

ADO PortGate DevOps

Installation & Setup Guide

From Marketplace to First Admin User

solutionade.net | Azure DevOps Extension

Version 1.0 | March 2026

Prerequisites

Before installing ADO PortGate DevOps, make sure the following requirements are met:

1. Azure DevOps Organization Admin Access

You must be an Organization Administrator in Azure DevOps to install extensions from the Marketplace.

Note: If you are not an Organization Admin, contact your Azure DevOps admin. You can verify your role at: Organization Settings → Users.

2. Supported Browsers

- Google Chrome (recommended, latest version)
- Microsoft Edge (latest version)
- Mozilla Firefox (latest version)
- Safari (latest version, macOS only)

3. Azure DevOps Organization

You need an active Azure DevOps organization with at least one project configured and work item types enabled (Bugs, Tasks, Features).

Step 1 — Install from the Visual Studio Marketplace

STEP 1

Open the Visual Studio Marketplace

Navigate to marketplace.visualstudio.com and sign in with the same Microsoft account used for your Azure DevOps organization.

Follow these steps to find and install ADO PortGate:

1. Go to marketplace.visualstudio.com
2. Click the Azure DevOps tab at the top.
3. In the search box, type: ADO PortGate and press Enter.
4. Click the ADO PortGate DevOps result (published by solutionade.net).
5. Click Get it free on the extension listing page.

Review Extension Details & Permissions

Before installing, Azure DevOps will show you a permissions review screen. ADO PortGate requires the following permission scopes:

Permission	Reason
Code (read, write, manage)	Required to interact with repository content
Graph (manage)	Required for user and group management
Identity (manage)	Required for authentication and user identity
Project and team (read, write, manage)	Required to access and manage DevOps projects
Security (manage)	Required to configure access control
Work items (read)	Required to read work item data

⚠ Note: These are high-privilege scopes. Azure DevOps will show a warning in orange. Only proceed if you trust the publisher and the extension source code.

6. Select your Azure DevOps organization from the dropdown (e.g., marencemilanova).
7. Click Install to complete the installation.
8. Wait for the confirmation screen showing Done.

✓ Tip: After installation, navigate to your Azure DevOps organization. You should see ADO PortGate in the left navigation sidebar inside your project.

Step 2 — Create a Personal Access Token (PAT)

STEP 2

Create a PAT with Full Permissions

ADO PortGate uses a Personal Access Token to authenticate to Azure DevOps on behalf of users. This token is entered during tenant configuration.

Follow these steps to create the PAT:

9. In Azure DevOps, click your profile picture (top right) and select Personal Access Tokens.
10. Click + New Token.
11. Fill in the token details:

Field	Value
Name	ADO PortGate Admin Token (or any descriptive name)
Organization	Select your organization (e.g., marencemilanova)
Expiration	Set to 1 year (or as per your security policy)
Scopes	Select Full access (recommended for initial setup)

⚠ Note: For production environments, you may use custom scopes: Work Items (Read & Write), Project and Team (Read, Write & Manage), Identity (Read & Manage), Graph (Read & Manage), Security (Manage).

12. Click Create.
13. **IMPORTANT:** Copy the generated token immediately and store it securely. Azure DevOps will NOT show it again after you close this dialog.

✓ Tip: Store the PAT in a password manager or secure vault. You will need it for Step 3 (Tenant Configuration).

Step 3 — Configure the Tenant (First-Time Setup)

STEP 3

Populate Azure DevOps Settings

After installing the extension, open ADO PortGate inside Azure DevOps and configure your tenant with the organization name and PAT you created.

How to access Tenant Configuration:

14. Open your Azure DevOps project.
15. In the left sidebar, click on ADO PortGate (the icon with the blue/orange circle).
16. The ADO PortGate portal will open. A Tenant Configuration dialog will appear automatically on first launch.
17. If the dialog does not appear automatically, look for a Settings or Admin menu inside the portal and click Tenant Configuration.

Fill In the Tenant Configuration Form

Complete the following fields in the Azure DevOps tab of the Tenant Configuration dialog:

Field	What to Enter
Organization Name	Your Azure DevOps organization name (e.g., marencemilanova)
Personal Access Token	Paste the PAT you created in Step 2

⚠ Note: The Organization Name must exactly match the name in your Azure DevOps URL (after dev.azure.com/). Check for typos — the name is case-sensitive.

18. Once all fields are filled in, click Save Configuration.
19. The dialog will close and ADO PortGate is now connected to your Azure DevOps organization.

✓ Tip: You can return to Tenant Configuration at any time via the Admin menu to update your PAT if it expires.

Step 4 — Register the First Admin User

STEP 4

Create the Administrator Account

The first person to set up ADO PortGate should register as an Admin user. This account will manage all users, project assignments, and system settings.

