

WorkshopPLUS - Essentials on Azure DevOps Services and GitHub

Lab Guides

Conditions and Terms of Use

Microsoft Confidential

This training package is proprietary and confidential, and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided **as is** without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Copyright and Trademarks

© 2017 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see Use of Microsoft Copyrighted Content at

<http://www.microsoft.com/en-us/legal/intellectualproperty/Permissions/default.aspx>

DirectX, Hyper-V, Internet Explorer, Microsoft, Outlook, OneDrive, SQL Server, Windows, Microsoft Azure, Windows PowerShell, Windows Server, Windows Vista, and Zune are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

Module 8: Collaboration and Reporting

Lab 1: Collaboration

Introduction

In this lab, you will learn how Microsoft Azure DevOps can help you to collaborate. You will explore Wiki.

[Exercise 1: Wiki collaboration with Azure DevOps](#)

Objectives

After completing this lab, you will be able to:

- Use Wiki to publish and share information.

Prerequisites

None

Estimated Time to Complete This Lab

30 minutes

Module 8: Collaboration and Reporting, Lab 1: Collaboration, Exercise 1: Wiki collaboration with Azure DevOps

Exercise 1: Wiki collaboration with Azure DevOps

Objectives

In this exercise, you will learn how to:

- Create a wiki page
- Search and filter wiki
- View wiki history
- Manage wiki content offline

Prerequisites

None

Scenario

Azure DevOps (formerly VSTS) now supports its own Wiki in each Team Project. This enables you to conveniently write pages that help your team members and other users understand, use, and contribute to your project.

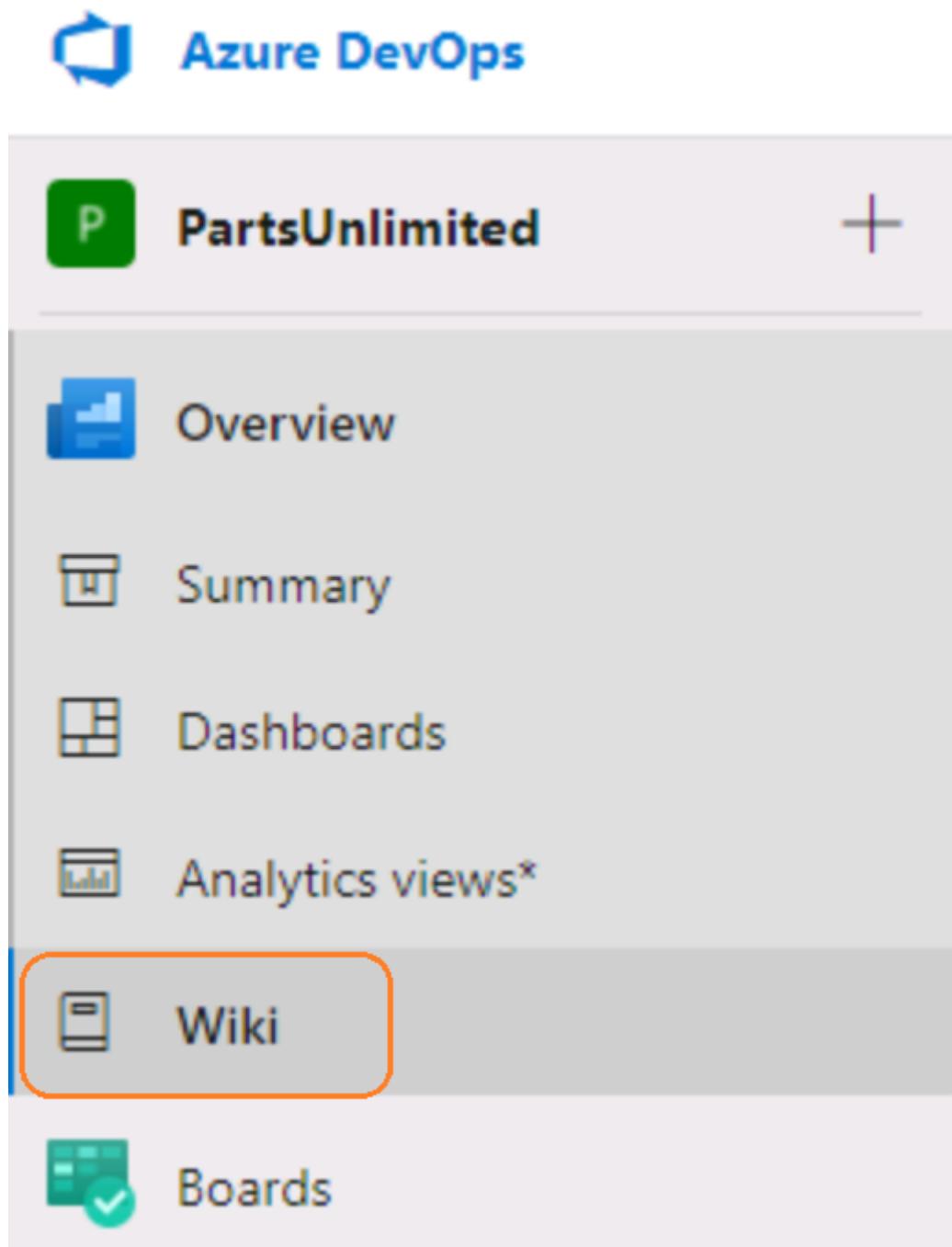
Tasks

- [Task 1: Creating and editing a project wiki](#)
- [Task 2: Working with the wiki offline](#)

Module 8: Collaboration and Reporting, Lab 1: Collaboration, Exercise 1: Wiki collaboration with Azure DevOps

Task 1: Creating and editing a project wiki

1. Navigate to your **PartsUnlimited** project in Azure DevOps Services in the browser.
2. Use the navigation menu on the left to go to **Overview | Wiki**.



3. This project should already have a Wiki (PartsUnlimited_wiki) configured. Move on to the next step if you see **PartsUnlimited_wiki**. If not, click **Create project wiki** to set one up.

When you create your first Wiki, Azure DevOps will provision a git repository that will store all your pages and artifacts.

The screenshot shows the 'PartsUnlimitedWiki' page. At the top, there's a navigation bar with 'PartsUnlimited_wiki' and a dropdown menu. Below it is a search bar with 'Filter pages by title'. The main content area has the title 'PartsUnlimitedWiki' in bold. Underneath the title, it says 'Student1-19678973 Wednesday'. There's a section for 'Sample wiki content' which is currently empty. Below that, a statistic shows '0 visits in last 30 days'. At the bottom, there's a comment input field with a placeholder 'Add a comment...' and a user icon.

4. Click **+ New page** at the bottom if a new page window is not opened already.

A large orange-outlined button labeled '+ New page' is highlighted with an orange box. To its right is a double-left arrow icon.

5. Enter **Home** as the title of your first Wiki page. Enter some body content, such as **Welcome to *our* project!**. Azure DevOps Wikis support Markdown. Click **Save**.

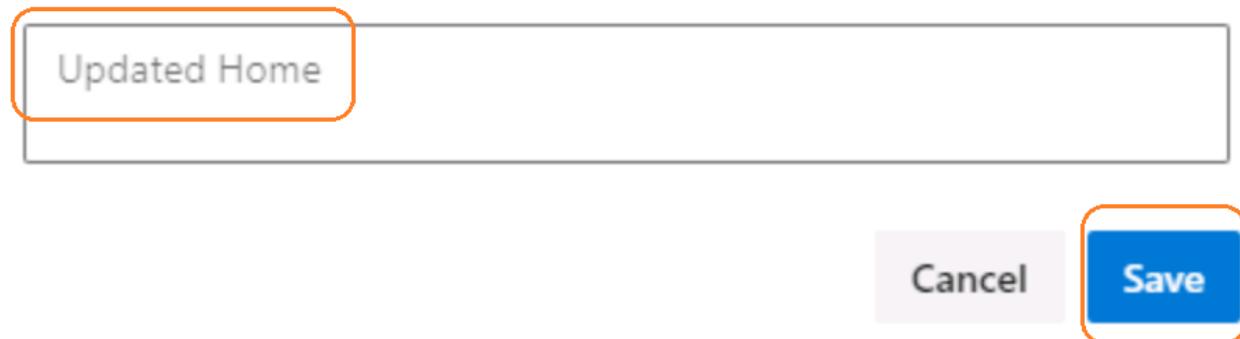
The screenshot shows the 'Home' page in edit mode. The title 'Home' is in the top bar. The content area contains the text 'Welcome to *our* project!'. The toolbar above the content includes buttons for bold, italic, underline, etc. On the far right, there are 'Close', 'Save', and a dropdown menu. A small note 'Markdown supported.' is visible.

6. Add another line in the existing page: **This is a home page.** Click on the dropdown of **Save** and select **Save with revision message**.

The screenshot shows the 'Home' page in edit mode. The content area now includes the text 'Welcome to *our* project!' and 'This is a home page.'. A dropdown menu for the 'Save' button is open, and the option 'Save with revision message' is highlighted with an orange box. The dropdown also includes 'Save' and 'Close'.

7. Since wiki files are stored in a Git repository, this is a commit message. Select the default message and click **Save**.

Save page



8. Click **Close**. Your wiki page **Home** is now available for everyone to see.

Home

Unfollow 1 Edit :

S Student1-19678973 Yesterday

Welcome to *our* project!

This is a home page.

0 visits in last 30 days

Add a comment...

9. By default, all members of the Contributors group can edit README files and Wiki pages. Stakeholders can read files and revisions but cannot edit anything. Click **More actions** for the **PartsUnlimited_wiki** and select **Wiki security** to review permissions.

