

- [Contributor's guide](#)
 - [When to contribute](#)
 - [Does the topic or doc exist?](#)
 - [Optional editing tools](#)
 - [Best practices](#)
 - [Before you start](#)
 - [Local Git repo](#)
 - [Prerequisites](#)
 - [Clone the repo](#)
 - [Start the development server](#)
 - [Edit content \(minor changes\)](#)
 - [Edit content \(significant changes\)](#)
 - [Add new content](#)
 - [Submit a Pull Request](#)
 - [Release notes](#)
 - [API documentation](#)

Contributor's guide

Thank you for taking an interest in contributing to the Beyond Identity Secure Workforce documentation.

We've built the documentation with [Docusaurus](#) and you can find the source code in our [customer-docs GitHub repo](#).

You're welcome to contribute to the documentation to help make it better and we want to make it possible for you to become a contributor. Additionally, anyone can [suggest ideas](#) to improve the content or open an [issue](#) about the content.

This guide outlines the different ways that you can get involved. It also provides some of the resources that are available to help you get started. All feedback is welcome.

When to contribute

When relying on others to draft and review documentation, it makes sense to have a process to make it more productive and encouraging for everyone.

The acceptance criteria for contributing to Beyond Identity developer documentation is:

Does the topic or doc exist?

- If yes, update the doc! Contribute to your heart's content. But first, determine which use case best fits your update:
 - **Minor changes: (use case #1)** If you only need to fix a typo or make a clarification to a sentence or section, this option is your ideal choice. However, you cannot upload files, such as an image, in this use case, and you can update only one file.

- **Significant changes: (use case #2)** Suppose you need to make substantial changes, such as adding an image or editing multiple topics. In that case, this method is your best friend, as you'll have one pull request to manage.
- If not, should it? If so, consider **adding it (use case #3)**, but don't add a topic or doc just because you can. Consult with the Sr. Technical Writer first before adding. For example, your new topic could be added to an existing one, consolidating and clarifying the information. It's also possible that someone else is working on a similar topic. The Sr. Technical Writer maintains the content's quality, reduces duplication, and provides a consistent user experience. They can also strategize how to deliver complex information—for example, adding an explainer video (<90 secs) can help with complex scenarios or issues.

Optional editing tools

If you want the WYSIWYG experience, you can install one of the VS Code extensions:

- [Rich Markdown Editor](#)
- [Markdown Editor](#)
- [UNOTES](#)

Other helpful VS Code extensions:

- [Code Spell Checker](#)
- [Markdown All in One](#)

Non VS Code markdown editors:

- [StackEdit](#)
- [Dillinger](#)
- [Pandeo MEditor.md](#)

Best practices

To promote consistency, follow the guidelines below and mentioned throughout to keep the process productive and beneficial for all involved.

- ☐ Always create a *working branch* whenever you introduce a set of logically related changes. This helps you manage your changes through the workflow. We refer to it here as a working branch because it's a workspace to iterate or refine your changes until they can be integrated into the default branch.
- ☐ The in-browser editing experience is best for minor or infrequent changes.
- ☐ Always consult with the Sr. Technical Writer when adding topics or want to propose changes to the content structure. The Sr. Technical Writer maintains the content's quality, reduces duplication, and provides a consistent user experience. They can also strategize how to deliver complex information—for example, adding an explainer video (<90 secs) can help with complex scenarios or issues.
- ☐ Always keep your branch up-to-date.

Before you start

If you're ready to get started contributing to the Beyond Identity customer-docs repo:

- ☐ Ensure you're familiar with Git and GitHub, and [markdown](#). We use [basic markdown](#), [extended markdown](#), and [Docusaurus-specific markdown](#).
- ☐ Review the [customer-docs repo's](#) organization and structure. You can learn more about the structure, see [Site organization and files](#).
- ☐ Familiarize yourself with our [style guide](#).
- ☐ Verify that you can open a [pull request](#) and review changes.
- ☐ Ensure an [issue](#) exists if you find a problem.
- ☐ Read the [style guide](#).

