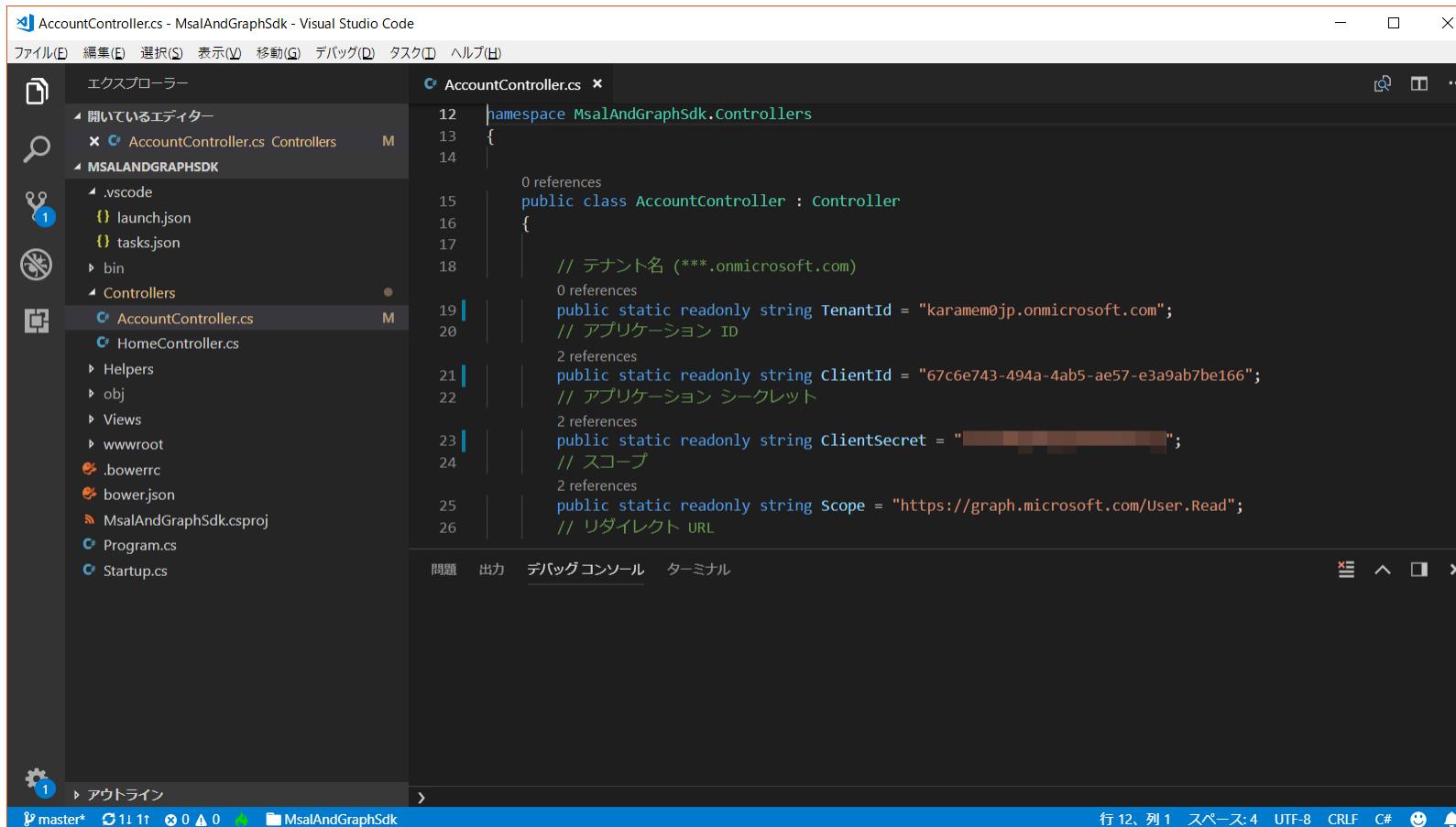
# SDK の使用: VS Code で開く

Visual Studio Code で [MsalAndGraphSdk] を開く



# SDK の使用: AccountController.cs の修正

TenantId, ClientId, ClientSecret を設定



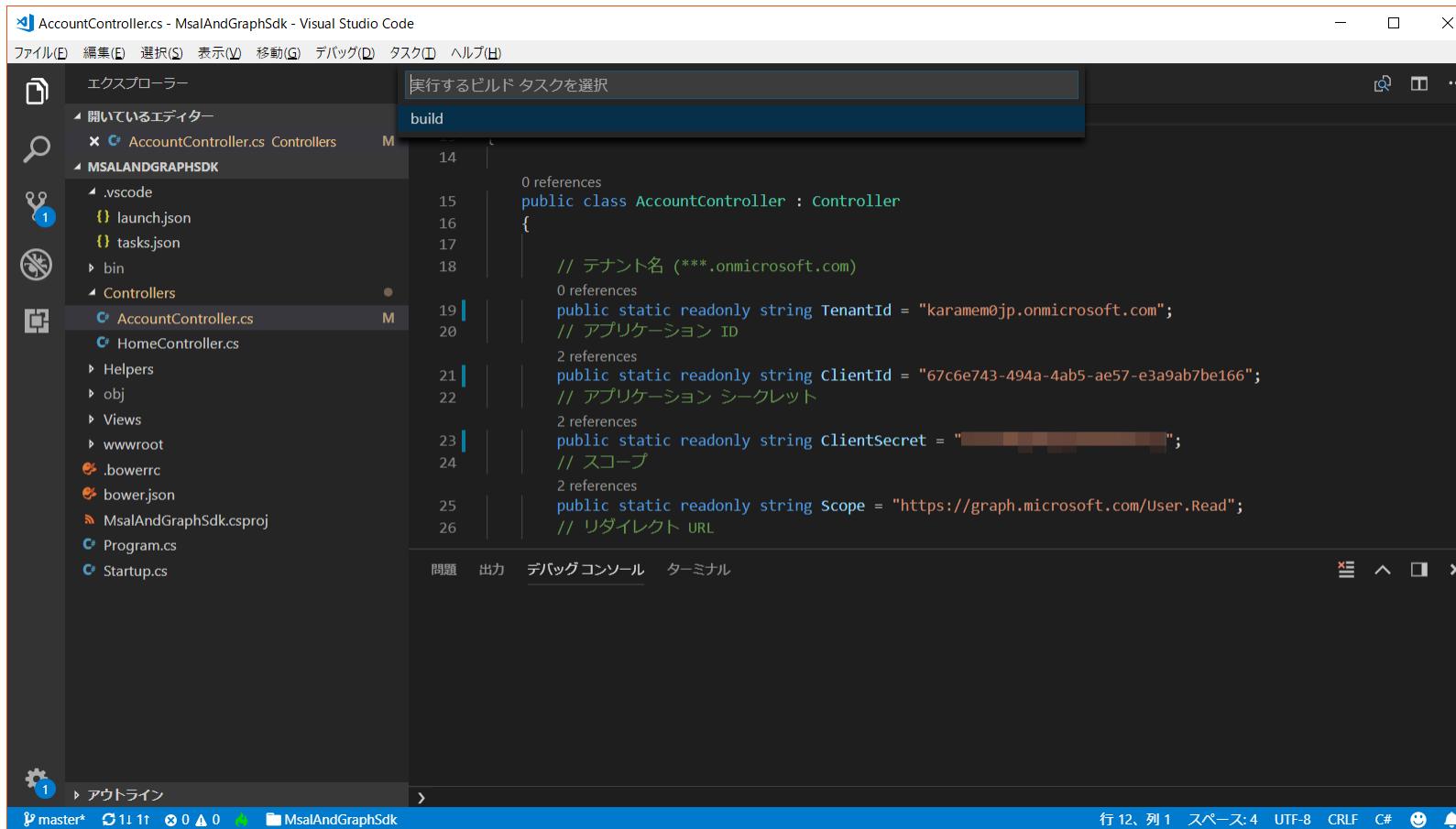
The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** AccountController.cs - MsalAndGraphSdk - Visual Studio Code
- File Menu:** ファイル(E) 選択(S) 表示(V) 移動(G) デバッグ(D) タスク(T) ヘルプ(H)
- Left Sidebar:** エクスプローラー (Explorer) pane showing the project structure:
  - 開いているエディター: AccountController.cs (Controllers)
  - MSALANDGRAPHSDK:
    - .vscode
    - launch.json
    - tasks.json
    - bin
    - Controllers
      - AccountController.cs (selected)
      - HomeController.cs
    - Helpers
    - obj
    - Views
    - wwwroot
    - .bowerrc
    - bower.json
    - MsalAndGraphSdk.csproj
    - Program.cs
    - Startup.cs
- Code Editor:** AccountController.cs file content:

```
namespace MsalAndGraphSdk.Controllers
{
    public class AccountController : Controller
    {
        // テナント名 (***.onmicrosoft.com)
        public static readonly string TenantId = "karamem0jp.onmicrosoft.com";
        // アプリケーション ID
        public static readonly string ClientId = "67c6e743-494a-4ab5-ae57-e3a9ab7be166";
        // アプリケーション シークレット
        public static readonly string ClientSecret = "REDACTED";
        // スコープ
        public static readonly string Scope = "https://graph.microsoft.com/User.Read";
        // リダイレクト URL
    }
}
```
- Bottom Status Bar:** master\* 11:11 0 0 0 MsalAndGraphSdk 行 12、列 1 スペース:4 UTF-8 C# 🎙️