The screenshot shows the Azure DevOps interface for the 'PartsUnlimited' project. On the left, there's a sidebar with links like Overview, Summary, Dashboards, Analytics views*, and Wiki. The 'Wiki' link is highlighted. On the right, a detailed view of the 'PartsUnlimited_wiki' repository is shown. A context menu is open over the repository name, containing options: Filter, Clone wiki, Wiki security (which is circled in orange), Part, Rename wiki, and Home.

10. Since the Wiki is stored as a Git repo, the permissions are set against the repo and passed through the user experience. **Close** the dialog.

The screenshot shows the 'Security for Wiki' dialog box. It has two main sections: 'Inheritance' (which is turned on) and a search bar for users or groups. Below that is a list of 'Azure DevOps Groups' and 'Users'. For each group/user, there are several permission settings listed on the right, such as 'Bypass policies when completing pull requests', 'Contribute', 'Create branch', etc., each with a dropdown menu showing the current setting (e.g., 'Not set', 'Allow (inherited)').

Setting	Value
Bypass policies when completing pull requests	Not set
Bypass policies when pushing	Not set
Contribute	Allow (inherited)
Contribute to pull requests	Allow (inherited)
Create branch	Allow (inherited)
Create tag	Allow (inherited)
Delete or disable repository	Allow (inherited)
Edit policies	Allow (inherited)
Force push (rewrite history, delete branches and tags)	Not set
Manage notes	Allow (inherited)
Manage permissions	Allow (inherited)
Read	Allow (inherited)
Remove others' locks	Allow (inherited)
Rename repository	Allow (inherited)

11. Locate the Pages panel on the left side of the window. It lists all of your wiki pages. You can easily add and manage pages here. From the **More actions** menu for the **Home** page, select **+ Add sub-page**.

The screenshot shows the 'PartsUnlimited_wiki' project in the Azure DevOps Wiki. The sidebar has a 'Filter pages by title' search bar and a navigation tree with 'PartsUnlimitedWiki' as the root. The 'Home' page is selected. A context menu is open over the 'Home' page, with the 'Add sub-page' option highlighted by a red box. Other options in the menu include 'Copy page path', 'Move page', 'Edit', 'Delete', and 'Open in new tab'. The main content area displays a welcome message 'Welcome to our project!', a note 'This is a home page.', and a statistic '0 visits in last 30 days'. There is also a comment input field with 'Add a comment...' placeholder.

12. Set the title of this new page to **Getting started** and type some body content like **This is the starting page**. Click **Save** to commit.

The screenshot shows the 'Getting started' page editor. The title bar says 'Getting started'. The editor toolbar includes bold (B), italic (I), and other rich text options. The main content area contains the text 'This is the starting page'. There is a 'Markdown supported.' link at the bottom right. The 'Save' button is highlighted with a red box.

13. Click **Close** and click **+ Add a sub-page** to the Getting started page.

14. Set the title for this page to **Development environment**. Add a list of system requirements that includes **Windows Server 2016**, **Team Foundation Server 2018**, and **Visual Studio 2017** using the markdown list format as shown below. Click **Save** and then **Close** the edit view.

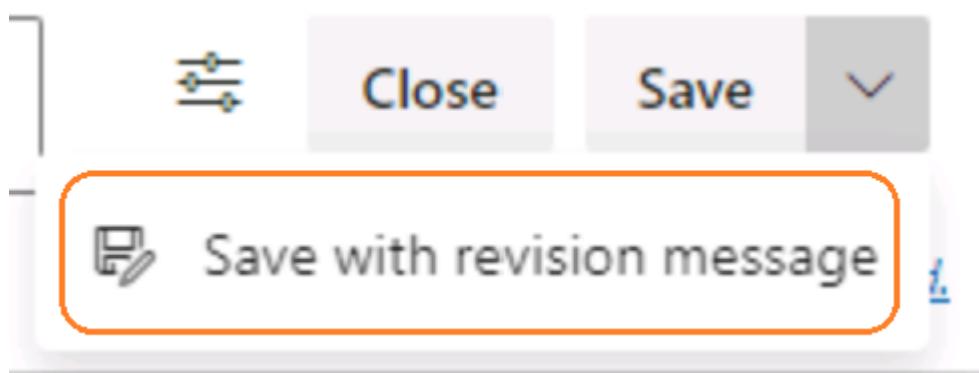
15. Now to edit this page, click **Edit**. Change the reference from TFS 2018 to **Azure DevOps**.

Development environment

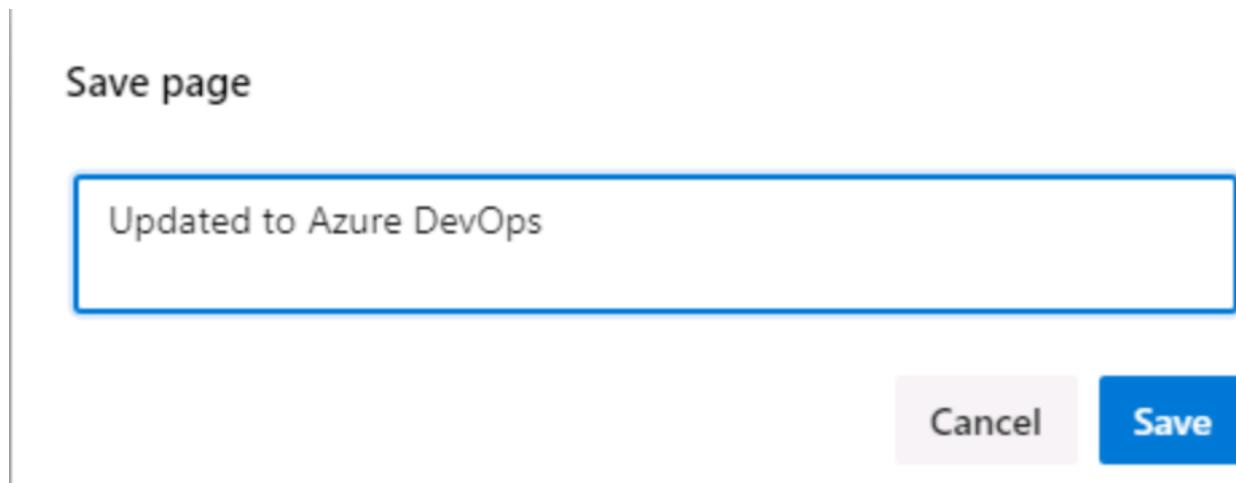
1. Windows Server 2016
1. Azure DevOps
1. Visual Studio 2017

1. Windows Server 2016
2. Azure DevOps
3. Visual Studio 2017

16. Use the **Save | Save with revision message** dropdown to provide a more descriptive commit message.



17. Enter a message explaining the change like **Updated to Azure DevOps** and click **Save**. Then **Close** the edit view.



18. Click the **More actions** and select **View revisions** to see a list of changes committed for this page.

Development environment

S Student1-19678973 Just now

1. Windows Server 2016
2. Azure DevOps
3. Visual Studio 2017

0 visits in last 30 days

Add a comment...

19. Click the most recent revision.

Wikis > PartsUnlimited_wiki > Development environment

Revisions

Author	Message
Student1-19678973	Updated to Azure DevOps
Student1-19678973	Added Development environment

20. This provides a diff view so that you can easily see what was changed. You also have the option to revert to this version with a single button click.

Wikis > PartsUnlimited_wiki > Development environment > Revisions

Updated to Azure DevOps

Student1-19678973 10/27/2021 4:42 PM (UTC) +1 -1

Revert

Compare Preview Go to previous difference Go to next difference Show diff inline

1 1. Windows Server 2016	1 1. Windows Server 2016
2-1. Team Foundation Server 2018	2+1. Azure DevOps
3 1. Visual Studio 2017	3 1. Visual Studio 2017

21. Return to this wiki's home using the breadcrumb navigation at the top.

Updated to Azure DevOps



Student1-19678973 10/27/2021 4:42 PM (UTC)

+1 -1

22. In the Pages panel, you can easily rearrange the order of the pages using the tree view on the right.

Drag the **Getting started** page slightly up until a green line appears **under the Home page**. This indicates that you want to make these pages peers.

The screenshot shows the 'Pages' panel in a wiki interface. At the top, there's a header for 'PartsUnlimited_wiki'. Below it is a search bar labeled 'Filter pages by title'. The main area displays a hierarchical list of pages:

- > [PartsUnlimitedWiki](#)
- ✓ [Home](#) (This page is expanded, showing its children.)
 - [Getting started](#) (This page is being moved, indicated by an orange arrow pointing towards the 'Home' page.)
 - ✓ [Getting started](#)
- [Development environment](#)

23. Since the page is being re-parented, you'll need to confirm the move. This kind of move will break links you will have manually added to other pages (none at this point), so you'll need to decide how you will want to deal with them. In this case there are no links, so click **Move**.

Move 'Getting started'



Moving from '/Home' to '/'.

Affected links to the page:

(Links to recently created or modified pages might not appear.)

Pages 0 Work items 0

No broken links found.

Update affected links in pages and work items (recommended)

Cancel

Move

24. Sometimes wikis can get pretty big, so it's important to be able to find specific pages. Type **env** to filter the pages down to just those whose titles include that text.

