

Microsoft Azure Devops (MS-TFS)

Por: Carlos Carreño
ccarreno@cienciadedatos.es

Octubre, 2020



Unidad 8 Recursos de Desarrollo

- Command line interface
- REST APIs
- DevOps resource center



Command line interface

- Azure DevOps cuenta con una **extensión** para la interfaz de línea de comandos (CLI) de Azure, puede administrar muchos servicios de Azure DevOps desde la línea de comandos.
- Los comandos de la CLI le permiten **optimizar sus tareas** con un marco de trabajo **interactivo más rápido y flexible**, sin pasar por los flujos de trabajo de la interfaz de usuario.



CLI: Instalación del cliente

- Descarga el archivo *.msi desde <https://aka.ms/installazurecliwindows>

Install or update

The MSI distributable is used for installing or updating the Azure CLI on Windows. You don't need to uninstall current versions before using the MSI installer because the MSI will update any existing version.

Microsoft Installer (MSI)

Microsoft Installer (MSI) with Command

When the installer asks if it can make changes to your computer, click the "Yes" box.

Azure CLI current version

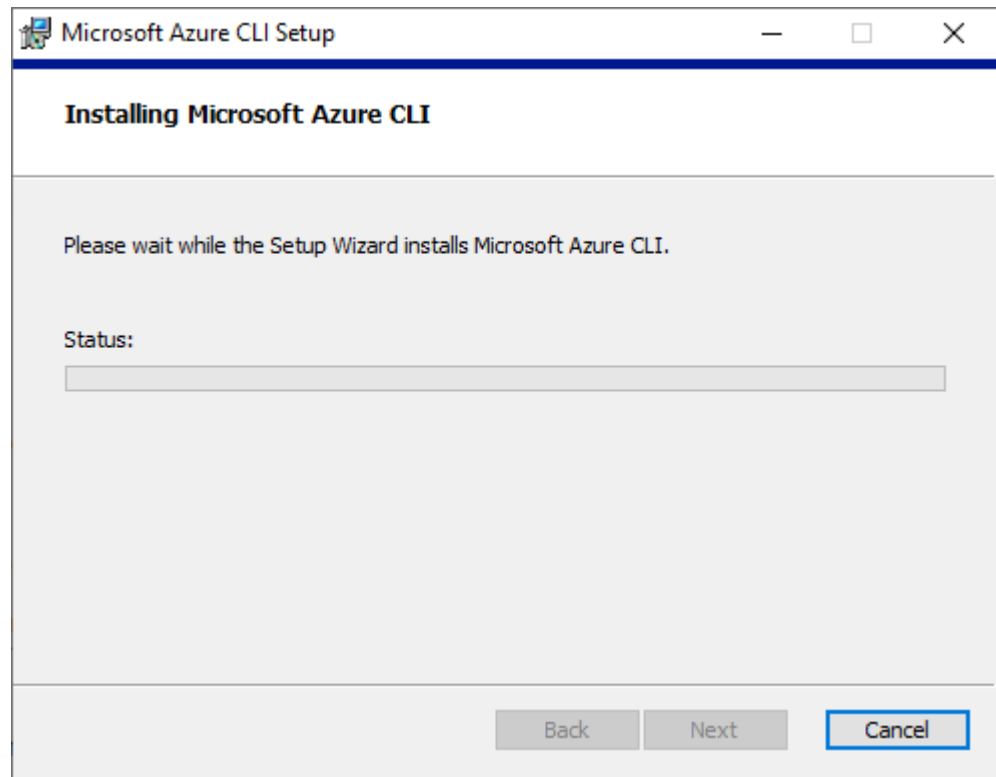
Download and install the current release of the Azure CLI.

Current release of the Azure CLI



...

