Fetch work items with queries programmatically

Article • 03/21/2025

Azure DevOps Services

Fetching work items using queries is a common scenario in Azure DevOps Services. This article explains how to implement this scenario programmatically using REST APIs or .NET client libraries.

Prerequisites

Expand table

Category	Requirements
Azure DevOps	- An organization .<\br>- A Personal Access Token (PAT).
Development environment	A C# development environment. You can use Visual Studio .

> **Important**
>
> We use Personal Access Tokens (PATs) as an example in this article, but we don't recommend using PATs. For more secure authentication mechanisms, see **Authentication guidance**.

Create a C# project in Visual Studio

For information about C# programming within Visual Studio, see the Visual Studio C# programming documentation.

C# code content

The following tasks occur in the code snippet:

- **Authenticate**

    1. Create credentials using your Personal Access Token (PAT).
    2. Generate the client using the credentials.

- **Get the work items**

    1. Create the query you want to use.
    2. Retrieve the results for that query.
    3. Fetch each of the work items by ID.

C# code snippet

cs

```cs
// nuget:Microsoft.TeamFoundationServer.Client
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.TeamFoundation.WorkItemTracking.WebApi;
using Microsoft.TeamFoundation.WorkItemTracking.WebApi.Models;
using Microsoft.VisualStudio.Services.Common;
using Microsoft.VisualStudio.Services.WebApi;

public class QueryExecutor
{
    private readonly Uri uri;
    private readonly string personalAccessToken;

    /// <summary>
    /// Initializes a new instance of the <see cref="QueryExecutor" /> class.
    /// </summary>
    /// <param name="orgName">
    /// An organization in Azure DevOps Services. If you don't have one, you can cre-
ate one for free:
    /// <see href="https://go.microsoft.com/fwlink/?LinkId=307137" />.
    /// </param>
    /// <param name="personalAccessToken">
    /// A Personal Access Token, find out how to create one:
    /// <see href="/azure/devops/organizations/accounts/use-personal-access-tokens-
to-authenticate?view=azure-devops" />.
    /// </param>
    public QueryExecutor(string orgName, string personalAccessToken)
    {
        this.uri = new Uri("https://dev.azure.com/" + orgName);
        this.personalAccessToken = personalAccessToken;
    }

    /// <summary>
    /// Execute a WIQL (Work Item Query Language) query to return a list of open
bugs.
    /// </summary>
```

```
    /// <param name="project">The name of your project within your organization.
</param>
    /// <returns>A list of <see cref="WorkItem"/> objects representing all the open
bugs.</returns>
    public async Task<IList<WorkItem>> QueryOpenBugs(string project)
    {
        var credentials = new VssBasicCredential(string.Empty,
this.personalAccessToken);
        var wiql = new Wiql()
        {
            Query = "Select [Id] " +
                    "From WorkItems " +
                    "Where [Work Item Type] = 'Bug' " +
                    "And [System.TeamProject] = '" + project + "' " +
                    "And [System.State] <> 'Closed' " +
                    "Order By [State] Asc, [Changed Date] Desc",
        };

        using (var httpClient = new WorkItemTrackingHttpClient(this.uri, new
VssCredentials(credentials)))
        {
            try
            {
                var result = await
httpClient.QueryByWiqlAsync(wiql).ConfigureAwait(false);
                var ids = result.WorkItems.Select(item => item.Id).ToArray();

                if (ids.Length == 0)
                {
                    return Array.Empty<WorkItem>();
                }

                var fields = new[] { "System.Id", "System.Title", "System.State" };
                return await httpClient.GetWorkItemsAsync(ids, fields,
result.AsOf).ConfigureAwait(false);
            }
            catch (Exception ex)
            {
                Console.WriteLine("Error querying work items: " + ex.Message);
                return Array.Empty<WorkItem>();
            }
        }
    }

    /// <summary>
    /// Execute a WIQL (Work Item Query Language) query to print a list of open bugs.
    /// </summary>
    /// <param name="project">The name of your project within your organization.
</param>
    /// <returns>An async task.</returns>
    public async Task PrintOpenBugsAsync(string project)
```

```
    {
        var workItems = await this.QueryOpenBugs(project).ConfigureAwait(false);
        Console.WriteLine("Query Results: {0} items found", workItems.Count);

        foreach (var workItem in workItems)
        {
            Console.WriteLine(
                "{0}\t{1}\t{2}",
                workItem.Id,
                workItem.Fields["System.Title"],
                workItem.Fields["System.State"]);
        }
    }
}
```

Related articles

- Create a bug
- Integrate samples

ⓘ **Note:** The author created this article with assistance from AI. Learn more