PartsUnlimited_wiki ▾ :

env X

PartsUnlimitedWiki

Development Environment

Getting started

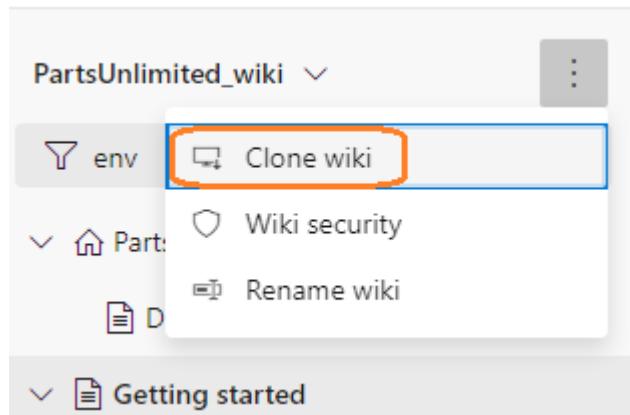
Development environment

Module 8: Collaboration and Reporting, Lab 1: Collaboration, Exercise 1: Wiki collaboration with Azure DevOps

Task 2: Working with the Wiki offline

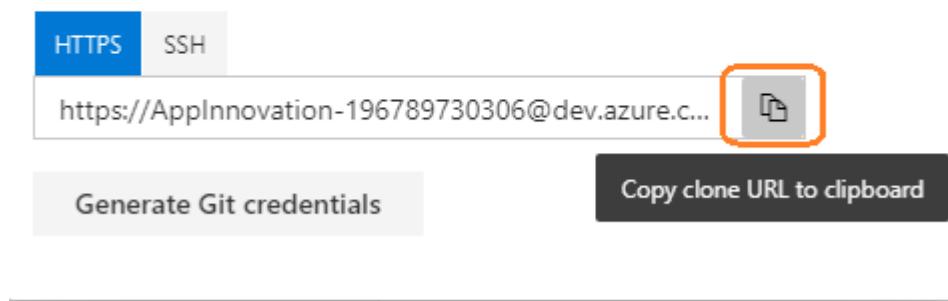
Sometimes you may want to work on the wiki without being connected via browser. Fortunately, your Azure DevOps Wiki is backed by a Git repo, so you could clone it and edit it just like any other Git project.

1. Select **Clone wiki** from **More actions** menu of the wiki to view the clone URL.



2. Click the **Copy** button to copy the URL to the clipboard.

Clone repository



3. Launch a new instance of **Visual Studio** from the taskbar.

4. Click on **Clone a repository** and paste the **copied URL** in **Repository location**. Keep the default **Path** which should look like C:\Users\StudentPC\Source\Repos\PartsUnlimited_wiki. Click **Clone**. Double-click on the folder in the solution explorer.

Clone a repository

Enter a Git repository URL

Repository location

196789730306@dev.azure.com/AppInnovation-196789730306/PartsUnlimited/_git/PartsUnlimited_wiki

Path

C:\Users\StudentPC\Source\Repos\PartsUnlimited_wiki

...

