Get started with the REST APIs

Article • 04/04/2025

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Integrate your application with Azure DevOps using the REST APIs provided in this article. These APIs allow you to interact with the services programmatically, enabling you to automate workflows, integrate with other systems, and extend the capabilities of Azure DevOps.

The APIs follow a common pattern, as shown in the following example:

```
no-highlight

VERB https://{instance}/{collection}/{team-project}/_apis/{area}/{resource}?api-version={version}
```

> **Tip**
>
> As APIs evolve, we recommend that you include an API version in every request. This practice can help you avoid unexpected changes in the API that could break.

Azure DevOps Services

For Azure DevOps Services, `instance` is `dev.azure.com/{organization}` and `collection` is `DefaultCollection`, so the pattern looks like the following example:

```
no-highlight

VERB https://dev.azure.com/{organization}/_apis/{area}/{resource}?api-version={version}
```

The following example shows how to get a list of projects in an organization:

```
dos

curl -u {username}:{personalaccesstoken} https://dev.azure.com/{organization}/_apis/projects?api-version=2.0
```

If you want to provide the personal access token (PAT) through an HTTP header, first prepend a colon to the PAT. Then, convert the concatenation of the colon and the PAT to a Base64 string. The following example shows how to convert to Base64 using C#. The resulting string can then be provided as an HTTP header in the format:

```
Authorization: Basic BASE64COLONANDPATSTRING
```

> **Note**
>
> Include the colon before the PAT to avoid authentication errors.

The following example shows C# using the HttpClient class:

```cs
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 = client.GetAsync(

"https://dev.azure.com/{organization}/_apis/projects").Result)
            {
                response.EnsureSuccessStatusCode();
                string responseBody = await response.Content.ReadAsStringAsync();
                Console.WriteLine(responseBody);
            }
        }
    }
    catch (Exception ex)
```

```
    {
        Console.WriteLine(ex.ToString());
    }
}
```

> **Important**
>
> While we use personal access tokens (PATs) in many examples for simplicity, we don't recomment using them for production applications. Instead, consider using more secure authentication mechanisms. For more information, see Authentication guidance.

Azure DevOps Server

For Azure DevOps Server, `instance` is `{server:port}`. The default port for a non-SSL connection is 8080.

The default collection is `DefaultCollection`, but you can use any collection.

Here's how to get a list of projects from Azure DevOps Server using the default port and collection across SSL:

dos

```
curl -u {username}:{personalaccesstoken}
https://{server}/DefaultCollection/_apis/projects?api-version=2.0
```

To get the same list across a non-SSL connection:

dos

```
curl -u {username}:{personalaccesstoken}
http://{server}:8080/DefaultCollection/_apis/projects?api-version=2.0
```

These examples use PATs, which require that you create a PAT.

Responses

You should get a response like the following example:

JSON


```
{
    "value": [
        {
            "id": "00000000-0000-0000-0000-000000000000",
            "name": "Fabrikam-Fiber-TFVC",
            "url": "https: //dev.azure.com/fabrikam-fiber-
inc/_apis/projects/00000000-0000-0000-0000-000000000000",
            "description": "TeamFoundationVersionControlprojects",
            "collection": {
                "id": "00000000-0000-0000-0000-000000000000",
                "name": "DefaultCollection",
                "url": "https: //dev.azure.com/fabrikam-fiber-
inc/_apis/projectCollections/00000000-0000-0000-0000-000000000000",
                "collectionUrl": "https: //dev.azure.com/fabrikam-fiber-inc"
            },
            "defaultTeam": {
                "id": "00000000-0000-0000-0000-000000000000",
                "name": "Fabrikam-Fiber-TFVCTeam",
                "url": "https: //dev.azure.com/fabrikam-fiber-
inc/_apis/projects/00000000-0000-0000-0000-000000000000/teams/00000000-0000-0000-
0000-000000000000"
            }
        },
        {
            "id": "00000000-0000-0000-0000-000000000000",
            "name": "Fabrikam-Fiber-Git",
            "url": "https: //dev.azure.com/fabrikam-fiber-
inc/_apis/projects/00000000-0000-0000-0000-000000000000",
            "description": "Gitprojects",
            "collection": {
                "id": "00000000-0000-0000-0000-000000000000",
                "name": "DefaultCollection",
                "url": "https: //dev.azure.com/fabrikam-fiber-
inc/_apis/projectCollections/00000000-0000-0000-0000-000000000000",
                "collectionUrl": "https: //dev.azure.com/fabrikam-fiber-inc"
            },
            "defaultTeam": {
                "id": "00000000-0000-0000-0000-000000000000",
                "name": "Fabrikam-Fiber-GitTeam",
                "url": "https: //dev.azure.com/fabrikam-fiber-
inc/_apis/projects/00000000-0000-0000-0000-000000000000/teams/00000000-0000-0000-
0000-000000000000"
            }
        }
    ],
    "count": 2
}
```

The response is JSON , which is generally what you get back from the REST APIs, although there are a few exceptions, like Git blobs.