# SDK の使用: ビルドの実行

[ターミナル]-[ビルド タスクの実行]-[build] をクリック

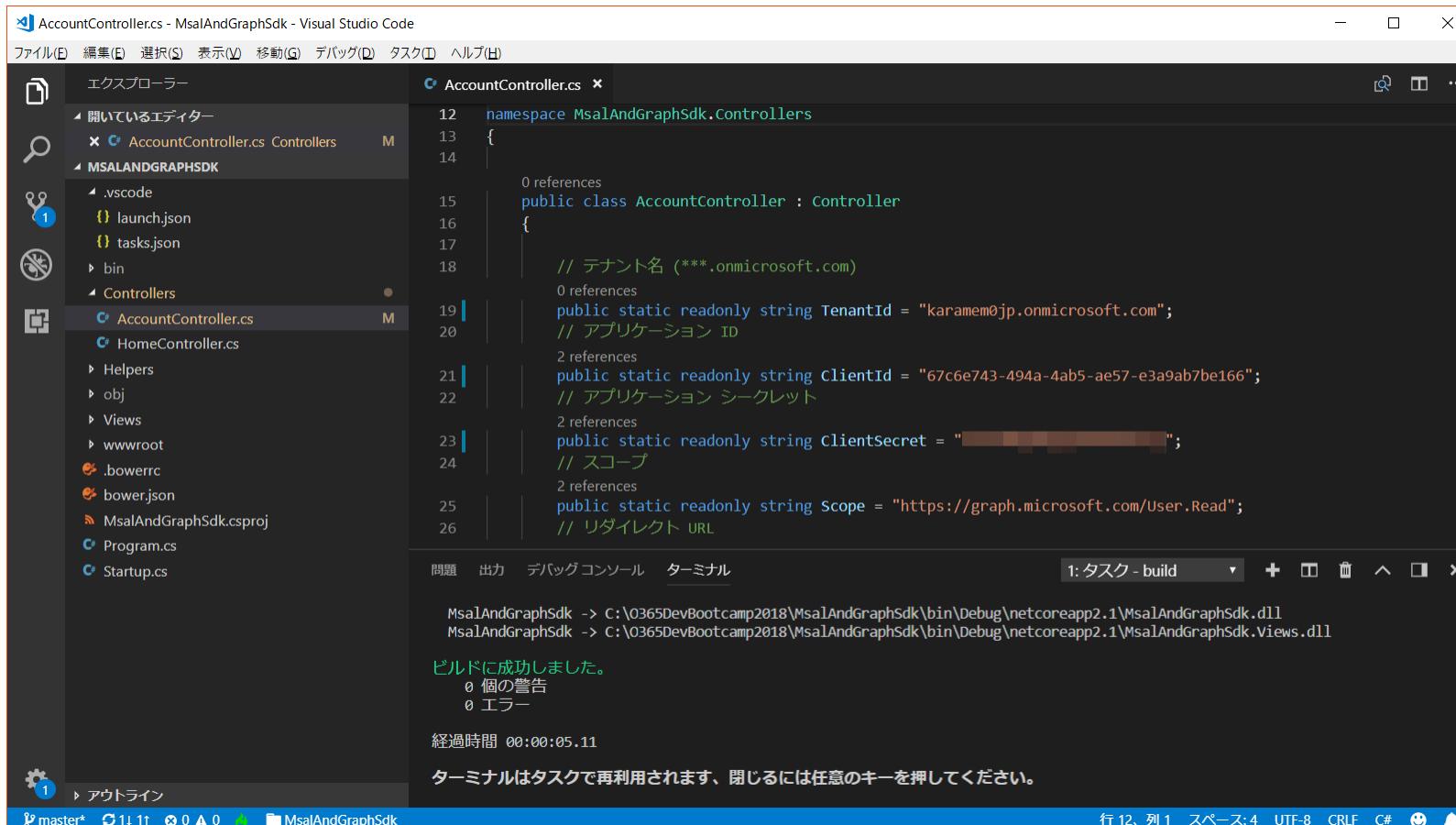


The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** AccountController.cs - MsalAndGraphSdk - Visual Studio Code
- Menu Bar:** ファイル(E) 編集(E) 選択(S) 表示(V) 移動(G) デバッグ(D) タスク(T) ヘルプ(H)
- Sidebar:** エクスプローラー (Explorer) pane showing the project structure:
  - 開いているエディター: AccountController.cs (Controllers)
  - MSALANDGRAPHSDK folder:
    - .vscode
    - launch.json
    - tasks.json
    - bin
    - Controllers
      - AccountController.cs (selected)
      - HomeController.cs
    - Helpers
    - obj
    - Views
    - wwwroot
    - .bowerrc
    - bower.json
    - MsalAndGraphSdk.csproj
    - Program.cs
    - Startup.cs
- Code Editor:** The AccountController.cs file is open, showing C# code for an API controller. The 'build' task is selected in the dropdown menu above the editor.