Browse a repository

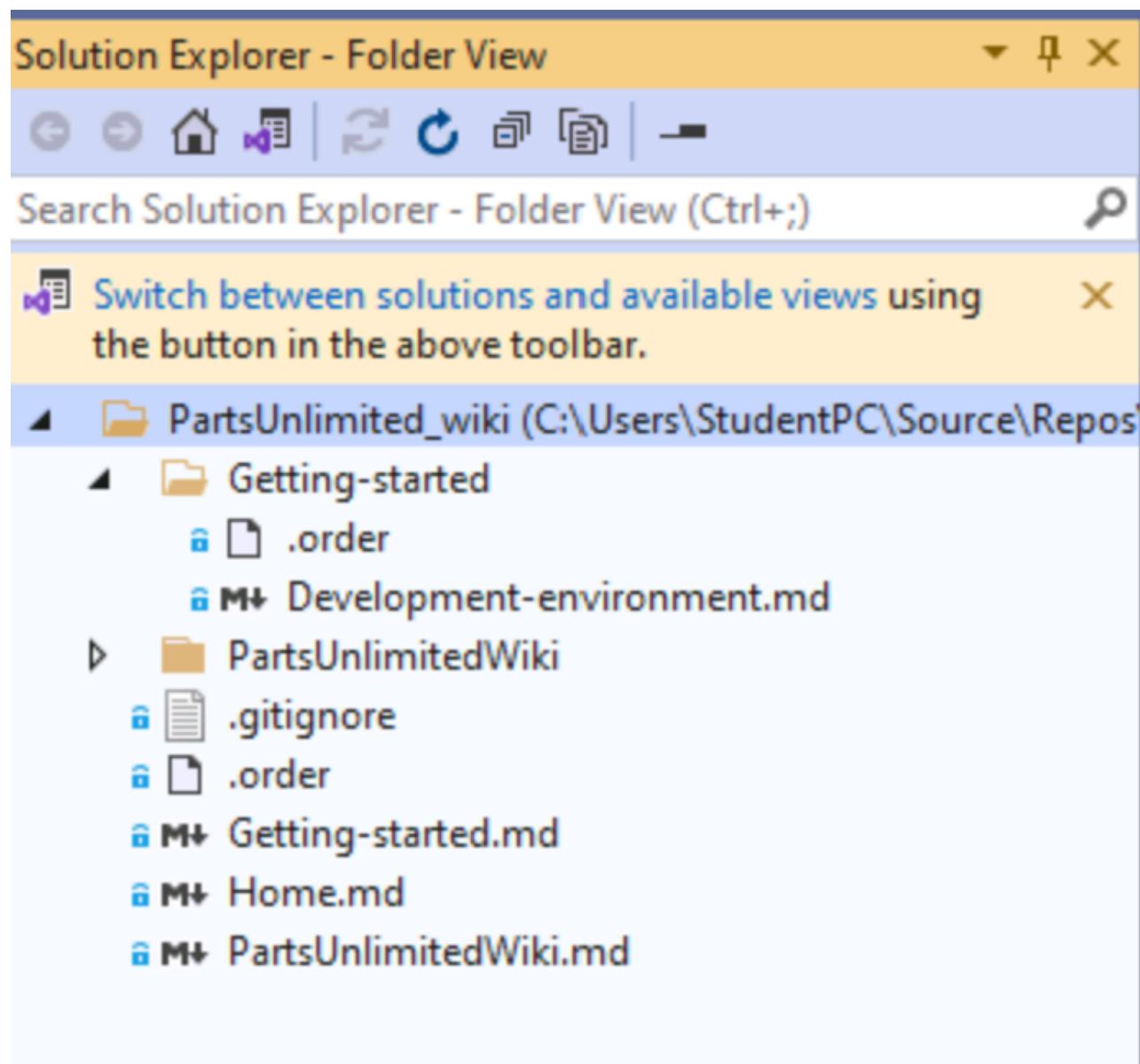
Cloud Azure DevOps

Github GitHub

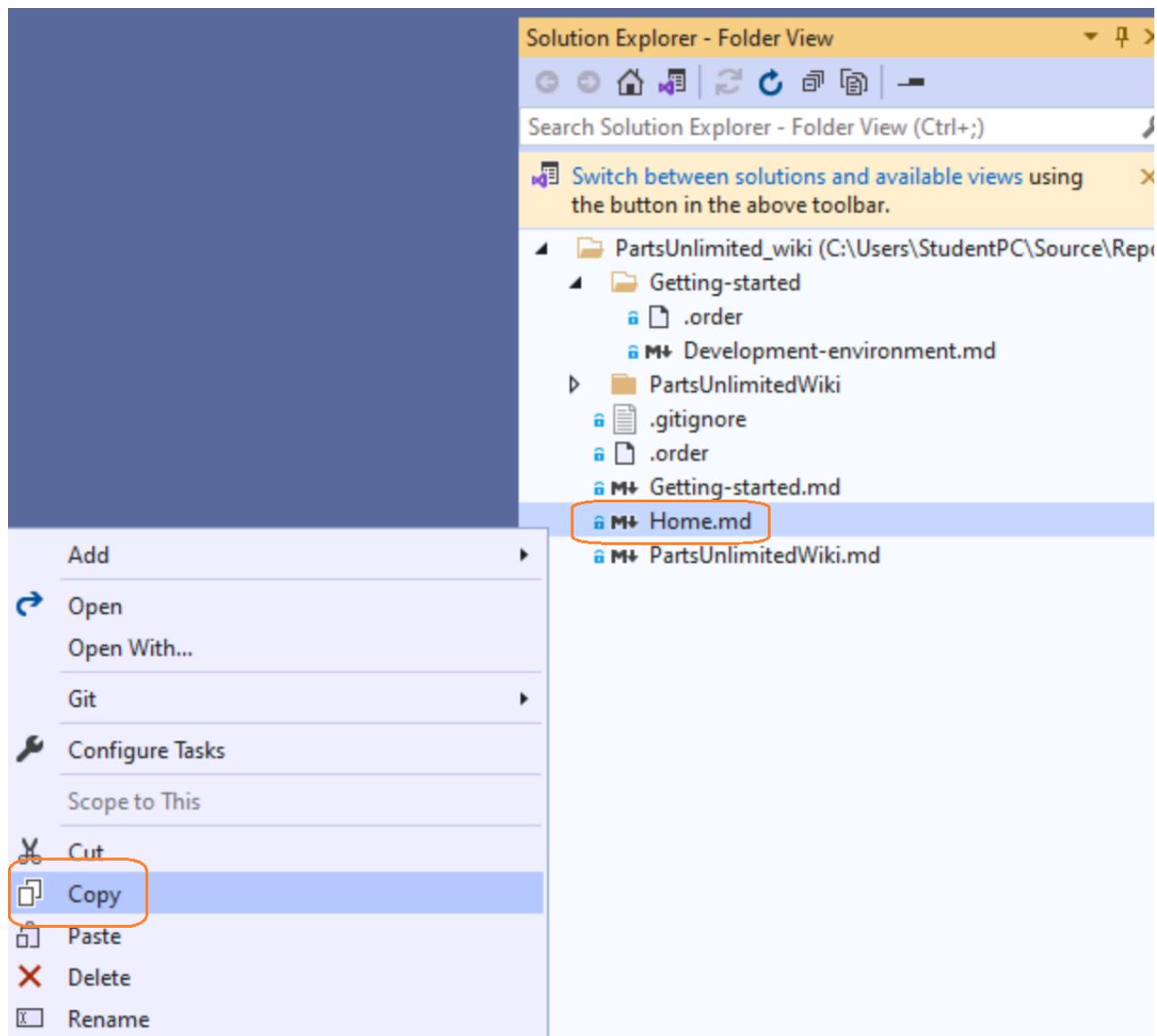
Back

Clone

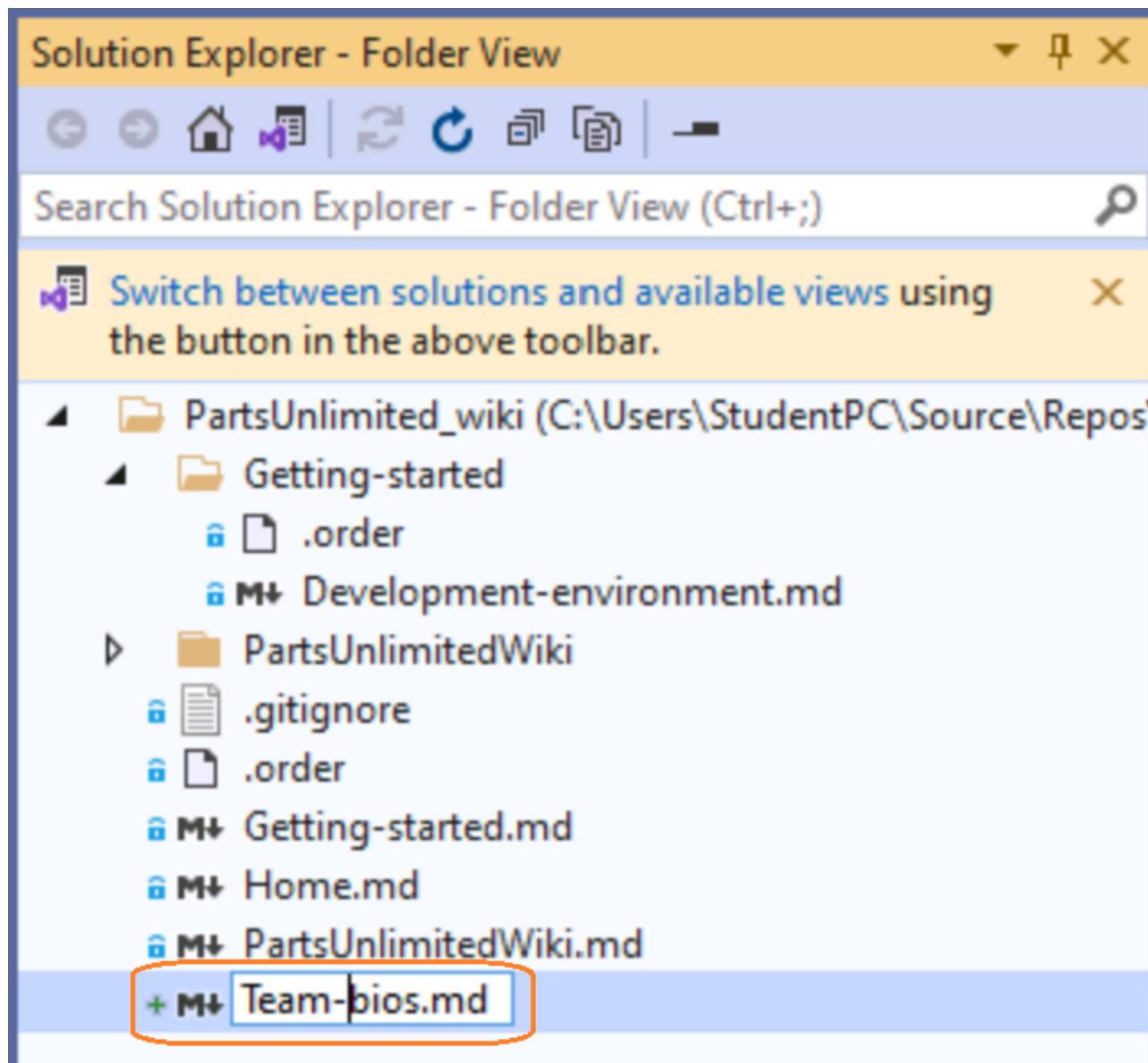
You can now edit the wiki in Visual Studio. You could alternatively do your editing in any other environment or even Notepad. It's all Markdown and Git at this point. The Wiki file structure is fairly straightforward. Markdown files (.md) contain content for each page at that path. If you want sub-pages, then create a folder with the same name of that file (without the .md), such as Getting-started as shown in the screenshot below. Every folder also requires a .order file to specify the order of the pages. If you have any attachments for your wiki, they are stored in a .attachments folder in the root directory.



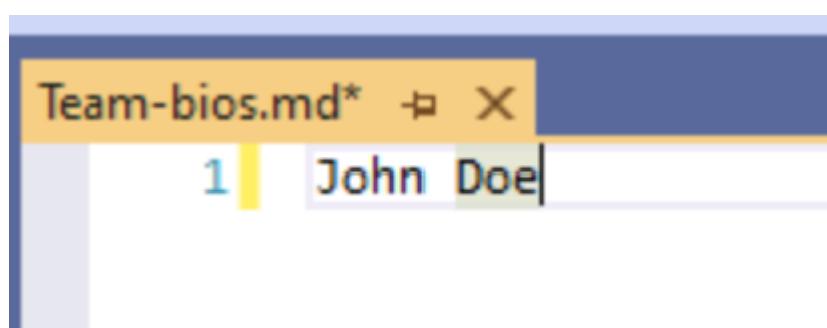
5. Start off by adding a new file. To keep things simple, just **copy and paste Home.md** in the Solution Explorer.



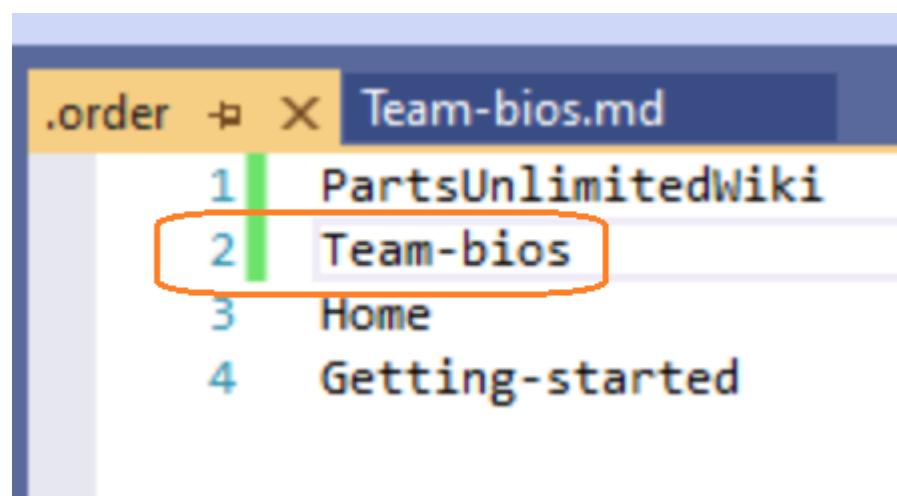
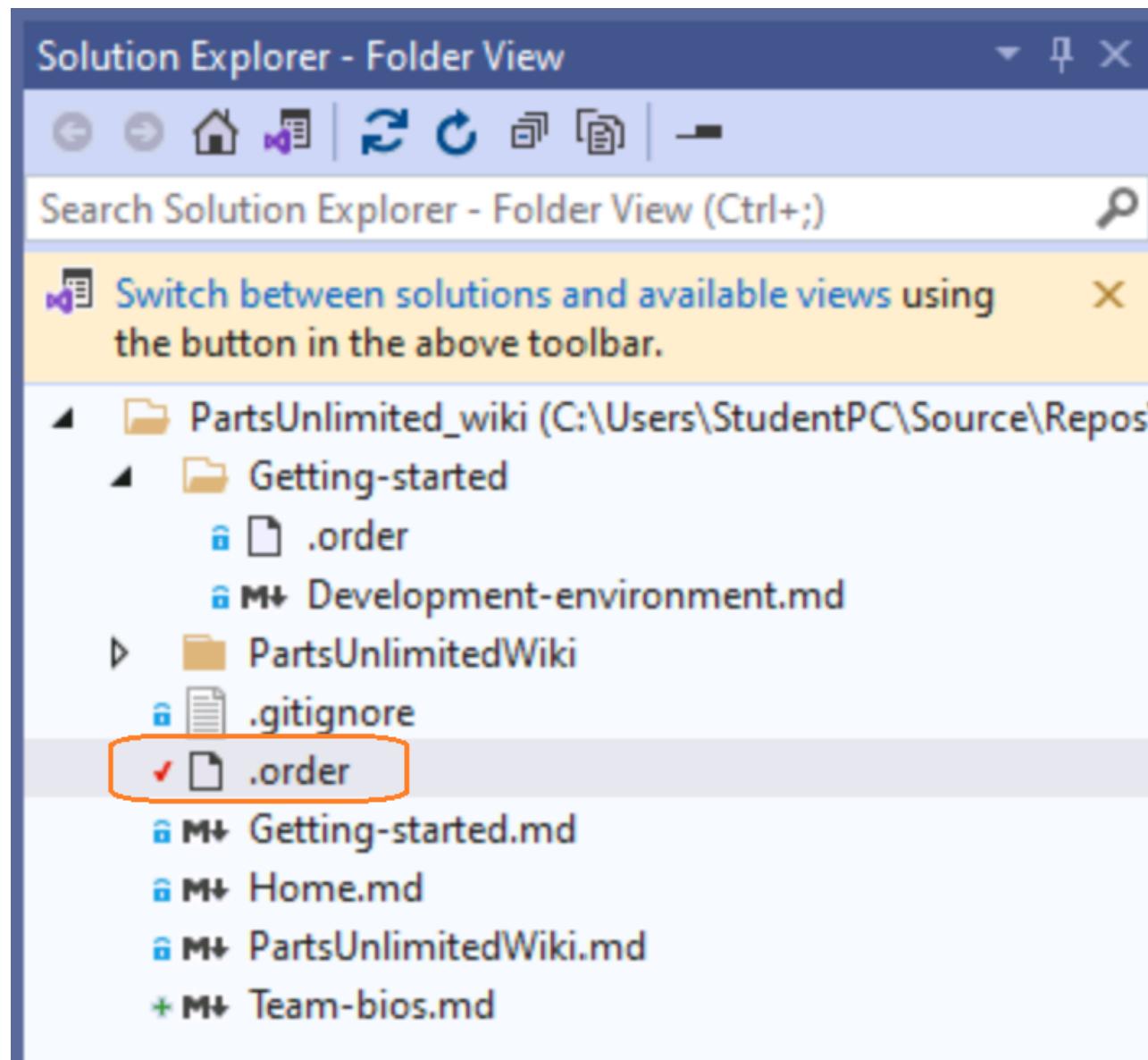
6. Rename the copied file to **Team-bios.md**.



