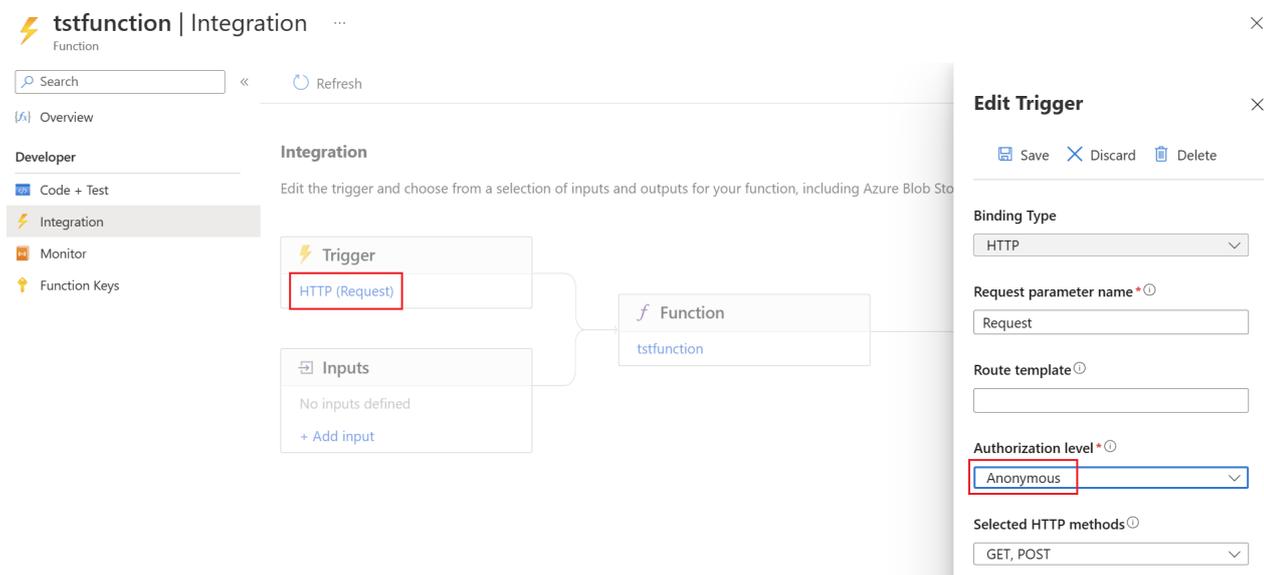


# Access Azure Function App via OAuth 2.0 authentication

This is a guide to protect Azure Function executions using OAuth 2.0. So the execution of the code is not possible without Client ID and ClientSecret. This allows a much more secure authentication than just using function codes in the URL in the query.

## Disable authentication

To use the function with OAuth 2.0, the authentication on the function itself must first be set to Anonymous.



The screenshot displays the Azure Portal interface for configuring an Azure Function. The main area shows the 'Integration' blade with a diagram of the function's trigger and inputs. The 'Trigger' section is expanded, and 'HTTP (Request)' is selected. The 'Inputs' section shows 'No inputs defined' and a '+ Add input' button. On the right, the 'Edit Trigger' pane is open, showing the following configuration:

- Binding Type: HTTP
- Request parameter name: Request
- Route template: (empty)
- Authorization level: Anonymous
- Selected HTTP methods: GET, POST

## Identity provider

Then a new identity provider must be added to the Azure Function App. This can be done by going into the blade "Authentication":

Search

Refresh Send us your feedback

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Microsoft Defender for Cloud
- Events (preview)
- Functions**
- Functions
- App keys
- App files
- Proxies
- Deployment**
- Deployment slots
- Deployment Center
- Settings**
- Configuration
- Authentication



### Add an identity provider

Choose an identity provider to manage the user identities and authentication flow for your application. Providers include Microsoft, Facebook, Google, and Twitter.

[Learn more about identity providers](#)

Add identity provider

There you have to select "Microsoft" as the identity provider. There you can decide if you want to use an existing App Registration or want to create one.