Registering a New Account

If you have not yet created an account in ADO PortGate:

20. On the ADO PortGate login page, click Sign Up.
21. Fill in the registration form:

Field	Description
Name	Your first name
Surname	Your last name
Email	Your email address (used for login)
Password	A strong password for your account

22. Click Register (or Submit).
23. You will be redirected to the login page.

Assigning the Admin Role

After registration, the account starts with a basic User role. You must assign the Admin role:

⚠ Note: If you are the very first user and no Admin exists yet, ADO PortGate may automatically grant Admin rights to the first registered user. If not, you may need to assign the role directly in the database or through the initial setup screen.

24. Log in to ADO PortGate with your registered account.
25. Navigate to User Management (visible in the Admin menu).
26. Find your own account in the user list.
27. Click Edit and change the Role from User to Admin.
28. Save the changes.

✓ Tip: Once you have Admin access, you can manage all other users, assign projects, and configure system settings from the Admin menu.

Step 5 — Manage Users and Assign Projects

STEP 5

Set Up Users and Project Access

As the Admin, you control which users can access which Azure DevOps projects. External users cannot access projects until an Admin explicitly assigns them.

Adding Users

Users can self-register through the Sign Up page, or an Admin can add them directly:

29. Go to the Admin menu and select User Management.
30. Click Add User.
31. Fill in Name, Surname, Email, and assign a Role (Admin or User).
32. Save the new user.

Assigning Azure DevOps Projects to a User

After a user is registered, you must assign them to one or more Azure DevOps projects before they can create work items:

33. In User Management, find the user and click their User Projects (folder/settings icon).
34. The User Projects modal will open. Click New Project.
35. Fill in the project details:

Field	Description
Organization URL	Full URL of your Azure DevOps org: <code>https://dev.azure.com/{org-name}</code>
Token	A PAT with Work Items Read/Write permissions for this project

36. Click Save. The project now appears in the user's project list.
37. Repeat for additional projects if needed.

⚠ Note: Each project assignment requires its own PAT. You can reuse the same PAT across multiple users if it has the necessary permissions.

Step 6 — User Workflow (Creating Work Items)

STEP 6

External Users: Submit and Track Work Items

Once a user has been assigned projects, they can log in, create work items (Bugs, Tasks, Features), and track their status in real-time.

Login and Home Page

Users log in with their email and password. After login, they land on the Home page which shows:

- A project dropdown to select their assigned project
- A work item status distribution chart (New, Active, Resolved, Closed)
- A work item type distribution chart (Bug, Task, Feature)
- A searchable work item list with ID, Title, Type, Assigned To, and Status

Creating a Work Item

38. Click Add Work Item on the Home page.
39. Select the Issue Type from the dropdown: Bug, Task, or Feature.
40. Fill in the form fields:

Issue Type	Fields Required
Bug	Severity, Acceptance Criteria, Title, Description
Task	Title, Description
Feature	Title, Description

41. Click Create Work Item.
42. The system creates the work item directly in Azure DevOps and shows a confirmation with the assigned work item ID.

✓ **Tip:** The work item ID is clickable and links directly to the item in Azure DevOps for team members who have ADO access.

Best Practices & Security Tips

For Administrators

- **Regularly rotate Personal Access Tokens (every 3-6 months) and update them in Tenant Configuration and all user project assignments.** PAT Rotation:
Least Privilege:
- **Only assign users to projects they actually need. Remove access when a user leaves or changes role.**
- **Use the Logging settings in Tenant Configuration to enable activity logs and monitor work item creation.** Monitor Activity:
- **Document your tenant settings, organization name, and project assignments in a secure internal wiki.** Backup Configuration:
- **Ensure ADO PortGate is only accessible over HTTPS to protect PATs and user credentials in transit.** HTTPS Only:

For Users

- **For bugs, always include steps to reproduce, expected behavior vs actual behavior, and a severity level.** Clear Bug Reports:
- **Use Bug for defects, Task for to-do items, and Feature for enhancement requests.** Right Issue Type:
- **Check the work item list regularly to see status updates from the development team.** Track Your Items:
- **Change your password regularly via Profile Settings.** Profile Security:

Appendix — Quick Reference

Setup Checklist

- Organization Admin access confirmed in Azure DevOps
- Extension installed from Visual Studio Marketplace
- PAT created with Full Access (or required scopes)
- Tenant Configuration completed with Organization Name, PAT, and Base URL
- First Admin user registered and Admin role assigned
- User Management populated with team members and roles
- Projects assigned to each user via User Projects modal

Glossary

Term	Definition
PAT	Personal Access Token — secure credential for Azure DevOps API access. Treat like a password.
Tenant	The organizational configuration connecting ADO PortGate to your Azure DevOps organization.
Work Item	A trackable unit of work in Azure DevOps: Bug, Task, or Feature.
Admin Role	Full access to configure tenant, manage users, and assign projects.
User Role	Limited access — can only view assigned projects and create/track work items.
JWT	JSON Web Token — used internally for session management and secure authentication.