7. Add names of some team members in the Team-bios.md file and **save** it.



8. Open the **.order** file from the root of the repo. This is the top-level order. Note that the format of this file is straightforward — just put the file names (without .md) in the order you want them to appear in the wiki. Add **Team-bios** as the second line. **Save** the file.



9. There is also a .order file in Getting-started folder, but that's just for .md files in that folder. We won't change that here.

10. Switch to the **Git Changes** at the bottom-right of Visual Studio.



i This repository's origin remote is hosted in Azure DevOps. [Connect](#) to Azure DevOps to enable access to work items and builds.

[Don't show again](#)

wikiMaster ▼ ▾ ↴ ↵ ⌂ ...

↑↓ 0 outgoing / 0 incoming

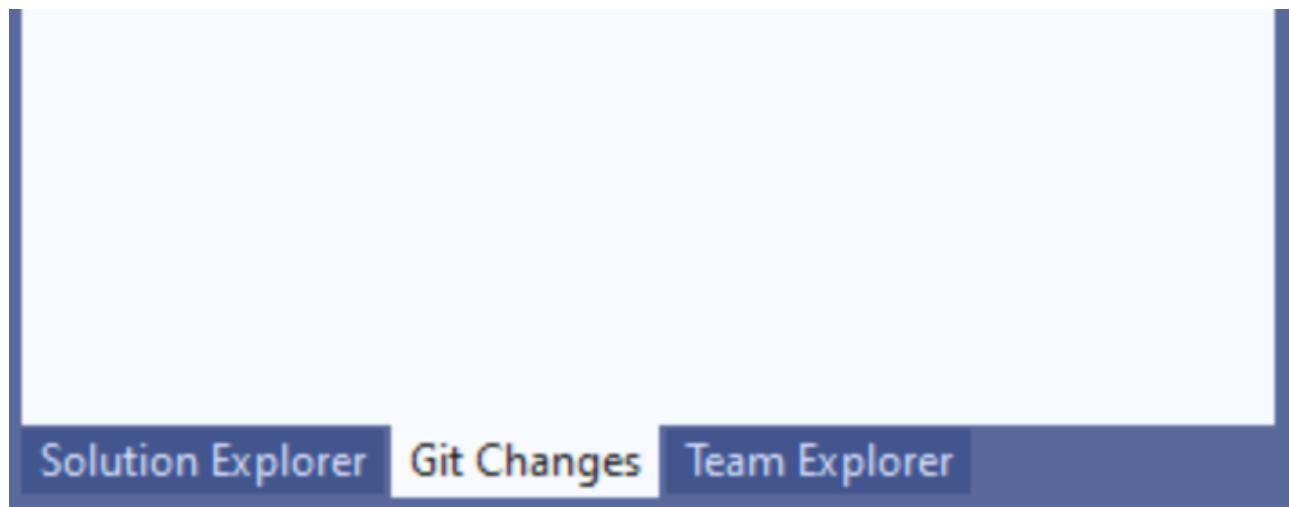
Enter a message <Required>

Commit All ▼ Amend

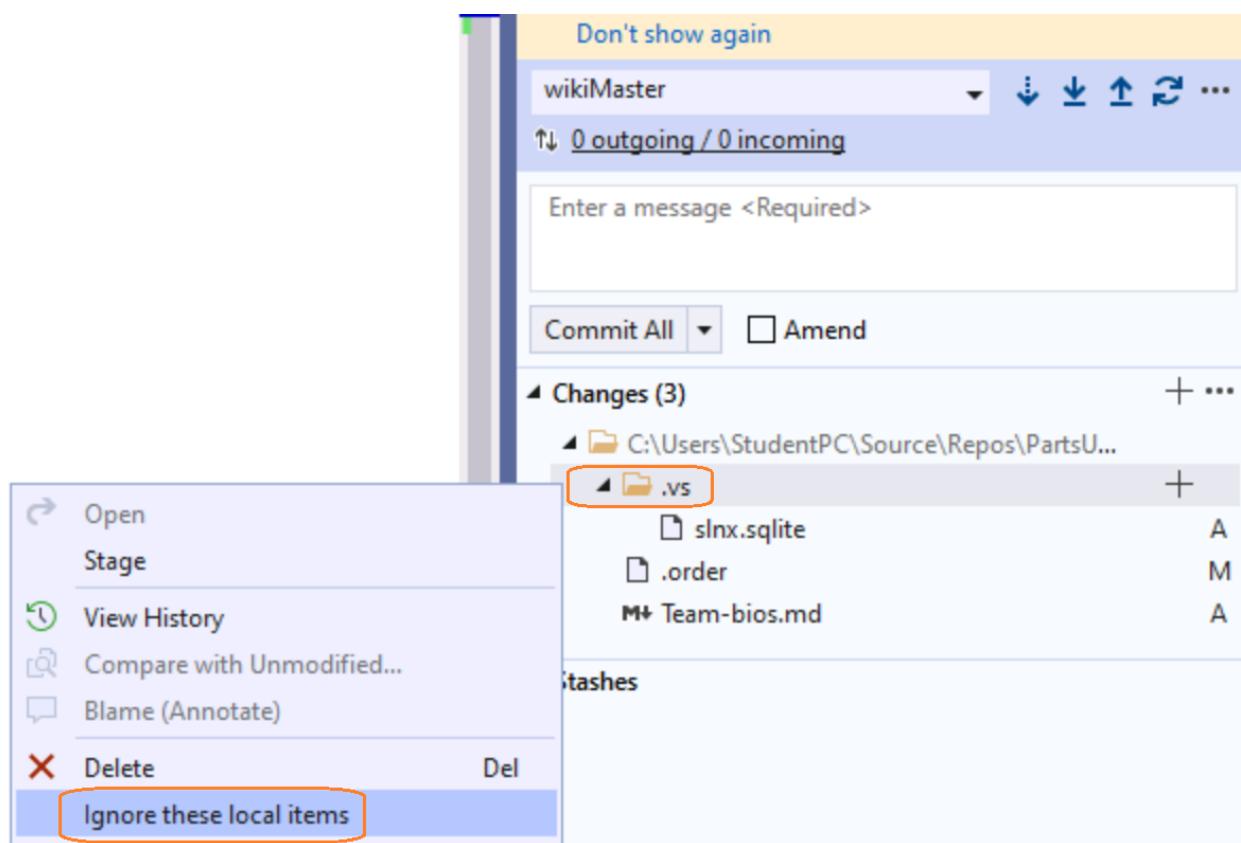
Changes (3) + ...

- ◀  C:\Users\StudentPC\Source\Repos\PartsU...
 - ◀  .vs
 -  slnx.sqlite A
 -  .order M
 -  Team-bios.md A

▷ Stashes



11. Visual Studio will have added some additional files to the folder. Right-click the **.vs** folder and select **Ignore these local items**. This will update .gitignore for you.



12. Add a message **Added team bios** and select **Commit All | Commit All and Sync**.

13. Once the sync has completed, return to the browser and **refresh** the page. Your new team bios page will appear in the navigation, so select it to view the content.

The screenshot shows a sidebar menu for a wiki titled "PartsUnlimited_wiki". The menu items include "PartsUnlimitedWiki", "Team bios" (which is selected and highlighted in grey), "Home", and "Getting started". Above the menu is a search bar with the placeholder "Filter pages by title". To the right of the sidebar, the main content area displays the "Team bios" page. The page title is "Team bios". Below the title, there is a user profile card for "Student1-19678973" (Just now) with the name "John Doe". A horizontal line follows this. Below the line, it says "0 visits in last 30 days". At the bottom right of the page area is a comment input field with a purple circular icon containing a white letter "S" and the placeholder text "Add a comment...".