# Add an identity provider ...

Basics Permissions

Identity provider \*

Microsoft

## App registration

An app registration associates your identity provider with your app. Enter the app registration information here, or go to your provider to create a new one. [Learn more](#)

App registration type \*

- Create new app registration
- Pick an existing app registration in this directory
- Provide the details of an existing app registration

Name \* ⓘ

<newappregistrationname>

Supported account types \*

- Current tenant - Single tenant
- Any Azure AD directory - Multi-tenant
- Any Azure AD directory & personal Microsoft accounts
- Personal Microsoft accounts only

[Help me choose...](#)

## App Service authentication settings

Requiring authentication ensures all users of your app will need to authenticate. If you allow unauthenticated requests, you'll need your own code for specific authentication requirements. [Learn more](#)

Restrict access \*

- Require authentication
- Allow unauthenticated access

Unauthenticated requests \*

- HTTP 302 Found redirect: recommended for websites
- HTTP 401 Unauthorized: recommended for APIs
- HTTP 403 Forbidden
- HTTP 404 Not found

Token store ⓘ



Add

< Previous

Next: Permissions >

It is also recommended to send a 401 Unauthorized Response for incorrectly authenticated requests.

Afterwards, the app registration has to be adjusted so that the token handling works properly. To adjust the URL, the identity provider must be adjusted using "Edit".

Search << Refresh Send us your feedback

You can choose an identity provider to manage user identities and authentication flows. Add providers here, edit settings, and decide which provider is handling authentication for your app. [Learn more](#)

**Authentication settings** [Edit](#)

App Service authentication Enabled

Restrict access Require authentication

Unauthenticated requests Return HTTP 401 Unauthorized

Token store Enabled

**Identity provider**

+ Add provider

Identity provider	App (client) ID	Learn more	Edit	Delete
Micr... (func-bkw-capri01-...	835d3048-d3f0-4816-9217-678e0b22...	<a href="#">Quickstart</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

The issuer URL must be adjusted. The "/v.2.0" at the end must be removed

Basics Permissions

**i** To edit the app registration name or account types, go to the app registration menu in Azure Active Directory. [Click here to access App registrations.](#)

Identity provider	Microsoft
App registration	[Redacted]
Supported account types	Current tenant - Single tenant
Application (client) ID	[Redacted] <a href="#">Copy</a>
Client secret value	<a href="#">Click to edit secret value</a>
Client secret setting name <sup>ⓘ</sup>	MICROSOFT_PROVIDER_AUTHENTICATION_SECRET
Issuer URL <sup>ⓘ</sup>	<a href="https://sts.windows.net/[Redacted]/v2.0">https://sts.windows.net/[Redacted]/v2.0</a>
Allowed token audiences	api://[Redacted] <a href="#">Delete</a> <a href="#">More</a>
	<input type="text" value="Enter allowed token audience value"/>

# Authentication via PowerShell

Then PowerShell can be used to authenticate against the app registration. The App Registration then has permissions to execute all Azure Functions in the Azure Function App.

```
$TenantId = "<yourtenantid>"
$ClientID = "<yourclientid>"
$ClientSecret = "<yourclientsecret>"

$FunctionAppId = "<yourfunctionappid>"
$FunctionApiAuthUrl = "$functionuri/.auth/login/aad"
$functionapi = "/api/HttpTrigger2"

# Authenticate against MEID to get access token with App Registration Client Secret
$Body = @{
    "tenant" = "$TenantId"
    "client_id" = "$ClientID"
    "scope" = "api://$functionappid/default"
    "grant_type" = "client_credentials"
    "client_secret" = $ClientSecret
}

$Params = @{
    "Uri" = "https://login.microsoftonline.com/$TenantId/oauth2/v2.0/token"
    "Method" = "Post"
    "Body" = $Body
    "ContentType" = "application/x-www-form-urlencoded"
}

$AuthResponse = Invoke-RestMethod @Params
```

## Function execution via PowerShell

The second part of the authentication is to ask the function api for a token and then execute it using the token received:

```
# Authenticate against function with the MEID access token
$FunctionAuthBody = @{
    "access_token" = $AuthResponse.access_token
}

$functionToken = Invoke-RestMethod -Method POST -Uri $FunctionApiAuthUrl -Body (ConvertTo-Json
```

```
$FunctionAuthBody) -ContentType "application/json"
```

```
$Header = @{
```

```
  "X-ZUMO-AUTH" = $functionToken.authenticationToken
```

```
}
```

```
# Run Azure Function with OAuth2.0 Token Authentication
```

```
Invoke-RestMethod -Method POST -Uri $functionuri$functionapi -Headers $Header
```

---

Revision #2

Created 9 January 2023 14:20:35

Updated 21 July 2024 15:11:16