- Sigue los pasos del asistente de instalación





CLI: Verificando la instalación

- Ejecuta: `az --version`

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18363.1198]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Daddy>az --version
azure-cli                2.15.1
core                     2.15.1
telemetry                1.0.6

Python location 'C:\Program Files (x86)\Microsoft SDKs\Azure\CLI2\python.exe'
Extensions directory 'C:\Users\Daddy\.azure\cliextensions'

Python (Windows) 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 23 2018, 23:31:17) [MSC v.1916 32 bit (Intel)]

Legal docs and information: aka.ms/AzureCliLegal

Your CLI is up-to-date.

Please let us know how we are doing: https://aka.ms/azureclihats
and let us know if you're interested in trying out our newest features: https://aka.ms/CLIUXstudy

C:\Users\Daddy>
```

CLI: Agregando la extensión Azure DevOps



- Ejecuta: `az extension add --name azure-devops`

```
Símbolo del sistema
C:\Users\Daddy>az extension add --name azure-devops
- Installing ..
C:\Users\Daddy>az extension show --name azure-devops
{
  "extensionType": "whl",
  "metadata": {
    "author": "Microsoft",
    "author_email": "VSTS_Social@microsoft.com",
    "azext.minCliCoreVersion": "2.2.0",
    "classifiers": [
      "Development Status :: 4 - Beta",
      "Intended Audience :: Developers",
      "Intended Audience :: System Administrators",
      "Programming Language :: Python",
      "Programming Language :: Python :: 3",
      "Programming Language :: Python :: 3.4",
      "Programming Language :: Python :: 3.5",
      "Programming Language :: Python :: 3.6",
      "License :: OSI Approved :: MIT License"
    ],
    "description": "Microsoft DevOps CLI Extension for Windows, Mac and Linux\n=====
=====\\n\\n1.0.0\\n-----\\n\\n* Initial preview
release.\\n\\n",
    "filename": "C:\\Users\\Daddy\\.azure\\cliextensions\\azure-devops\\azure_devops-0.18.0.dist-
info",
    "home_page": "https://github.com/Microsoft/azure-devops-cli-extension",
    "license": "MIT",
    "metadata_version": "2.0",
    "name": "azure-devops",
    "platforms": [
      "UNKNOWN"
    ],
  },
}
```



CLI: Lista de extensiones

- Mostrar la lista de extensiones instaladas

A screenshot of a Windows Command Prompt window titled "Símbolo del sistema". The window has a black background with white text. The command prompt shows the user's current directory as "C:\Users\Daddy" and the command "az extension list" has been executed. The output is a JSON array containing one object for the "azure-devops" extension. The object includes fields for "experimental" (false), "extensionType" ("whl"), "name" ("azure-devops"), "path" (the local installation path), "preview" (false), and "version" ("0.18.0").

```
C:\Users\Daddy>az extension list
[
  {
    "experimental": false,
    "extensionType": "whl",
    "name": "azure-devops",
    "path": "C:\\Users\\Daddy\\.azure\\cliextensions\\azure-devops",
    "preview": false,
    "version": "0.18.0"
  }
]
C:\Users\Daddy>
```


Login en Azure



- Ingresa a Azure desde CLI

```
Símbolo del sistema
"version": "0.18.0"
}
]
C:\Users\Daddy>az login
The default web browser has been opened at https://login.microsoftonline.com/common/oauth2/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with `az login --use-device-code`.
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "c4b205c7-fa01-4d7f-b88b-ad27eba49c34",
    "id": "4a1690da-7fb2-44d0-8168-c2b54f28205e",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure Redsoft",
    "state": "Warned",
    "tenantId": "c4b205c7-fa01-4d7f-b88b-ad27eba49c34",
    "user": {
```

CLI: Definiendo la Organización y proyecto por defecto



```
Símbolo del sistema
C:\Users\Daddy>az devops configure --defaults organization=https://dev.azure.com/AzureDevopsCognos/ project=CognosApp
C:\Users\Daddy>
```



CLI: Creando un repo

- `az repos create --name "CLISample" --detect true --open --organization "https://dev.azure.com/AzureDevopsCognos" --project "CognosApp"`