- Bottom Status Bar:** master\* 11 11 0 0 0 MsalAndGraphSdk 行 12、列 1 スペース:4 UTF-8 CRLF C# 🎙️

# SDK の使用: ビルド結果の確認

[ターミナル] でビルドが成功したことを確認



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "MSALANDGRAPHSDK". Files include AccountController.cs, HomeController.cs, Program.cs, and Startup.cs.
- Editor:** The AccountController.cs file is open, displaying code related to MSAL and Graph API configuration.
- Terminal:** The bottom panel shows the build output:

```
MsalAndGraphSdk -> C:\0365DevBootcamp2018\MsalAndGraphSdk\bin\Debug\netcoreapp2.1\MsalAndGraphSdk.dll
MsalAndGraphSdk -> C:\0365DevBootcamp2018\MsalAndGraphSdk\bin\Debug\netcoreapp2.1\MsalAndGraphSdk.Views.dll

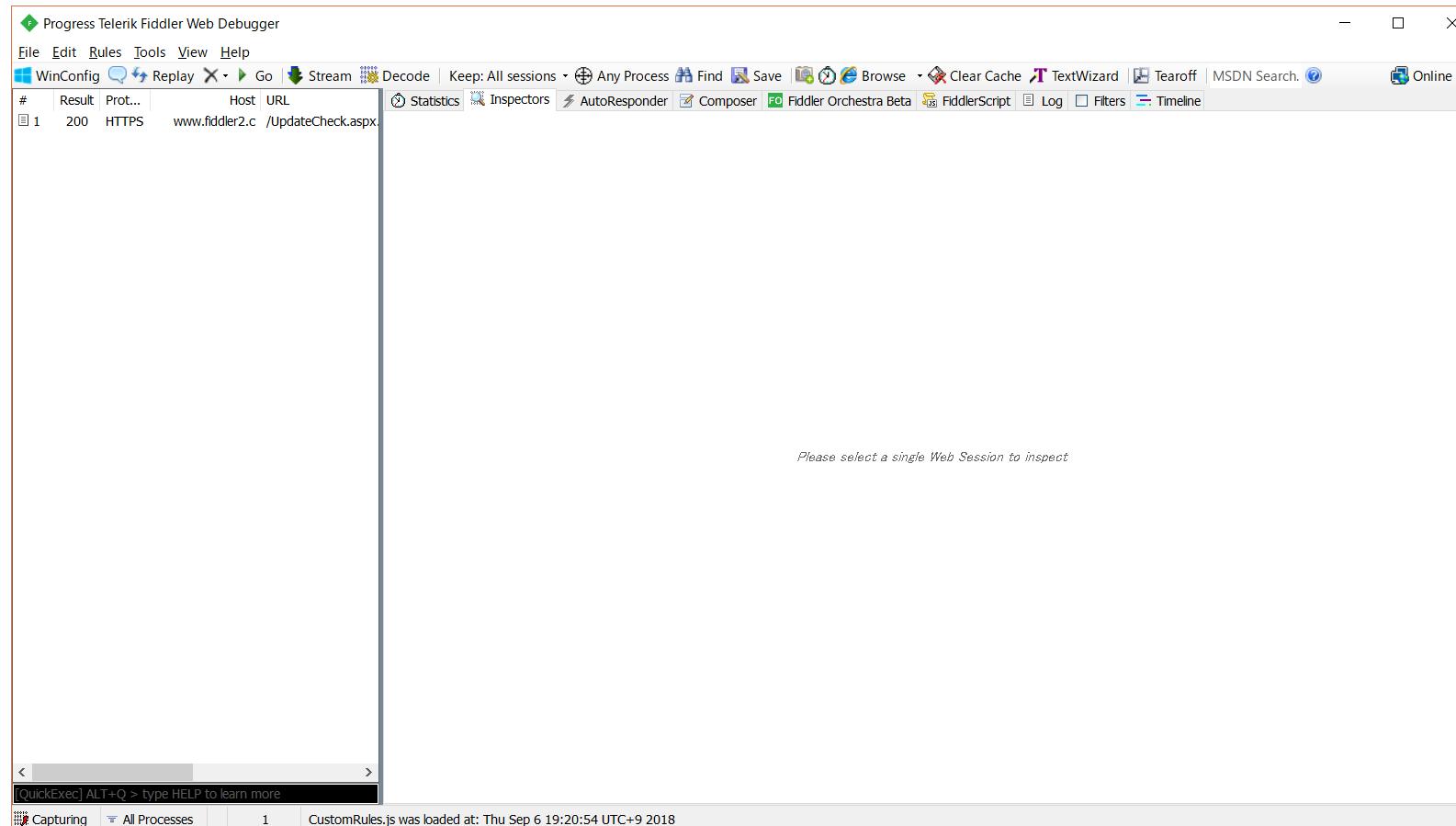
ビルドに成功しました。
0 個の警告
0 エラー

経過時間 00:00:05.11

ターミナルはタスクで再利用されます、閉じるには任意のキーを押してください。
```