Now, you can look around the specific API areas like work item tracking or Git and get to the resources that you need. Keep reading to learn more about the general patterns that are used in these APIs.

HTTP verbs

Expand table

Verb	Used for...
GET	Get a resource or list of resources
POST	Create a resource, Get a list of resources using a more advanced query
PUT	Create a resource if it doesn't exist or, if it does, update it
PATCH	Update a resource
DELETE	Delete a resource

Request headers and request content

When you provide request body (usually with the POST, PUT and PATCH verbs), include request headers that describe the body. For example,

no-highlight

```
POST https://dev.azure.com/fabrikam-fiber-inc/_apis/build-release/requests
```

HTTP

```
Content-Type: application/json
```

JSON

```
{
   "definition": {
      "id": 3
```

```
    },
    "reason": "Manual",
    "priority": "Normal"
}
```

HTTP method override

Some web proxies might only support the HTTP verbs GET and POST, but not more modern HTTP verbs like PATCH and DELETE. If your calls might pass through one of these proxies, you can send the actual verb using a POST method, with a header to override the method. For example, you might want to update a work item (`PATCH _apis/wit/workitems/3`), but you might have to go through a proxy that only allows GET or POST. You can pass the proper verb (PATCH in this case) as an HTTP request header parameter and use POST as the actual HTTP method.

no-highlight

```
POST https://dev.azure.com/fabrikam-fiber-inc/_apis/wit/workitems/3
```

HTTP

```
X-HTTP-Method-Override: PATCH
```

JSON

```
{
   (PATCH request body)
}
```

Response codes

Expand table

Response	Notes
200	Success, and there's a response body.
201	Success, when creating resources. Some APIs return 200 when successfully creating a resource. Look at the docs for the API you're using to be sure.

Response	Notes
204	Success, and there's no response body. For example, you get this response when you delete a resource.
400	The parameters in the URL or in the request body aren't valid.
401	Authentication failed. Often, this response is because of a missing or malformed Authorization header.
403	The authenticated user doesn't have permission to do the operation.
404	The resource doesn't exist, or the authenticated user doesn't have permission to see that it exists.
409	There's a conflict between the request and the state of the data on the server. For example, if you attempt to submit a pull request and there's already a pull request for the commits, the response code is 409.

Cross-origin resource sharing (CORS)

Azure DevOps Services supports CORS, which enables JavaScript code served from a domain other than `dev.azure.com/*` to make Ajax requests to Azure DevOps Services REST APIs. Each request must provide credentials (PATs and OAuth access tokens are both supported options). Example:

JavaScript

```
$( document ).ready(function() {
    $.ajax({
        url: 'https://dev.azure.com/fabrikam/_apis/projects?api-version=1.0',
        dataType: 'json',
        headers: {
            'Authorization': 'Basic ' + btoa("" + ":" + myPatToken)
        }
    }).done(function( results ) {
        console.log( results.value[0].id + " " + results.value[0].name );
    });
});
```

Replace `myPatToken` with a PAT.

Versioning

Azure DevOps REST APIs are versioned to ensure applications and services continue to work as APIs evolve.

Guidelines

- Specify the API version with every request (**required**).
- Format API versions as follows: {major}.{minor}-{stage}.{resource-version}. For example, `1.0`, `1.1`, `1.2-preview`, `2.0`.
- Specify a particular revision of the API when it's in preview, by using the following version format: `1.0-preview.1`, `1.0-preview.2`. Once an API is released (1.0, for example), its preview version (1.0-preview) is deprecated and can be deactivated after 12 weeks.
- Upgrade to the released version of the API. Once a preview API is deactivated, requests that specify `-preview` version get rejected.

Usage

Specify the API version in the header of the HTTP request or as a URL query parameter.

HTTP request header:

```HTTP
Accept: application/json;api-version=1.0
```

Query parameter:

```no-highlight
GET https://dev.azure.com/{organization}/_apis/{area}/{resource}?api-version=1.0
```

Supported versions

For information on supported versions, see REST API versioning, Supported versions.