

08 - Implementarea funcțiilor Azure (5 min)

În această demonstrație, vom crea o aplicație funcțională pentru a afișa un mesaj de salut atunci când există o solicitare HTTP.

Sarcina 1: Crearea unei aplicații funcționale

În această sarcină, vom crea o aplicație Function.

1. Conectați-vă la [portalul Azure](#).
2. În bara **de căutare** din partea de sus a portalului, căutați și selectați **Aplicație funcțională**, apoi, din lama **Aplicație funcțională**, faceți clic pe **+ Adăugare, + Creare, + Nou**.
3. Pe fila **De bază** a bladei **Aplicație funcții**, specificați următoarele setări (înlocuiți **xxxx** în numele funcției cu litere și cifre astfel încât numele să fie unic la nivel global și lăsați toate celelalte setări cu valorile lor implicate):

Setări	Valoare
Abonament	Păstrați valoarea implicită furnizată
Grup de resurse	Creați un nou grup de resurse
Numele aplicației funcționale	funcție-xxxx
Publica	Cod
Stiva de execuție	.NET
Versiune	3.1
Regiune	Estul SUA

4. **Notă** - Nu uitați să modificați **xxxx** astfel încât să creeze un **nume unic pentru aplicația de funcții**.
5. Faceți clic pe **Revizuire + Creare** și, după validarea cu succes, faceți clic pe **Creare** pentru a începe furnizarea și implementarea noii aplicații Azure Function.
6. Așteptați notificarea că resursa a fost creată.

7. Când implementarea s-a finalizat, faceți clic pe Accesați resursa din lama de implementare. Alternativ, navigați înapoi la lama **Aplicație funcțională**, faceți clic pe **Reîmprospătare** și verificați dacă aplicația funcțională nou creată are starea **În curs de execuție**.

The screenshot shows the 'Function App' list page in the Azure portal. At the top, there are navigation links: 'Home > Function App'. Below that is the title 'Function App' and a 'Default Directory' link. A toolbar with various actions like 'Add', 'Manage view', 'Refresh', 'Export to CSV', 'Assign tags', 'Start', 'Restart', 'Stop', and 'Delete' is visible. Below the toolbar are filters for 'Subscription == all', 'Resource group == all', 'Location == all', and a 'Filter by name...' input field. The main table displays one record:

<input type="checkbox"/>	Name ↑↓	Status ↑↓	Location ↑↓	Pricing Tier ↑↓	App Service Plan ↑↓
<input type="checkbox"/>	⚡ function-9007	Running	East US	Dynamic	ASP-myRGFunction-a50c

Below the table, it says 'Showing 1 to 1 of 1 records.'

Sarcina 2: Creați o funcție declanșată prin HTTP și testați-o

În această sarcină, vom folosi funcția Webhook + API pentru a afișa un mesaj atunci când există o cerere HTTP.

1. În lama **Aplicație funcțională**, faceți clic pe aplicația funcțională nou creată.
2. În lama aplicației de funcții, în secțiunea **Functii**, faceți clic pe **Functii**, apoi pe **+ Adăugare, + Creare, + Nou**.

The screenshot shows the 'function-9007 | Functions' blade in the Azure portal. At the top, there is a breadcrumb trail: 'Home > Function App > function-9007 | Functions'. Below that is the title 'function-9007 | Functions' and an 'App Service' icon. A toolbar with 'Search (Ctrl+I)', 'Develop Locally', 'Refresh', 'Enable', 'Disable', and 'Delete' buttons is visible. To the left is a sidebar with links: 'Diagnose and solve problems', 'Security', 'Events', 'Functions' (which is selected and highlighted with a red box), 'App keys', 'App files', and 'Proxies'. In the main area, there is a search bar 'Filter by name...', a table header with columns 'Name ↑↓', 'Trigger ↑↓', and 'Status ↑↓', and a message 'No results.' Below the table, there is a large red box around the '+ Add' button.

3. O fereastră pop-up **Adăugare funcție** va apărea în dreapta. În secțiunea **Selectați un şablon**, faceți clic pe **Declanșator HTTP**. Faceți clic pe **Adăugare**.

The screenshot shows the Azure Functions portal interface. On the left, there's a sidebar with various navigation items like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Events (preview), Functions, Deployment, and Settings. The 'Functions' item is highlighted with a red box. At the top center, there's a search bar and a red-bordered '+ Add' button. To the right of the search bar is a 'Refresh' button and a 'Delete' link. Below the search bar is a 'Filter by name...' input field. The main area displays a table with columns 'Name' and 'Trigger', showing 'No results.' A red box highlights the 'Name' column header. On the far right, there's a 'Select development environment' section with a dropdown set to 'Develop in portal' and a 'Select a template' section with a 'Filter' input field. The 'Template' table lists several options: 'HTTP trigger' (selected and highlighted with a red box), 'Timer trigger', 'Azure Queue Storage trigger', 'Azure Service Bus Queue trigger', 'Azure Service Bus Topic trigger', and 'Azure Blob Storage trigger'. Each row has a detailed description. Below the table is a 'Template details' section with fields for 'New Function*' (set to 'HttpTrigger1') and 'Authorization level*' (set to 'Function'). At the bottom are 'Add' and 'Cancel' buttons.