```
Símbolo del sistema
C:\Users\Daddy>az repos create --name "CLISample" --detect true --open --organization "https://dev.azure.com/AzureDevopsCognos"
--project "CognosApp"
{
  "defaultBranch": null,
  "id": "979a1926-ee7a-4204-ab45-0dce7cbbc561",
  "isFork": null,
  "name": "CLISample",
  "parentRepository": null,
  "project": {
    "abbreviation": null,
    "defaultTeamImageUrl": null,
    "description": "Aplicacion Demostrativa de Cognos",
    "id": "59725bf6-cf1c-40c4-b4b1-3521671394ec",
    "lastUpdateTime": "2020-11-24T15:23:25.387Z",
    "name": "CognosAPP",
    "revision": 51,
    "state": "wellFormed",
    "url": "https://dev.azure.com/AzureDevopsCognos/_apis/projects/59725bf6-cf1c-40c4-b4b1-3521671394ec",
  }
}
```



CLI: Creando un pipeline

- Crea un pipeline en el repositorio:

`az pipelines create --name "MyCLISample" --description "Pipeline for CLI project" --repository CLISample --branch master --repository-type tfsgit`

```
Símbolo del sistema - az pipelines create --name "MyCLISample" --description "Pipeline for CLI project" --repository CLISample --branch master --repo...
C:\Users\Daddy>az pipelines create --name "MyCLISample" --description "Pipeline for CLI project" --repository CLISample
--branch master --repository-type tfsgit
This command is in preview. It may be changed/removed in a future release.
Which template do you want to use for this pipeline?
[1] Starter pipeline
[2] Android
[3] Ant
[4] ASP.NET
[5] ASP.NET Core
[6] .NET Core Function App to Windows on Azure
[7] ASP.NET Core (.NET Framework)
[8] Deploy to Azure Kubernetes Service
[9] Deploy to Kubernetes - Review app with Azure DevSpaces
[10] Docker
[11] Docker
[12] C/C++ with GCC
[13] Go
[14] Gradle
[15] HTML
[16] Jekyll site
[17] Maven
[18] Maven package Java project Web App to Linux on Azure
[19] .NET Desktop
[20] Node.js
[21] Node.js Express Web App to Linux on Azure
```

REST APIs



- Las API de **Representational State Transfer** (REST) admiten un conjunto de operaciones (métodos) HTTP, que proporcionan acceso para crear, recuperar, actualizar o eliminar a los recursos del servicio. Este artículo lo guía a través de:
 - ☐ Los componentes básicos de un par de solicitud / respuesta de API REST.
 - ☐ Descripción general de la creación y el envío de una solicitud REST y el manejo de la respuesta.



API REST Components

- Un par de solicitud / respuesta de API REST se puede dividir en cinco componentes:

1. The request URI, in the following form: `VERB https://{instance}/{team-project}/{_apis}/{area}/{resource}?api-version={version}`