Team bios

S Student1-19678973 Just now

John Doe

0 visits in last 30 days

S Add a comment...

Module 8: Collaboration and Reporting, Lab 2: Reporting, Exercise 1: Create Dashboard w/ Analytics Widgets

Lab 2: Reporting

Introduction

In this lab, you will learn how Microsoft Azure DevOps can help you to report on the status of the work within the projects using Dashboards and the Analytics widgets. These widgets provide additional insight into the health and status of the work.

[Exercise 1: Create Dashboard w/ Analytics Widgets](#)

Objectives

After completing this lab, you will be able to:

- Create Dashboards and Advanced Analytics service widgets

Prerequisites

None

Estimated Time to Complete This Lab

15 minutes

Module 8: Collaboration and Reporting, Lab 2: Reporting, Exercise 1: Create Dashboard w/ Analytics Widgets

Exercise 1: Create Dashboard w/ Analytics Widgets

Objectives

In this exercise, you will:

- Modify an existing dashboard for the team project
- Add analytics widget

Prerequisites

None

Scenario

Dashboards allow teams to visualize status and monitor progress across the project. At a glance, you can make informed decisions without having to drill down into other parts of your team project site. The Overview page provides access to a default team dashboard which you can customize by adding, removing, or rearranging the tiles. Each tile corresponds to a widget that provides access to one or more features or functions. In this exercise, you will modify the dashboard that you created in a previous lab.

Tasks

- [Task 1: Verify Analytics Views](#)
- [Task 2: Add Work Item Query to Dashboard](#)
- [Task 3: Add Analytics Widget to Dashboard](#)

Module 8: Collaboration and Reporting, Lab 2: Reporting, Exercise 1: Create Dashboard w/ Analytics Widgets

Task 1: Verify Analytics Views

1. Navigate to the PartsUnlimited project and select **Boards | Analytics views**.

The screenshot shows the 'Boards' section of the 'PartsUnlimited' project in Azure DevOps. On the left, there's a sidebar with links like Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views (which is selected and highlighted in grey), Repos, Pipelines, Test Plans, and Artifacts. The main area has a title 'Analytics views (Boards only)' and tabs for Favorites, All (which is selected), and New view. Below that is a descriptive text: 'Analytics views let you create filtered view'. There's a 'Name' input field and two expandable sections: 'My Views' and 'Shared Views'. Under 'Shared Views', there's a list of 10 items: Backlog items - All history by month, Backlog items - Last 26 weeks, Backlog items - Last 30 days, Backlog items - Today, Bugs - All history by month, Bugs - Last 26 weeks, Bugs - Last 30 days, Bugs - Today, Tasks - All history by month, and Tasks - Last 26 weeks.

Name
Backlog items - All history by month
Backlog items - Last 26 weeks
Backlog items - Last 30 days
Backlog items - Today
Bugs - All history by month
Bugs - Last 26 weeks
Bugs - Last 30 days
Bugs - Today
Tasks - All history by month
Tasks - Last 26 weeks

2. The list of views should be like the following:

Analytics views

Favorites All [+ New view](#) [Filter views](#) [Edit](#)

Analytics views are data sets that are exposed to Power BI. You can use views to create reports based on your Azure DevOps data. **This feature is in preview.** [How do I use analytics views?](#)

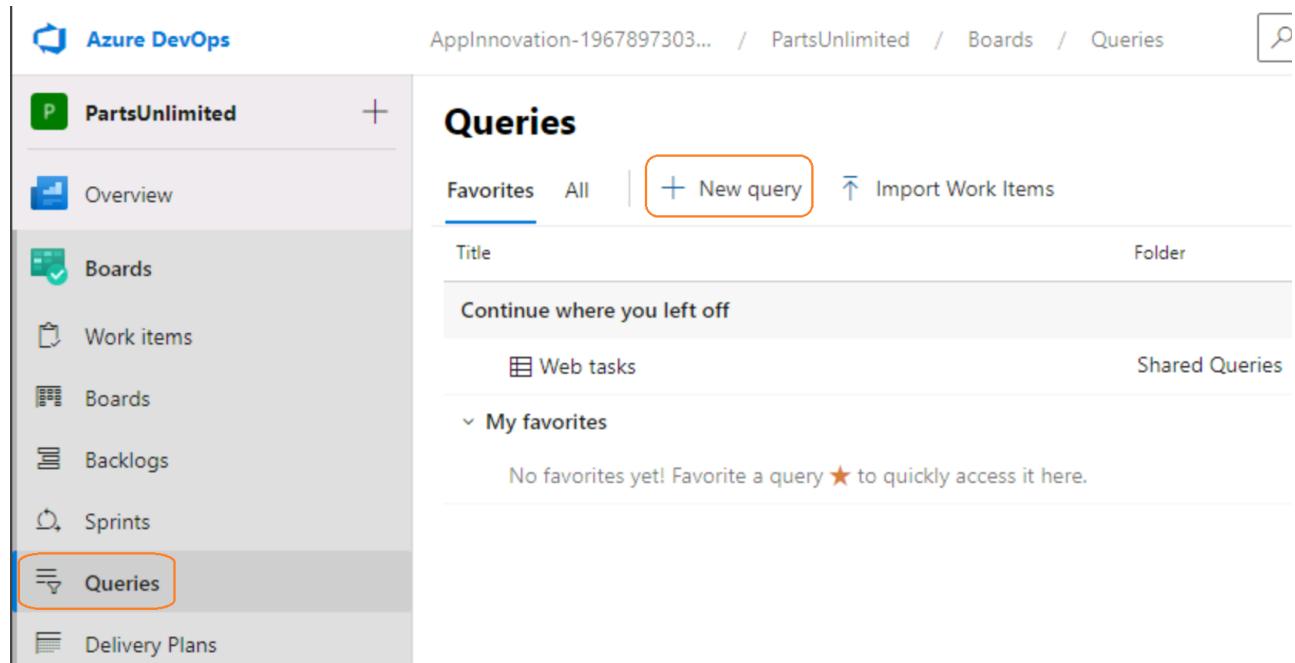
Name	Description	Last modified by
My Views		
Shared Views		
Backlog items - All history by month	All Backlog items for the entire team project. All history with m...	
Backlog items - Last 26 weeks	All Backlog items for the entire team project. Last 26 weeks of ...	
Backlog items - Last 30 days	All Backlog items for the entire team project. Last 30 days of hi...	
Backlog items - Today	All Backlog items for the entire team project. No history.	
Bugs - All history by month	All Bugs for the entire team project. All history with monthly int...	
Bugs - Last 26 weeks	All Bugs for the entire team project. Last 26 weeks of history wi...	
Bugs - Last 30 days	All Bugs for the entire team project. Last 30 days of history wit...	

Analytics view provides a simplified way to specify the filter criteria for a Power BI report based on Analytics data. Analytics views only support Azure Boards data (work items). You can read more about Analytics views [here](#)

Module 8: Collaboration and Reporting, Lab 2: Reporting, Exercise 1: Create Dashboard w/ Analytics Widgets

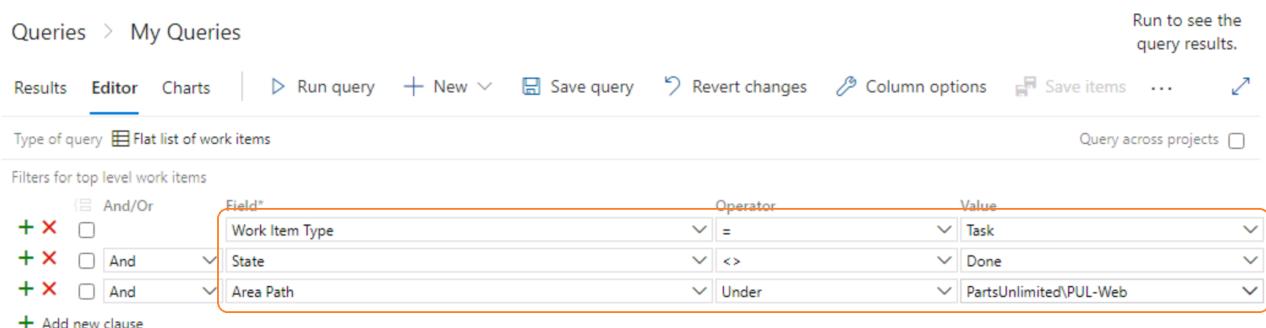
Task 2: Add Work Item Query to Dashboard

1. For the **PartsUnlimited** project, select **Boards | Queries** and click **New Query**.



The screenshot shows the Azure DevOps interface for the 'PartsUnlimited' project. The top navigation bar includes 'Azure DevOps', the project name 'ApplInnovation-1967897303...', and 'Boards / Queries'. The left sidebar has links for 'Overview', 'Boards', 'Work items', 'Backlogs', 'Sprints', and 'Queries' (which is highlighted with an orange box). The main area is titled 'Queries' with tabs for 'Favorites' and 'All'. A button '+ New query' is highlighted with an orange box. Below it, there's a search bar and a section titled 'Continue where you left off' with a 'Web tasks' entry under 'Shared Queries'. A section for 'My favorites' is shown with the message 'No favorites yet! Favorite a query ★ to quickly access it here.'

2. Set the first clause as **Work Item Type = Task** and second clause as **State <> Done**. Add another clause as **Area Path under PartsUnlimited\PUL-Web**.