Local Git repo

Prerequisites

- Code editor, [VS Code](#) is recommended
- [Git](#)
- [Node.js](#)
- [Yarn](#)

Clone the repo

1. Clone the repo. This creates a directory named [developer-docs](#) in your current working directory.

```
git clone https://github.com/gobeyondidentity/developer-docs.git
cd developer-docs
```

2. Next, create a new [branch](#) from [main](#) (always give your branch a meaningful, descriptive name).

```
git checkout -b <branch_name> main
```

Start the development server

We're finally able to get to the fun stuff! Install the dependencies and start a local development server:

```
yarn
yarn start
```

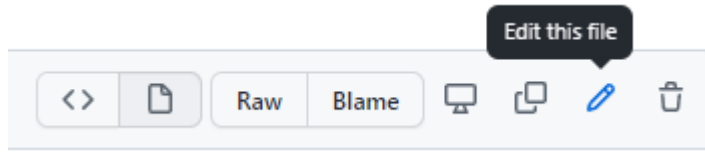
The <http://localhost:3000> opens in your browser so you can see your changes reflected live.

Edit content (minor changes)

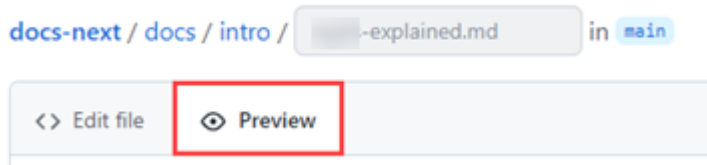
If you only need to fix a typo or clarify a sentence or section, this option is your best choice. It streamlines the process of reporting and correcting minor errors and omissions in the documentation.

Despite all efforts, small grammar and spelling errors *do* make their way into our published documents. While you can create GitHub issues to report mistakes, creating a PR to fix the issue is faster and easier when the option is available.

1. In the [repo](#), navigate to the /docs folder and locate the file you want to edit. Then, select the pencil icon to edit the article.



2. Edit the file in the web editor. Choose the **Preview** tab to check the formatting of your changes.



3. When you're finished editing, scroll to the bottom of the page and propose your changes.

1. In the **Commit changes** area, enter a title and an optional description for your changes. The title will be the first line of the commit message.
2. Select the option to create a branch and then **Propose changes** to commit your changes.

Commit changes

A screenshot of the GitHub 'Commit changes' form. The title field contains the text 'removed incorrect image' and is highlighted with a red box. Below it is a text area for an optional extended description. At the bottom, there are two radio button options: 'Commit directly to the main branch.' and 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' The second option is selected and highlighted with a red box. Below the radio buttons is a text field containing the branch name 'pmcphoe-patch-1'. At the very bottom, there are two buttons: 'Propose changes' (highlighted with a red box) and 'Cancel'.

4. Now that you've proposed and committed your changes, you need to ask the owners to review your changes. This is done using a pull request (PR).

Enter a title and a description for the PR, and then select **Create pull request**. You can select **Create a draft pull request** instead. The difference is that the reviewers are notified only when the PR is *Open* rather than *Draft*.

removed incorrect image

Write Preview

H B I

@

Ensure your markdown formatting follows the [markdown reference guide] ([../markdown-reference.md](#)) of this project.

Description of change or addition

<!-- Describe your changes in detail. Provide a link to the page in the doc site you're referencing -->

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

- ✓ Create pull request
Open a pull request that is ready for review
- Create draft pull request
Cannot be merged until marked ready for review

That's it! Team members will review your PR and merge it when approved. You may get feedback requesting changes.

Edit content (significant changes)

If you need to make substantial changes, add an image or add a section to a topic. On the other hand, if your change is minor that doesn't involve uploading a file or making changes to multiple files, follow the [Edit content - minor change](#) instructions.

1. Make sure you're back on the main (default) branch so you can sync your local working branch.

```
git checkout main
git pull origin main
```

2. Create a working branch for your proposed changes. For example, **content-dev-faqs** or **update-visual-guidelines**.

```
git checkout -b {branch-name}
```

3. Navigate to /docs and locate the topic you want to change.

4. Change the date in the metadata and ensure the title, description, and author are correct. Add any additional keywords if they are missing.

```
---
title: Welcome to Secure Workforce
id: welcome
description: ''
slug: /
keywords:
  - faq
  - help
pagination_next: null
pagination_prev: null
last_update:
  date: 03/29/2023
  author: Patricia McPhee
draft: true
doc_type: overview
product: secureWorkforce
hide_table_of_contents: false
displayed_sidebar: secureWorkforceSidebar
---
```

5. Make your changes to the content and run the following command to build and review your changes.

```
yarn start
```

This command starts a local development server and opens a browser window. Most changes are reflected live without having to restart the server. If a browser window doesn't open, go to **<http://localhost:3000/SCOOBY/docs-next/>** to view your local changes.