- *instance*: The Azure DevOps Services organization or TFS server you're sending the request to. They are structured as follows:
 - Azure DevOps Services: `dev.azure.com/{organization}`
 - TFS: `{server:port}/tfs/{collection}` (the default port is 8080, and the value for collection should be `DefaultCollection` but can be any collection)
- *resource path*: The resource path is as follows: `_apis/{area}/{resource}`. For example `_apis/wit/workitems`.
- *api-version*: Every API request should include an api-version to avoid having your app or service break as APIs evolve. api-versions are in the following format: `{major}.{minor}[-{stage}][.{resource-version}]`, for example:
 - `api-version=1.0`
 - `api-version=1.2-preview`
 - `api-version=2.0-preview.1`



...

2. HTTP request message header fields:

- A required **HTTP method** (also known as an operation or verb), which tells the service what type of operation you are requesting. Azure REST APIs support GET, HEAD, PUT, POST, and PATCH methods.
- Optional additional header fields, as required by the specified URI and HTTP method. For example, an **Authorization** header that provides a bearer token containing client authorization information for the request.

3. Optional HTTP request message body fields, to support the URI and HTTP operation. For example, POST operations contain MIME-encoded objects that are passed as complex parameters.

- For POST or PUT operations, the MIME-encoding type for the body should be specified in the **Content-type** request header as well. Some services require you to use a specific MIME type, such as `application/json`.



...

4. HTTP response message header fields:

- An [HTTP status code](#), ranging from 2xx success codes to 4xx or 5xx error codes. Alternatively, a service-defined status code may be returned, as indicated in the API documentation.
- Optional additional header fields, as required to support the request's response, such as a `Content-type` response header.

5. Optional HTTP response message body fields:

- MIME-encoded response objects may be returned in the HTTP response body, such as a response from a GET method that is returning data. Typically, these objects are returned in a structured format such as JSON or XML, as indicated by the `Content-type` response header. For example, when you request an access token from Azure AD, it will be returned in the response body as the `access_token` element, one of several name/value paired objects in a data collection. In this example, a response header of `Content-Type: application/json` is also included.

Ejemplo de uso de API REST en C#

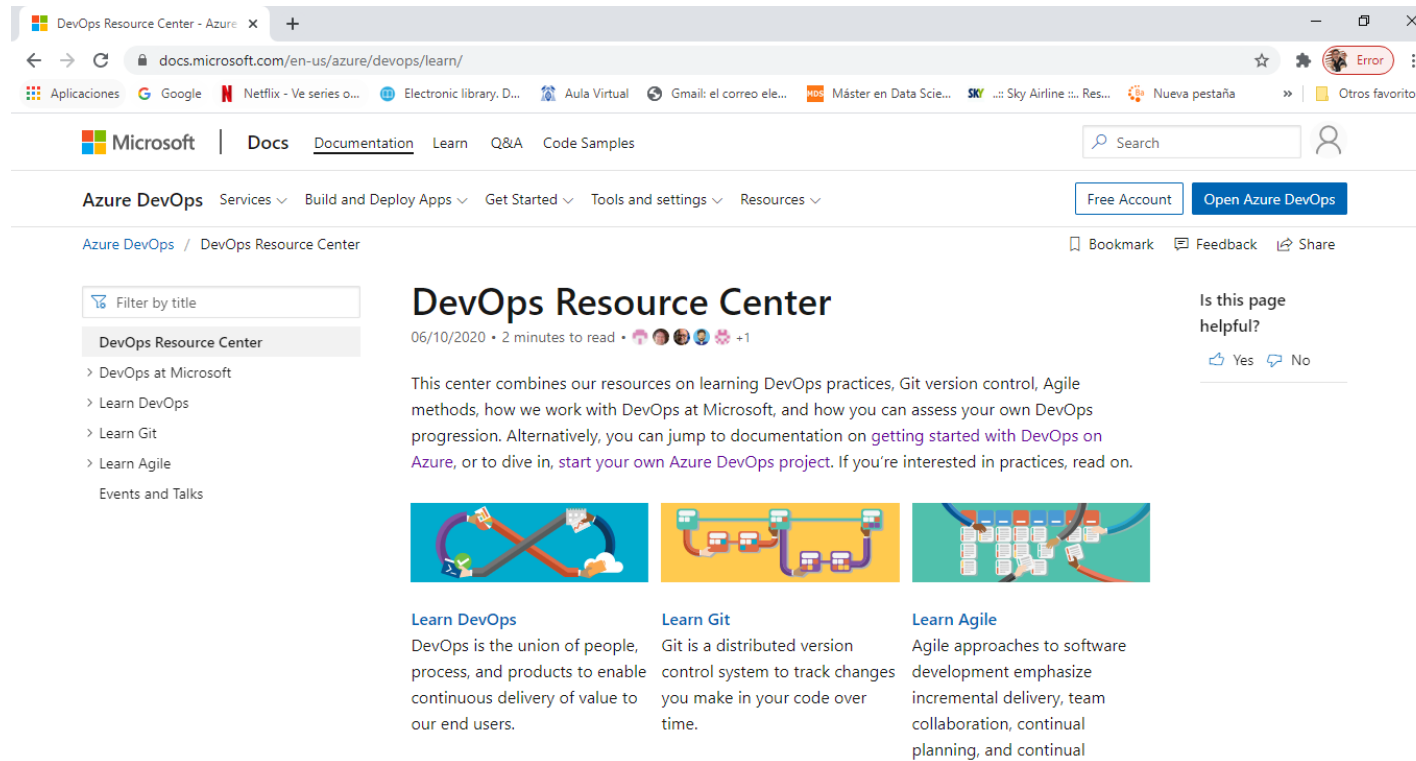