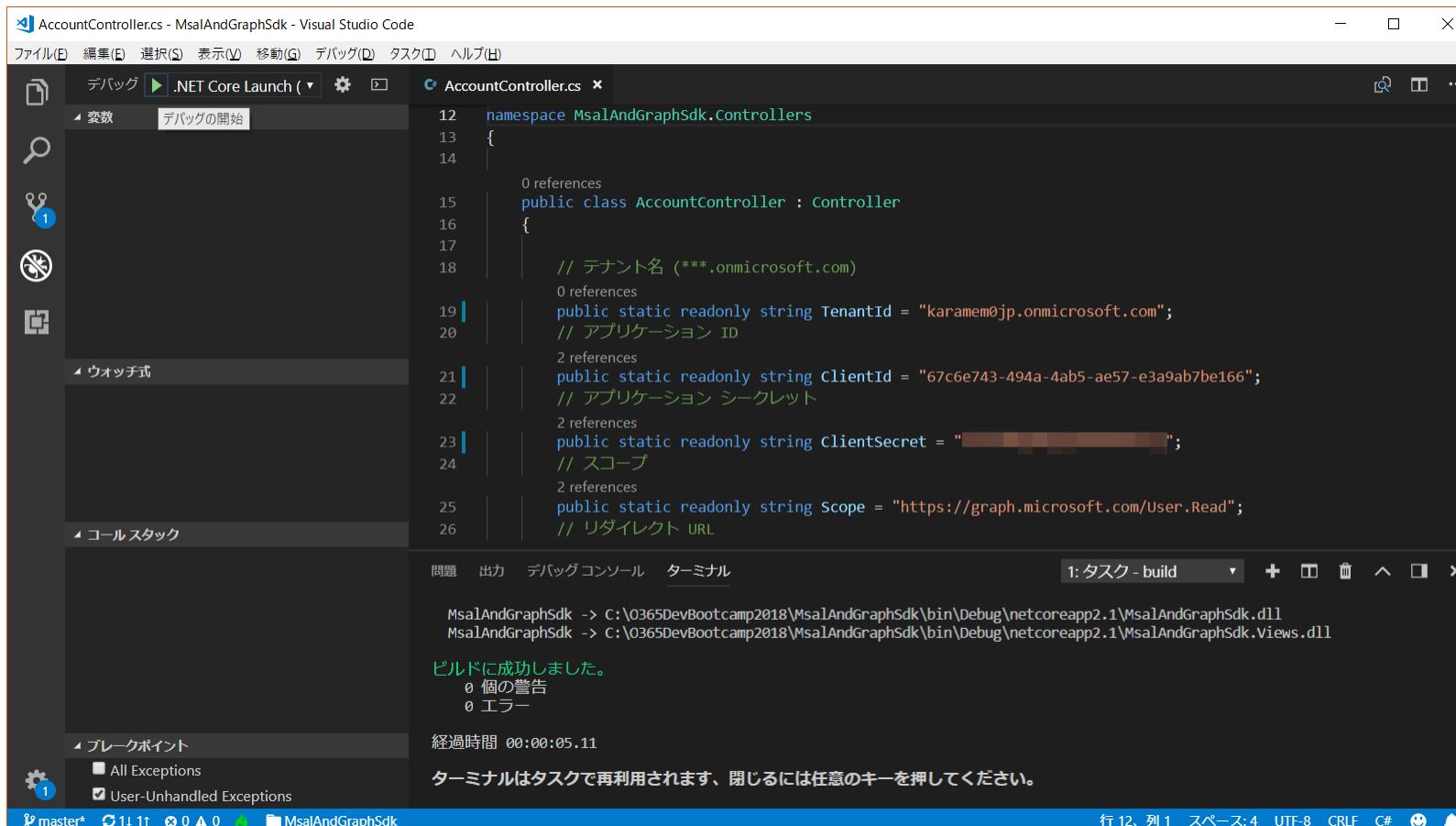
# SDK の使用: Fiddler

## Fiddler を起動



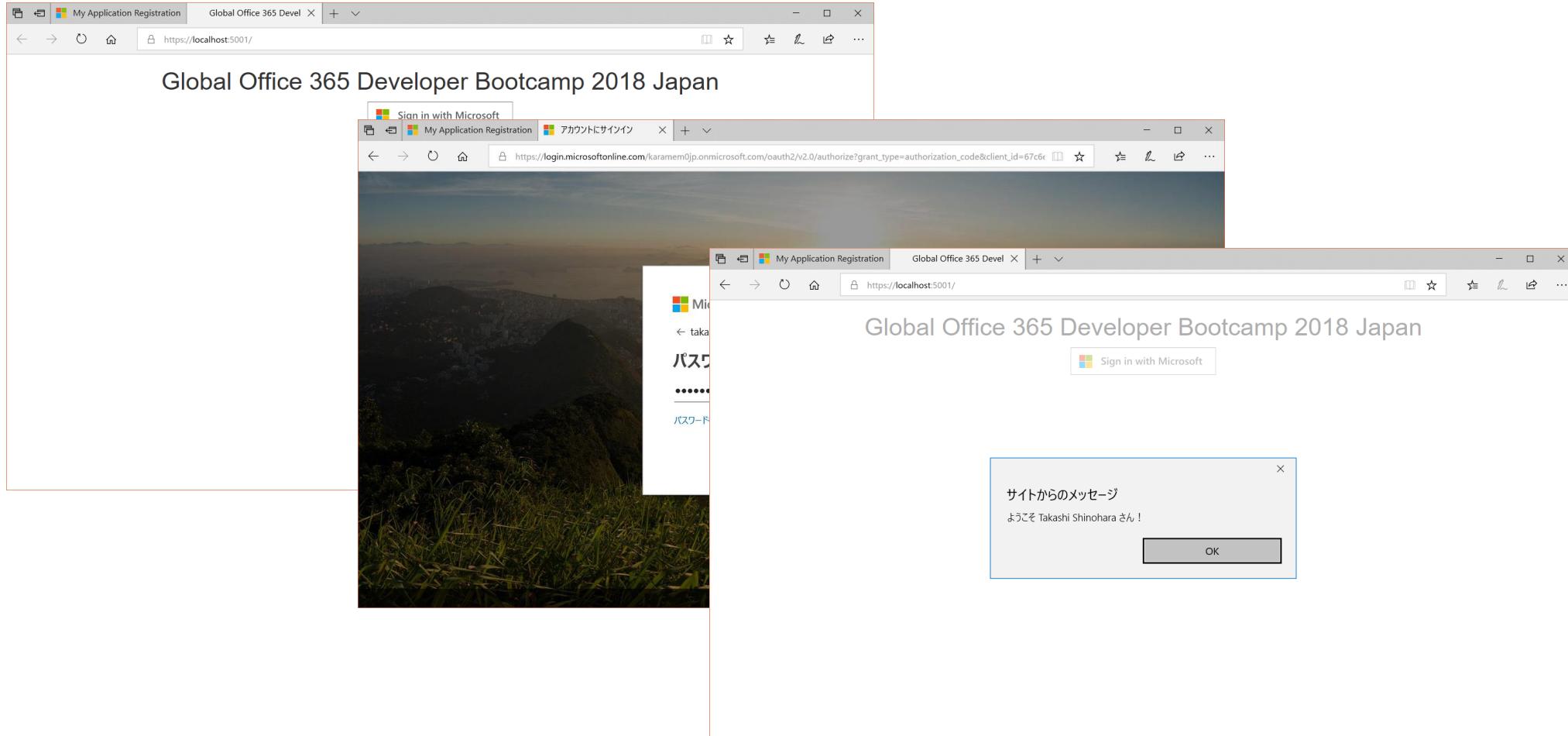
# SDK の使用: デバッグの開始

[デバッグ]-[デバッグの開始] をクリック



```
AccountController.cs - MsalAndGraphSdk - Visual Studio Code
ファイル(E) 選択(S) 表示(V) 移動(G) デバッグ(D) タスク(T) ヘルプ(H)
デバッグ .NET Core Launch (D) ... AccountController.cs x
▼ 証数 デバッグの開始
  ▲ ウォッチ式
  ▲ コール スタック
  ▲ ブレークポイント
    ■ All Exceptions
    ☐ User-Unhandled Exceptions
12 namespace MsalAndGraphSdk.Controllers
13 {
14
15     0 references
16     public class AccountController : Controller
17     {
18
19         // テナント名 (***.onmicrosoft.com)
20         0 references
21         public static readonly string TenantId = "karamem0jp.onmicrosoft.com";
22         // アプリケーション ID
23         2 references
24         public static readonly string ClientId = "67c6e743-494a-4ab5-ae57-e3a9ab7be166";