6. Add your changes.

```
git add --all
```

7. Commit your changes.

```
git commit -m "short description of the change"
```

8. Publish, or push, your changes to the main repo for merging.

```
git push --set-upstream origin {branch name}
```

9. From the main (default) branch, select **Compare & pull request**.
10. Provide additional information, add reviewers, and select **Create pull request**.

Alternatively, you can select **Create draft pull request**.

11. After going through the review process for technical accuracy and a copy edit from a Content Designer, enter **#sign-off** and select **Comment**.

Your changes get merged into the default branch (**main**). After each merge, the Sr. Technical Writer reviews all merges, fixes any merge conflicts, and publishes all changes.

Add new content

1. Make sure you're back on the main (default) branch so you can sync your local working branch.

```
git checkout main
git pull origin main
```

2. Create a working branch for your proposed changes. For example, **content-dev-faqs** or **update-visual-guidelines**.

```
git checkout -b {branch-name}
```

3. Navigate to **/docs** and locate the subdirectory for your new topic.
4. Add the metadata (frontmatter).

Keywords are optional but helpful. All other metadata is required.

```
---
title: Welcome to Secure Workforce
id: welcome
description: ''
slug: /
keywords:
  - faq
  - help
pagination_next: null
pagination_prev: null
last_update:
  date: 03/29/2023
  author: Patricia McPhee
draft: true
doc_type: overview
product: secureWorforce
hide_table_of_contents: false
```

```
displayed_sidebar: secureWorkforceSidebar
---
```

5. Add the new markdown file in the sidebar.js file.

```
{
  type: 'category',
  label: 'Content',
  link: {
    type: 'doc',
    id: 'content/content-overview',
  },
  items: [ 'content/language-inclusivity', 'content/grammar-mechanics',
'content/voice-tone-principles', 'content/in-product-word-list',
'content/writing-for-errors', 'content/writing-guidelines' ],
},
```

It's a best practice to add the markdown file early in the content development phase. Otherwise, you'll get an error when you run the command to build and review your content.

6. Create your content and run the following command to build and review your changes.

```
yarn start
```

This command starts a local development server and opens a browser window. Most changes are reflected live without having to restart the server. If a browser window doesn't open, go to **<http://localhost:3000/SCOOBY/docs-next/>** to view your local changes.

7. When ready, commit and publish your changes to the main (default) branch, add your changes.

```
git add --all
```

8. Commit your changes with a descriptive summary, for example, *add clarity to the data table component usage examples*.

```
git commit -m "short description of the change"
```

9. Publish, or push, your changes to the main repo for merging.

```
git push --set-upstream origin {branch name}
```


10. From the main (default) branch, select **Compare & pull request**.

11. Provide additional information, add reviewers, and select **Create pull request**.

Alternatively, you can select **Create draft pull request**.

12. After going through the review process for technical accuracy and a copy edit from a Content Designer, enter **#sign-off** and select **Comment**.

Your changes get merged into the default branch (**main**). After each merge, the Sr. Technical Writer reviews all merges, fixes any merge conflicts, and publishes all changes.

Submit a Pull Request

All code in Beyond Identity aligns with the established licensing and copyright notice guidelines.

Before submitting a Pull Request, review the general Beyond Identity Pull Request Guidelines and make sure that you provide the information that is required in the Pull Request template in that specific repo.

Beyond Identity requires the use of commit signoffs for each PR. Every commit to this repo must be signed off using **#approved** by the required reviewers and **#sign-off** to trigger the merge. Otherwise, the PR could not be merged.

Once you've reviewed the PR and approved it, add a Signed-off-by line to your commit message. For example, **Signed-off-by: John Doe john.doe@beyondidentity.com**.

Release notes

API documentation