```
public static async void GetProjects() {  
    try {  
        var personalaccesstoken = "PAT_FROM_WEBSITE";  
        using (HttpClient client = new HttpClient()){  
            client.DefaultRequestHeaders.Accept.Add(  
                new System.Net.Http.Headers.MediaTypeWithQualityHeaderValue("application/json"));  
            client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Basic",  
                Convert.ToBase64String(  
                    System.Text.ASCIIEncoding.ASCII.GetBytes(  
                        string.Format("{0}:{1}", "", personalaccesstoken))));  
            using (HttpResponseMessage response = await client.GetAsync(  
                "https://dev.azure.com/{organization}/_apis/projects")) {  
                response.EnsureSuccessStatusCode();  
                string responseBody = await response.Content.ReadAsStringAsync();  
                Console.WriteLine(responseBody);  
            }  
        }  
    }  
    catch (Exception ex) {  
        Console.WriteLine(ex.ToString());  
    }  
}
```

DevOps resource center



- Centro de Recursos de Devops



The screenshot shows the DevOps Resource Center page in a web browser. The browser's address bar shows the URL `docs.microsoft.com/en-us/azure/devops/learn/`. The page has a Microsoft header with navigation links for Docs, Documentation, Learn, Q&A, and Code Samples. Below the header, there are buttons for 'Free Account' and 'Open Azure DevOps'. The main content area is titled 'DevOps Resource Center' and includes a sub-header '06/10/2020 • 2 minutes to read'. The page describes the center's purpose and provides links to 'getting started with DevOps on Azure' and 'start your own Azure DevOps project'. There are three featured articles: 'Learn DevOps', 'Learn Git', and 'Learn Agile', each with a brief description and an illustration. A sidebar on the left allows filtering by title, and a 'Feedback' section is on the right.

Microsoft | Docs | Documentation | Learn | Q&A | Code Samples

Azure DevOps | Services | Build and Deploy Apps | Get Started | Tools and settings | Resources

Free Account | Open Azure DevOps

DevOps Resource Center

06/10/2020 • 2 minutes to read

This center combines our resources on learning DevOps practices, Git version control, Agile methods, how we work with DevOps at Microsoft, and how you can assess your own DevOps progression. Alternatively, you can jump to documentation on [getting started with DevOps on Azure](#), or to dive in, [start your own Azure DevOps project](#). If you're interested in practices, read on.

Learn DevOps
DevOps is the union of people, process, and products to enable continuous delivery of value to our end users.

Learn Git
Git is a distributed version control system to track changes you make in your code over time.

Learn Agile
Agile approaches to software development emphasize incremental delivery, team collaboration, continual planning, and continual

Laboratorio



- Lab 12 Automating Selenium Tests in Azure Pipelines



Referencias

- <https://docs.microsoft.com/en-us/cli/azure/>
- <https://docs.microsoft.com/en-us/azure/devops/learn/>
- <https://docs.microsoft.com/en-us/rest/api/azure/devops/?view=azure-devops-rest-6.1&viewFallbackFrom=azure-devops>