The screenshot shows the 'My Queries' page with the 'Editor' tab selected. The query editor displays the following clauses:

- Field*: Work Item Type Operator: = Value: Task
- Field*: State Operator: <> Value: Done
- Field*: Area Path Operator: Under Value: PartsUnlimited\PUL-Web

There are buttons for 'Run query', 'New', 'Save query', 'Revert changes', 'Column options', 'Save items', and a 'Run to see the query results' link.

3. Click **Save Query**.

4. Set the Name as **All Open Tasks** and the Folder to **Shared Queries**. Click **OK**.

Only the charts created from the Shared Queries can be added to the dashboard.

New query

Name *

Folder *

OK Cancel

5. Select the **Charts** tab and click **New chart**.

Queries > Shared Queries > All Open Tasks ▾ ★

Results Editor **Charts** | Run query New Save query Save as... Rename

Type of query Flat list of work items

Filters for top level work items

(And/Or	Field*	Operator
+ <input type="checkbox"/>	Work Item Type	=
+ <input type="checkbox"/> And	State	<>
+ <input type="checkbox"/> And	Area Path	Under

+ Add new clause

6. Set the **Chart Type** to **Column**, **Name** of the chart to **All Open Tasks – Assigned To** and **Group by** to **Assigned To**. Click **OK** to save.

Configure Chart

Chart Type

- Pie
- Bar
- Column
- Stacked bar
- Pivot table
- Stacked area
- Area
- Line

Name: All Open Tasks – Assigned To

Group by*: Assigned To

Aggregation: Col of work items

Sort: Value Descend

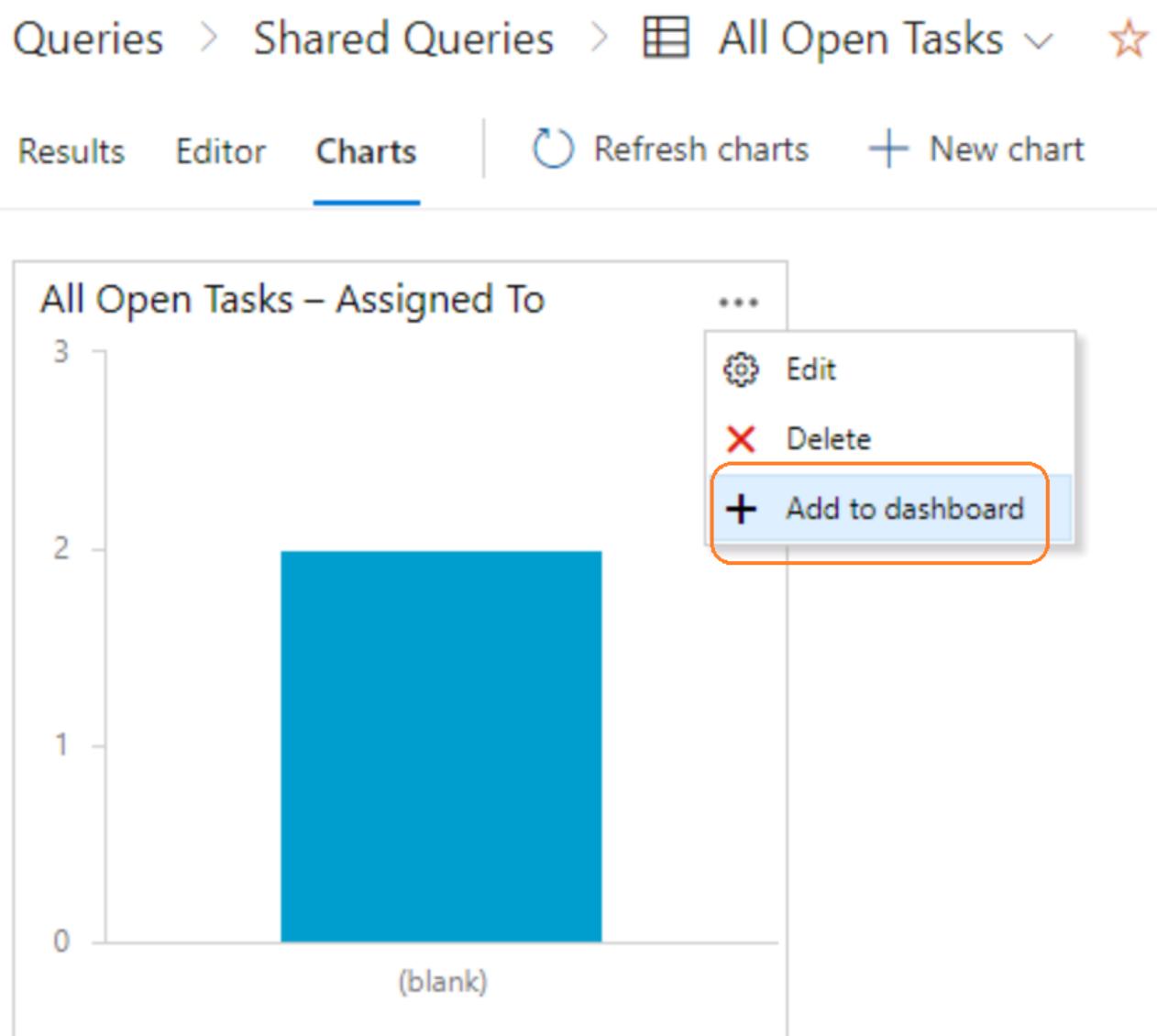
Series: (blank)

Clear custom colors

The chart displays a single blue bar representing the count of open tasks assigned to '(blank)'. The y-axis ranges from 0 to 3.

OK Cancel

7. Add the chart to the dashboard.



8. Select the **Product training** dashboard under PUL-Web and click **OK**.

Select a dashboard

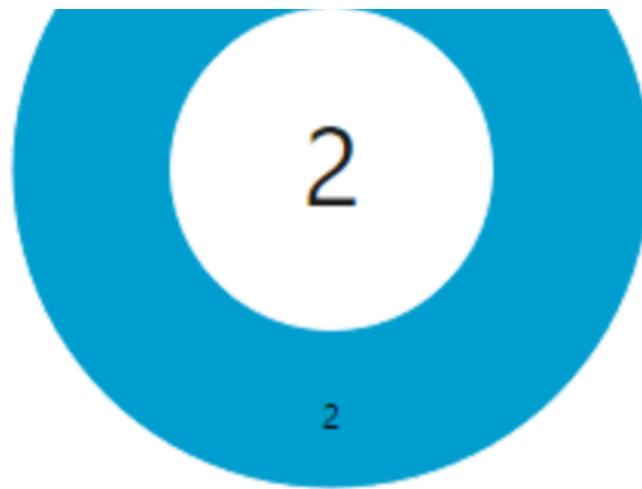
By selecting a dashboard you will create a copy of **All Open Tasks – Assigned To** as a widget on that dashboard

The screenshot shows a search bar at the top with the placeholder 'Search'. Below it is a list of dashboards under 'PartsUnlimited Team': 'Overview' for 'PartsUnlimited Team', 'External integration team', and 'PUL-Web'. Under 'PUL-Web', there is an 'Overview' and a 'Product training' option, which is highlighted with an orange border. At the bottom of the list is a link 'All team dashboards'.

9. Select **Overview | Dashboards** on the left navigation pane.

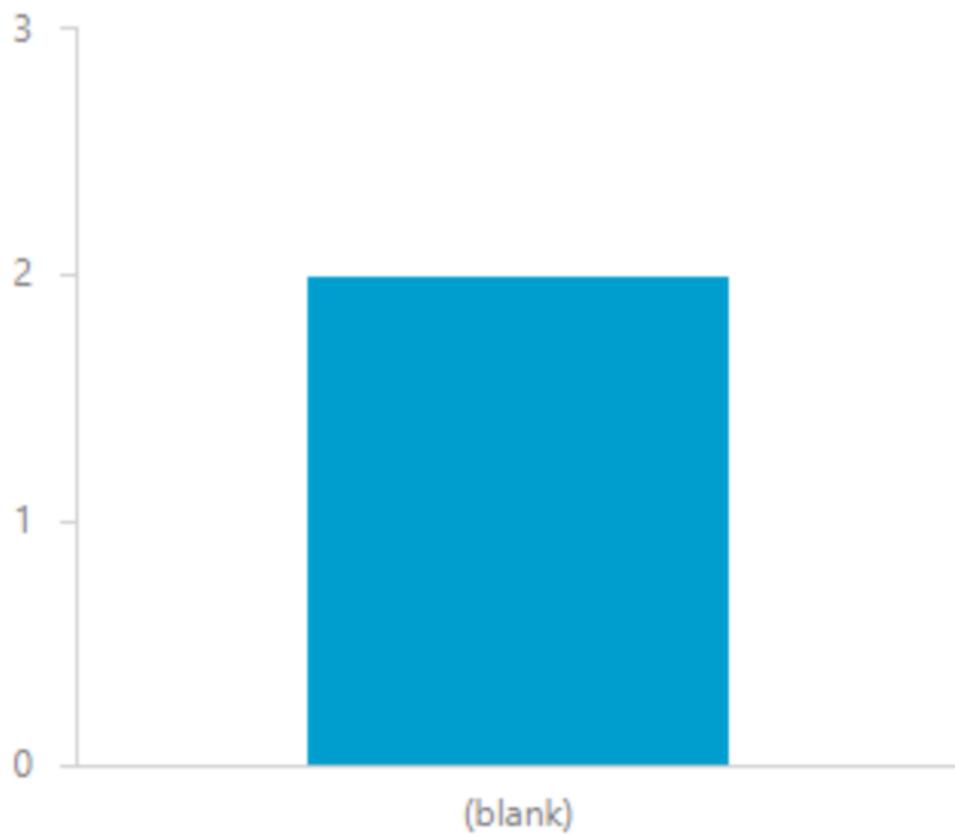
10. Select the **PUL-Web - Product training** dashboard from under PUL-Web to see the chart that was added to the dashboard.