4. Pe lama **HttpTrigger1** , în secțiunea **Dezvoltator** , faceți clic pe **Cod + Test** .
5. În lama **Cod + Test** , examinați codul generat automat și rețineți că acesta este conceput să execute o solicitare HTTP și să înregistreze informații în jurnal. De asemenea, observați că funcția returnează un mesaj Bună ziua cu un nume.

Home > Function App > function-9007 | Functions > HttpTrigger1 | Code + Test

HttpTrigger1 | Code + Test

Function

Search (Ctrl+ /) <

Overview

Developer

- Code + Test
- Integration
- Monitor
- Function Keys

Save Discard Refresh Test Get function URL

function-9007 \ HttpTrigger1 \ run.csx

```

1 #r "Newtonsoft.Json"
2
3 using System.Net;
4 using Microsoft.AspNetCore.Mvc;
5 using Microsoft.Extensions.Primitives;
6 using Newtonsoft.Json;
7
8 public static async Task<IActionResult> Run(HttpContext req, ILogger log)
9 {
10     log.LogInformation("C# HTTP trigger function processed a request.");
11
12     string name = req.Query["name"];
13
14     string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
15     dynamic data = JsonConvert.DeserializeObject(requestBody);
16     name = name ?? data?.name;
17
18     return name != null
19         ? (ActionResult)new OkObjectResult($"Hello, {name}")
20         : new BadRequestObjectResult("Please pass a name on the query string or in the request body");
21 }
22

```

6. Faceți clic pe **Obțineți adresa URL a funcției** din secțiunea superioară a editorului de funcții.
7. Asigurați-vă că valoarea din lista derulantă **Cheie este setată la valoarea implicită** și faceți clic pe **Copiere** pentru a copia adresa URL a funcției.

Home > Function App > function-9007 | Functions > HttpTrigger1 | Code + Test

HttpTrigger1 | Code + Test

Function

Search (Ctrl+ /) <

Overview

Developer

- Code + Test
- Integration
- Monitor
- Function Keys

Save Discard Refresh Test Get function URL

function-9007 \ HttpTrigger1 \ run.csx

```

1 #r "Newtonsoft.Json"
2
3 using System.Net;
4 using Microsoft.AspNetCore.Mvc;
5 using Microsoft.Extensions.Primitives;
6 using Newtonsoft.Json;
7
8 public static async Task<IActionResult> Run(HttpContext req, ILogger log)
9 {
10     log.LogInformation("C# HTTP trigger function processed a request.");
11
12     string name = req.Query["name"];
13
14     string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
15     dynamic data = JsonConvert.DeserializeObject(requestBody);
16     name = name ?? data?.name;
17
18     return name != null
19         ? (ActionResult)new OkObjectResult($"Hello, {name}")
20         : new BadRequestObjectResult("Please pass a name on the query string or in the request body");
21 }
22

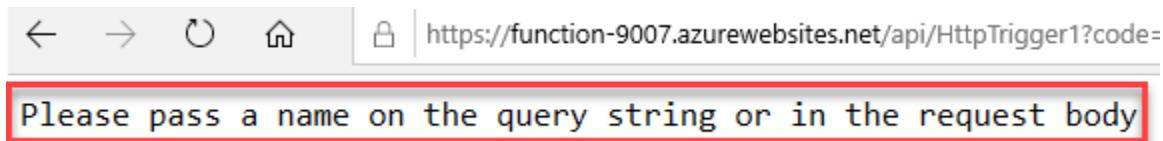
```

Get function URL

Key	URL
default	https://function-9007.azurewebsites.n...

or in the request body");

8. Deschideți o filă nouă de browser și lipiți adresa URL a funcției copiate în bara de adrese a browserului web. Când pagina este solicitată, funcția va rula. Observați mesajul returnat care precizează că funcția necesită un nume în corpul solicitării.



9. Adăugați **&name= yourname** la sfârșitul adresei URL.

Notă : De exemplu, dacă numele dumneavoastră este Cindy, adresa URL finală va fi următoarea: <https://azfuncxxx.azurewebsites.net/api/HttpTrigger1?code=X9xx9999xXXXXX9x9xxXX==&name=cindy>



10. Când apăsați Enter, funcția rulează și fiecare invocare este urmărită. Pentru a vizualiza urmăririle, reveniți la lama Portal **HttpTrigger1 | Code + Test** și faceți clic pe **Monitor**. Puteți **configura** Application Insights selectând marcajul temporal și făcând clic pe **Executare interogare în Application Insights**.

A screenshot of the Azure portal showing the "HttpTrigger1 | Monitor" page. The left sidebar has "Monitor" selected. The main area shows "Invocations" and "Logs" tabs. Under "Invocations", it shows "Success Count: 2" and "Error Count: 0" for the last 30 days. Under "Invocation Traces", it lists two recent traces: "2020-05-15 01:38:54.716" and "2020-05-15 01:36:18.615", both marked as "Success".

Felicitări! Ați creat o aplicație funcțională pentru a afișa un mesaj de salut atunci când există o solicitare HTTP.

Notă : Pentru a evita costuri suplimentare, puteți elimina opțional acest grup de resurse. Căutați grupuri de resurse, faceți clic pe grupul dvs. de resurse, apoi faceți clic pe **Ștergeți grupul de resurse**. Verificați numele grupului de resurse, apoi faceți clic pe **Ștergeți**. Monitorizați **notificările** pentru a vedea cum se desfășoară ștergerea.