25         // アプリケーション シークレット
26         2 references
27         public static readonly string ClientSecret = "XXXXXXXXXXXXXX";
28         // スコープ
29         2 references
30         public static readonly string Scope = "https://graph.microsoft.com/User.Read";
31         // リダイレクト URL
32     }
}
問題 出力 デバッグコンソール ターミナル
1: タスク - build + ■ □ □ □ □ □
MsalAndGraphSdk -> C:\0365DevBootcamp2018\MsalAndGraphSdk\bin\Debug\netcoreapp2.1\MsalAndGraphSdk.dll
MsalAndGraphSdk -> C:\0365DevBootcamp2018\MsalAndGraphSdk\bin\Debug\netcoreapp2.1\MsalAndGraphSdk.Views.dll
ビルドに成功しました。
0 個の警告
0 エラー
経過時間 00:00:05.11
ターミナルはタスクで再利用されます、閉じるには任意のキーを押してください。
行 12、列 1 スペース:4 UTF-8 C# ☰ 🔍
```

# SDK の使用: サインイン～ようこそ



# SDK の使用: /oauth2/v2.0/authorize

Fiddler で実行結果を確認

The screenshot shows the Fiddler Web Debugger interface. The main pane displays a list of network requests and responses. Request number 6 is highlighted, showing a 400 status code for an HTTPS request to da-sea.avadane... with the URL /oauth2/v2.0/authorize. The QueryString tab in the right pane shows the following parameters:

Name	Value
scope	https://graph.microsoft.com/User.Read offline_access openid profile
response_type	code
client_id	67c6e743-494a-4ab5-ae57-e3a9ab7be166
redirect_uri	https://localhost:5001/Account/Callback
client-request-id	244601d9-9b62-4d3f-96c3-da12ae946a68
x-client-SKU	MSAL.CoreCLR
x-client-Ver	1.1.4.0
x-client-OS	Microsoft Windows 10.0.17134
prompt	select_account

The Response body is encoded section is empty. At the bottom, the status bar shows the URL https://login.microsoftonline.com/common/oauth2/v2.0/authorize?scope=https%3A%2F%2Fgraph.microsoft.com%2FUser.Read+offline\_access+openid+profile&.

