



Announcing the Azure DevOps Integration for Exalate

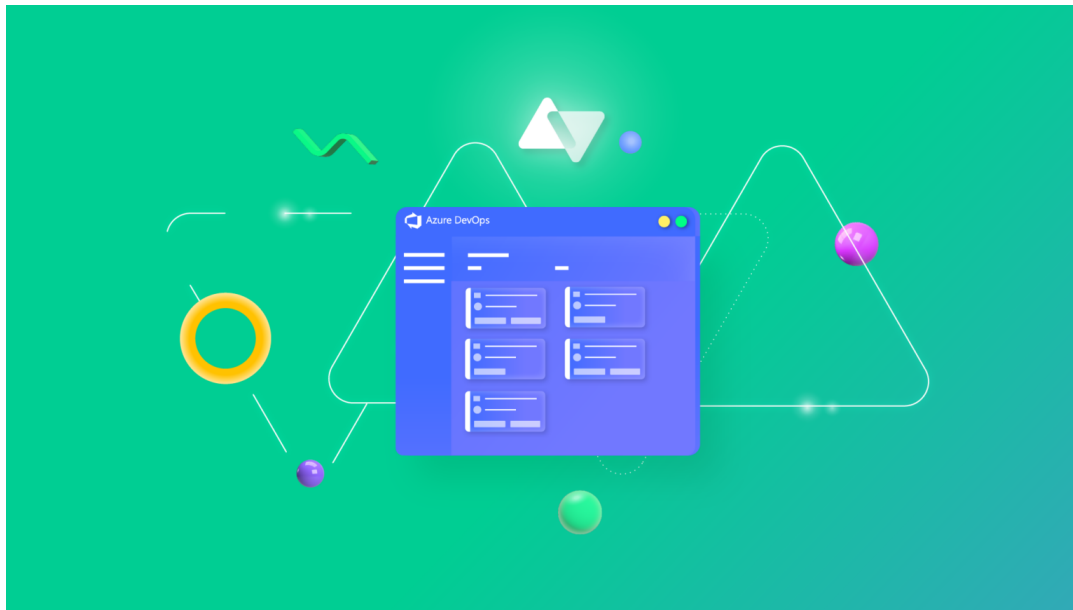


Table of contents

What is Azure DevOps?

The Azure DevOps-Exalate Integration

Use Case For the Azure DevOps-Exalate Plugin

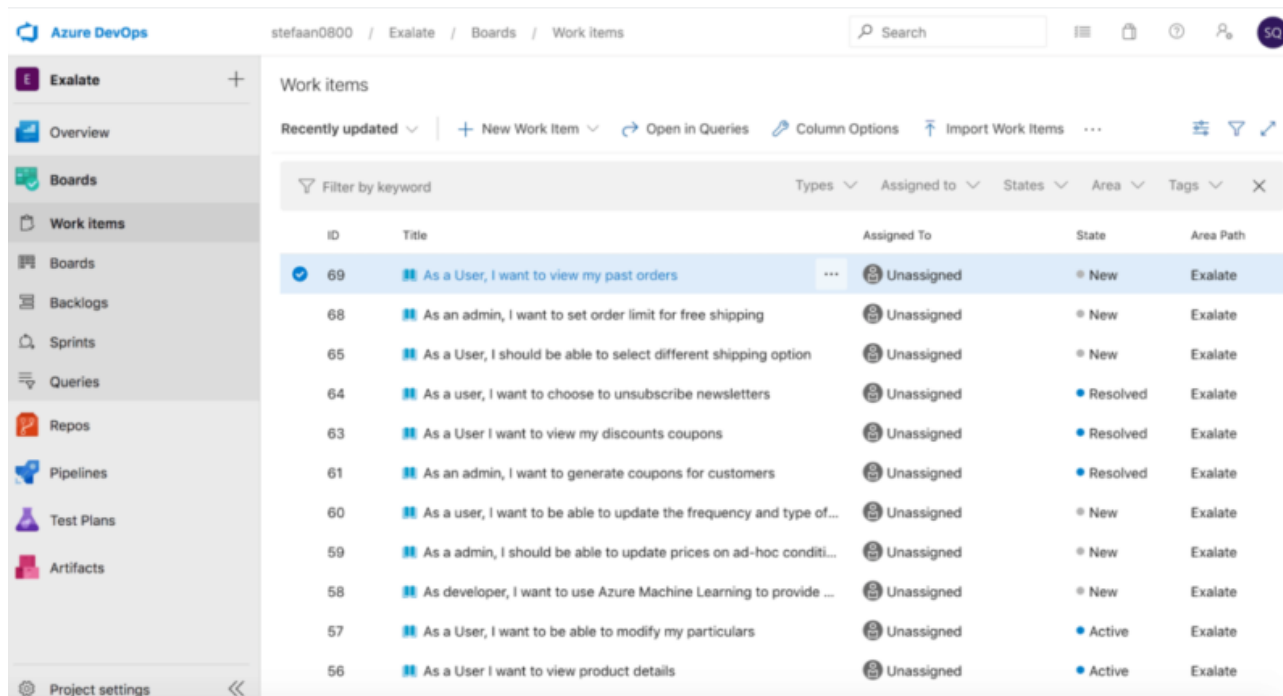
Wrapping it Up



When using Exalate, your teams already expect to have an easy experience collaborating across different trackers. They no longer have to use email or call just to get a status update. They no longer have to work in another system they are not familiar with. You can set up a synchronization and things will just "work".