The screenshot shows the 'PUL-Web - Product training' dashboard. The title bar has a star icon, a magnifying glass icon, and a pencil icon. Below the title, there is a progress bar labeled 'Backlog items: 1 NOT started, 1 in progress'. A large blue donut chart at the bottom indicates '32 / 38'.



■ (blank)

All Open Tasks – Assigned To



Module 8: Collaboration and Reporting, Lab 2: Reporting, Exercise 1: Create Dashboard w/ Analytics Widgets

Task 3: Add Analytics Widget to Dashboard

1. While staying in the the **PUL-Web - Product training** Dashboard and select **Edit**.
2. Enter the search term **burn** in the search box and **add** the **Burndown** and **Burnup** widgets.

Add Widget

burn



Burndown
Displays burndown across multiple teams and multiple sprints. Create a release burndown or bug burndown.



Burnup
Displays burnup across multiple teams and multiple sprints. Create a release burnup or bug burnup.

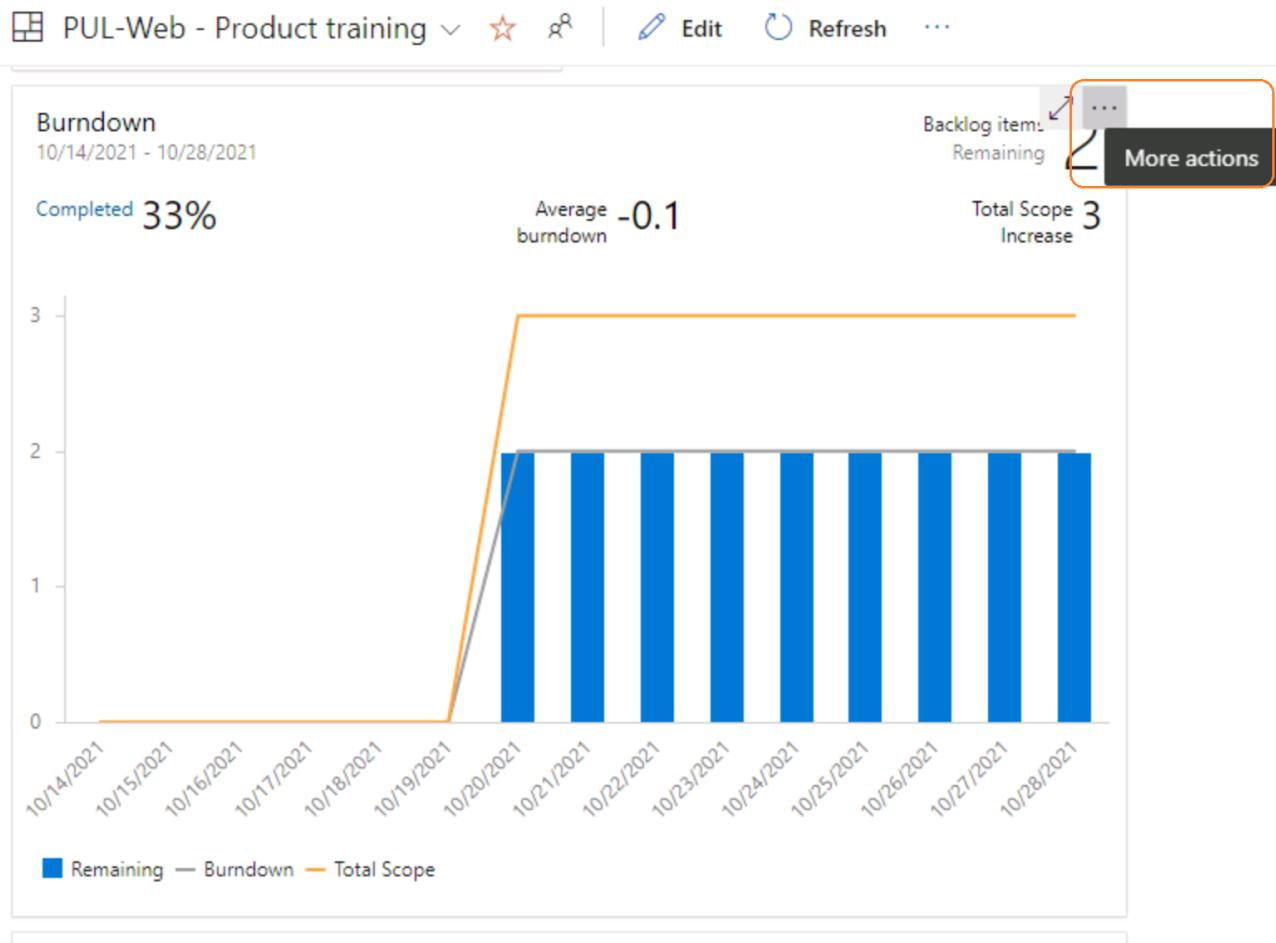


Sprint Burndown
Displays a burndown chart for the work of a team in a single iteration.



Sprint Burndown (Legacy)

3. Click **Done Editing**.
4. On the **Burndown** widget, select **More actions** and **Configure**.



5. Keep the defaults to the for all setting and select the checkbox **Show Completed Work** and select **Save**.

Configuration

BURNDOWN CHART

Count

of

work items

▼

Time period

Start date

10/14/2021



Plot burndown by

Date

▼

End date

10/28/2021



Plot interval

Days

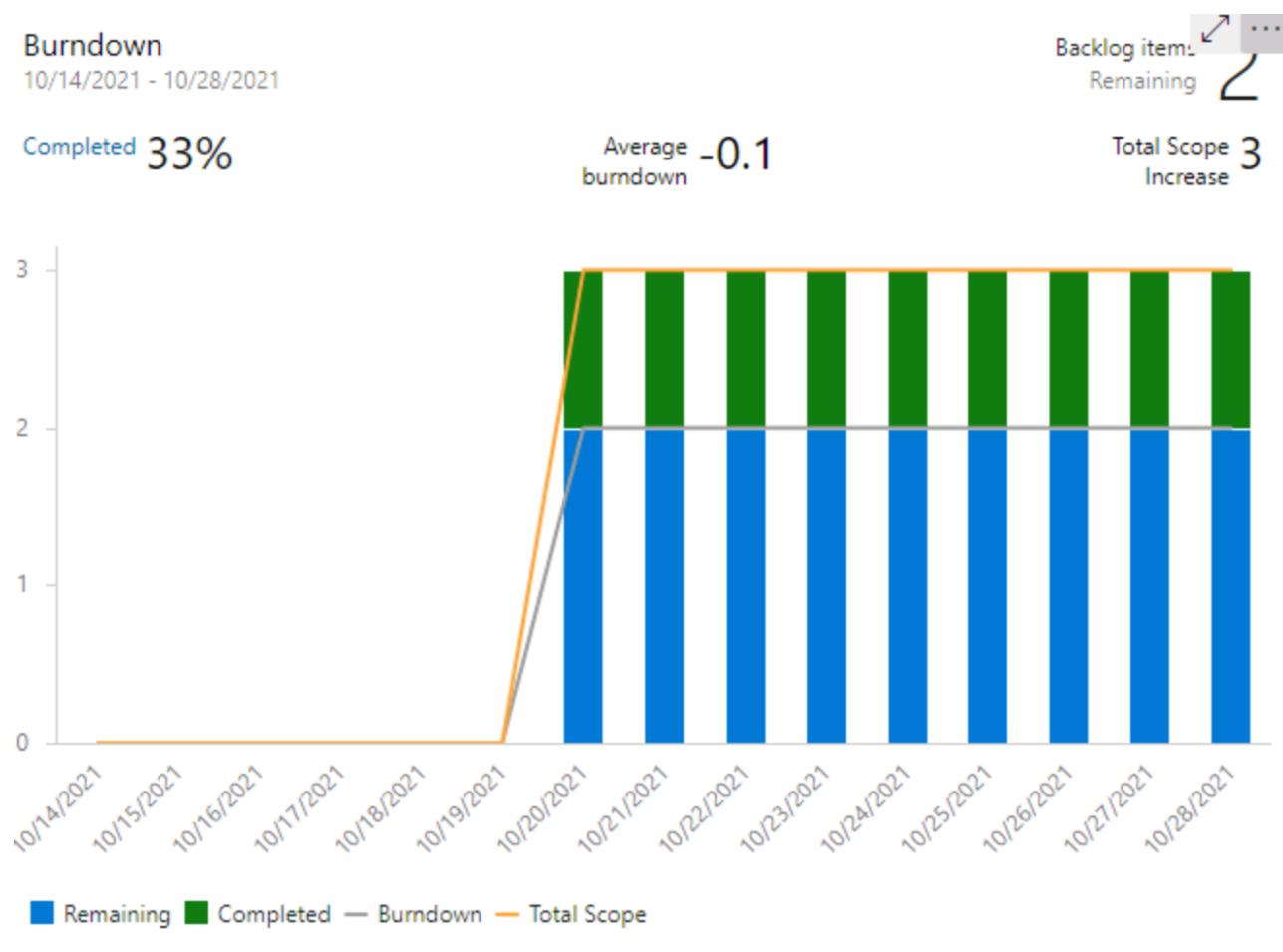
▼

Advanced features

- Show burndown
- Show total scope
- Show completed work
- Plot remaining work using work item type color

Close

6. The Burndown chart would look like this:



7. Follow the same steps for the Burnup widget.