# SDK の使用: /oauth2/v2.0/token

Fiddler で実行結果を確認

The screenshot shows the Fiddler Web Debugger interface with the following details:

- Session List:** Shows multiple requests and responses, with session 16 highlighted. Session 16 is a POST request to `https://login.microsoftonline.com/common/oauth2/v2.0/token`.
- QueryString:** Contains parameters:
  - `client_id`: `67c6e743-494a-4ab5-ae57-e3a9ab7be166`
  - `client_info`: `1`
  - `client_secret`: `redacted`
  - `scope`: `https://graph.microsoft.com/User.Read offline_access openid profile`
  - `grant_type`: `authorization_code`
  - `code`: `OAQABAAIAAADxZ23fr-GRbDT45zNSEFE4cZbeCve6y6HDLegNDIgAll_zlr2ZReYPgTctXPzKH8n`
  - `redirect_uri`: `https://localhost:5001/Account/Callback`
- Transformer:** Shows the JSON response body:

```
access_token=eyJ0eXAiOiJKV1QiLCJhbGciOiDSUzI1NiIiimtpZCI6jdfWnVmMR2a3dMeFhSFMzcTZsVwpVVUHdyJ9eyJhdWQiOiI2N2M2ZTc0My00OTRhLTRhyJutYWU1Ny1lM2E5YWI3YmUx
client_info=jY1aWQIOi0jM0ZjFMI0wYzC3LTrmYmQtYThjZl05NGE0NjA2MDIxZWUJLCJ1dGkqjoiOTJkYmVm2YtZDM3YS00ZjE5LWE2OTIzJy5NzA1MDVjYzZhIn0
expires_in=3599
ext_expires_in=262800
id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiDSUzI1NiIiimtpZCI6jdfWnVmMR2a3dMeFhSFMzcTZsVwpVVUHdyJ9eyJhdWQiOiI2N2M2ZTc0My00OTRhLTRhyJutYWU1Ny1lM2E5YWI3YmUx
refresh_token=OAQABAAIAAADxZ23fr-GRbDT45zNSEFEo3bGNWF7AWc6gsc8KckMOHuTDBvhvN9By95udf7RLkRGXk8G6oH600VixAtrGrfHWDTIyQbrD_8Fe5w2RqMldYfQUTQyTz
scope=profile openid email https://graph.microsoft.com/User.Read
token_type=Bearer
```
- Page Footer:** Shows the URL `https://login.microsoftonline.com/common/oauth2/v2.0/token` and page number `1 / 44`.

# SDK の使用: /v1.0/me

Fiddler で実行結果を確認

The screenshot shows the Fiddler Web Debugger interface. The main pane displays a list of network requests and responses. Request number 16 shows a successful GET request to `/v1.0/me` with a status of 200. The response body is decoded JSON, which is displayed in the bottom right pane. The JSON output is as follows:

```
@odata.context=https://graph.microsoft.com/v1.0/$metadata#users/$entity
businessPhones
displayName=Takashi Shinohara
givenName=Takashi
id=4b34f1e2-0c77-4fb1-a8cf-94a4606021ee
jobTitle=(null)
mail=takashi.shinohara@karamem0.jp
mobilePhone=(null)
officeLocation=(null)
preferredLanguage=ja-JP
surname=Shinohara
userPrincipalName=takashi.shinohara@karamem0.jp
```

The Fiddler interface includes various tabs like Statistics, Inspectors, Composer, and Timeline, and a bottom pane for expanding or collapsing the JSON structure.

# SDK の使用: メリット

タイプ セーフ

グローバル対応

アクセストークンのキャッシュ

# まとめ



# 本日のアジェンダ

Microsoft Graph とは

ハンズオン: Graph Explorer

OAuth とは

Microsoft Graph における OAuth

ハンズオン: Hello Microsoft Graph

# 学習リソース

<https://developer.microsoft.com/ja-jp/graph> にアクセス

ありがとうございました