


# Create an Analytics widget for Azure DevOps

04/04/2019 • 2 minutos para ler • Colaboradores 

## Neste artigo

[Prerequisites](#)

[Provide a configuration view](#)

[Render data within a Widget](#)

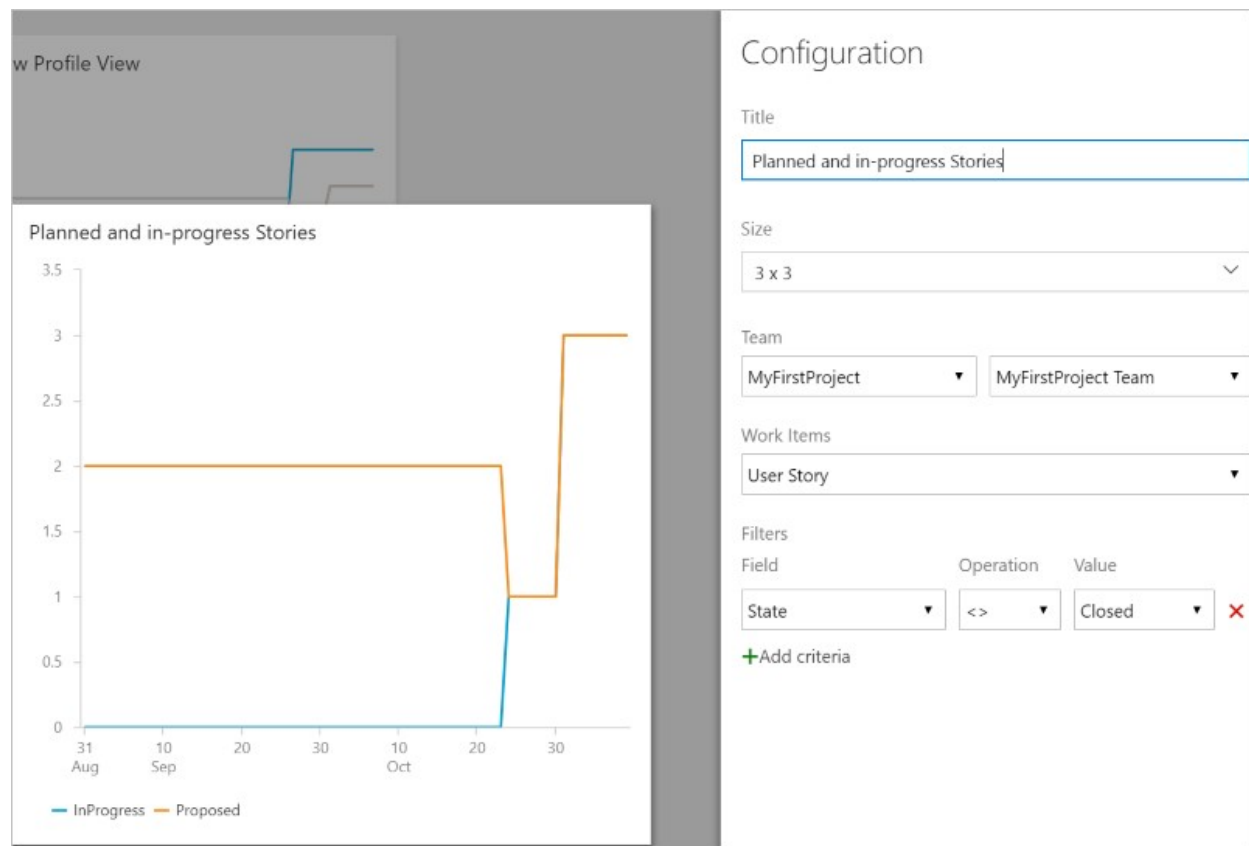
[Next steps](#)

[Related articles](#)

## Azure DevOps Services | Azure DevOps Server 2019

You can build your own Analytics widget to display in a [Dashboard](#) in Azure DevOps. The [example widget on github](#) demonstrates the following:

- ✓ How to render trend lines associated with work item states
- ✓ How to query a dataset, whe a user will configure through the widget configuration view
- ✓ How to build and publish the widget to the Azure DevOps marketplace With this information, you'll be able to create your own Analytics widget.



### ⓘ Observação

The Analytics Service is generally available for all organizations using Azure DevOps Services. It provides several [advanced widgets](#). [Power BI integration](#) and access to the [OData feed](#) of the Analytics Service is in Preview. We encourage you to use it and provide us feedback. As we add features, we will post them

on the [Microsoft DevOps Blog](#).

If you are looking for information about Azure Analysis Services, see [Azure Analysis Services](#).

## Prerequisites

This example provides a ready-made widget, covering basics from topics in Dashboards, Charting and Analytics. The following documents provide more grounding on details demonstrated in this example:

1. Create an Azure DevOps Widget Extension, reference the [Widget extensions sample](#)
2. Render an Azure DevOps Chart Control, reference [Add a Chart](#)
3. Query OData from [Analytics](#)

## Provide a configuration view

1. [Run simple analytics queries](#) required by UI controls used for configuring a view.
2. [Manage state of configuration UI](#), with updates based on user actions, and with new data from Analytics queries.
3. [Render configuration UI](#) using Typescript and React.

## Render data within a Widget

1. Run a user configured query as a [POST Request](#)
2. Interpret data from analytics to [render a chart](#)

## Next steps

To avoid excess complexity in the sample, we omitted certain technologies and practices, which a production widget should certainly include. The [ui-fabric-react](#) sample on github highlights a build process which exercises these details.

1. [Javascript bundling and content minification](#) - The set of small, loose script files in the sample can load much more quickly when consolidate into a single file, and minified.
2. [Fabric UI Controls](#) - Fabric UI controls provide a rich set of configuration UI components for React.

## Related articles

- [What is the Analytics service?](#)
- [Widgets based on the Analytics Service](#)