You can already use Exalate for popular trackers like Jira, ServiceNow, Zendesk, Github and more. And, today we're announcing a new integration: [Azure DevOps!](#)

What is Azure DevOps?



The screenshot displays the Azure DevOps interface for a project named 'Exalate'. The left sidebar shows the navigation menu with 'Work items' selected. The main area shows a list of work items with columns for ID, Title, Assigned To, State, and Area Path. The items are filtered by keyword and sorted by 'Recently updated'.

ID	Title	Assigned To	State	Area Path
69	As a User, I want to view my past orders	Unassigned	New	Exalate
68	As an admin, I want to set order limit for free shipping	Unassigned	New	Exalate
65	As a User, I should be able to select different shipping option	Unassigned	New	Exalate
64	As a user, I want to choose to unsubscribe newsletters	Unassigned	Resolved	Exalate
63	As a User I want to view my discounts coupons	Unassigned	Resolved	Exalate
61	As an admin, I want to generate coupons for customers	Unassigned	Resolved	Exalate
60	As a user, I want to be able to update the frequency and type of...	Unassigned	New	Exalate
59	As a admin, I should be able to update prices on ad-hoc condi...	Unassigned	New	Exalate
58	As developer, I want to use Azure Machine Learning to provide ...	Unassigned	New	Exalate
57	As a User, I want to be able to modify my particulars	Unassigned	Active	Exalate
56	As a User I want to view product details	Unassigned	Active	Exalate

Azure DevOps is an agile planning and tracking tool that provides developer services to support teams to plan work, collaborate on code development, and build and deploy applications. You can track the features you're developing, code defects, and bugs using **work items**. Work items are defined in an area in Azure DevOps, called Boards, and they are similar to **Jira Issues**. Boards provide the possibility to plan, track, and discuss work across your teams.

The Azure DevOps-Exalate Integration

If you're already using both platforms and are looking for a way to build a bi-directional data transfer between the two, you'll find this new [Exalate integration](#) very useful.

Exalate enables you to synchronize your Azure DevOps work items with Jira Issues. The integration makes it all easier for both sides to track issues/ work items, and collaborate with each other while they are working in their own environment. So, an issue created in Jira can easily be assigned to somebody in Azure DevOps as a work item, and Azure DevOps users will try to close it on their own platform. It is also true for the other side, Jira users track and resolve work items that are imported into Jira as issues.

The users on both sides might also need to share an attachment with their teammates or the other end. And they'd prefer not to use work channels or email. In this case, they can just have this attachment synchronized in their work item/ issue. So they will not even have to leave their environment. With Exalate, they'll have the flexibility to decide when, where, and with whom they want to share that particular piece of information.

Use Case For the Azure DevOps-Exalate Plugin

Let me walk you through a possible use case of the Exalate integration for Azure DevOps. Imagine you have a development team working on an Azure DevOps project, while another Support team is communicating in a Jira Service Desk environment. Exalate allows the client to create an issue in Jira, and then a support engineer can synchronize it with the development team.

On one end, you have the customer service desk receiving tickets. And on the other end, you're receiving work items from Azure DevOps tools. Traditionally, customer service agents would receive a ticket and email the information to your developer team. From there, the dev team would need to manually create an issue in Azure DevOps, adding this to their backlog.

However, the developers are too busy to put in data manually. So, you can set up a connection with Exalate within your environment, add the necessary triggers, and they will automatically detect new work items coming in from your customer service desk (as issues). From this point on, you no longer need to manually put the work items into your dev team's Azure DevOps project.

As a result, once the dev team updates a bit of information on their Azure DevOps instance, this information will be synchronized with the other side. The synchronization is automatic and two-way. This means your customer service desk will automatically be aware of any updates on the related issue, and that streamlines communication with the customers.

Wrapping it Up

If you already feel like you're drowning in Azure DevOps work items, then Azure DevOps-Exalate integration will work like a charm for you.

This solution will help you simplify your customer service pipeline with:

- Automatic, two-way synchronization
- Customizable triggers
- Multiple platform support (Jira, GitHub, HP QC/ALM, and more very soon)
- Heroic support from our team