Figma Guide for Designers

02 Jan 2024

Document changes

Author	Date	Description	Version
Lucas toh	29/Nov/2023	File created	1.0.0
Lucas Toh	02/Jan/2024	Update Release note preparation guide	1.0.1

Purpose & Objective

Be able to navigate, find design, and know which file to work on and which to send for development.

At the end of this guide you will understand:

- Create branches
- Manage and use UI Library
- Create and update the cover with a plugin/widget
- Prepare pages with the proper structure
- Use dots with colour on pages
- Build a user flow
- Prepare instruction guide
- Create prototype
- Request branch review
- Prepare the requirement for the “Hand-shake”
- Merge approved branch

Branching

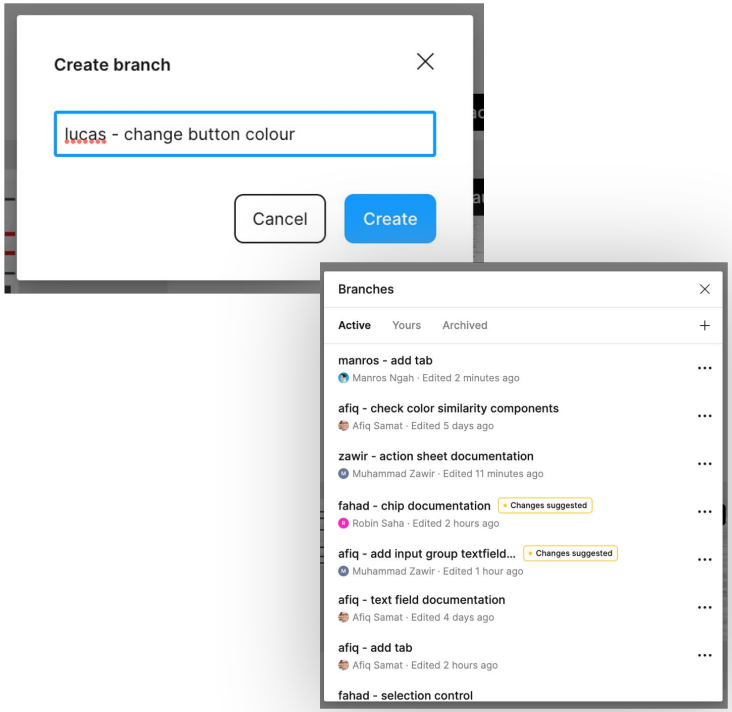
When and how to create branch

- Branching in Figma allows us to create alternate versions of a design file and make changes to it without affecting the original file.
- We create multiple branches for different sprint and merge these branches back into the main design file when they're ready for developers.

- **When to do branching?**
  A branch must be created from the main file if you want to make any changes to the design for a task.

- **How to do branching?**
  Refer to the video and the two links below for more information.
  - Video guide
  - Best practice
  - Guide to branching

While creating a branch, type your name and what you will do in this branch by following this format: “name – what you did”.

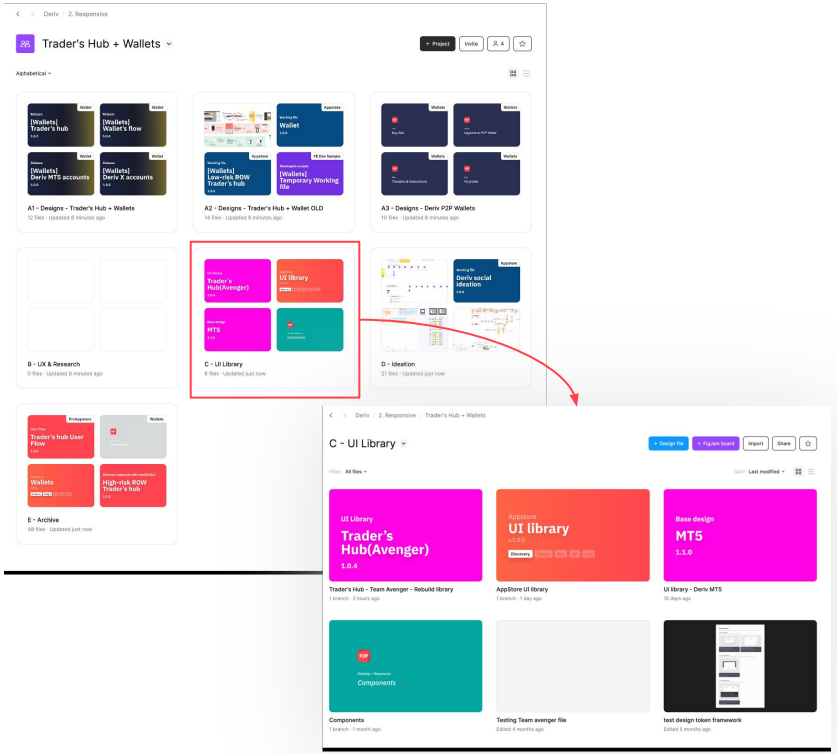


Manage and use UI library

What is and where should I store a UI Library

- A UI library is a central location for all the unique components needed for a product. It can be used as a reference for designers and developers to ensure consistency across the product.
- Each component within the library must follow the style guide and be composed of components sourced from the design system.

- UI library must be stored in “Folder” under the relevant product.
  - It will be named as “**C. UI library**” folder.
  - Any designer can update and create components in the UI library provided the changes are done inside a branch.

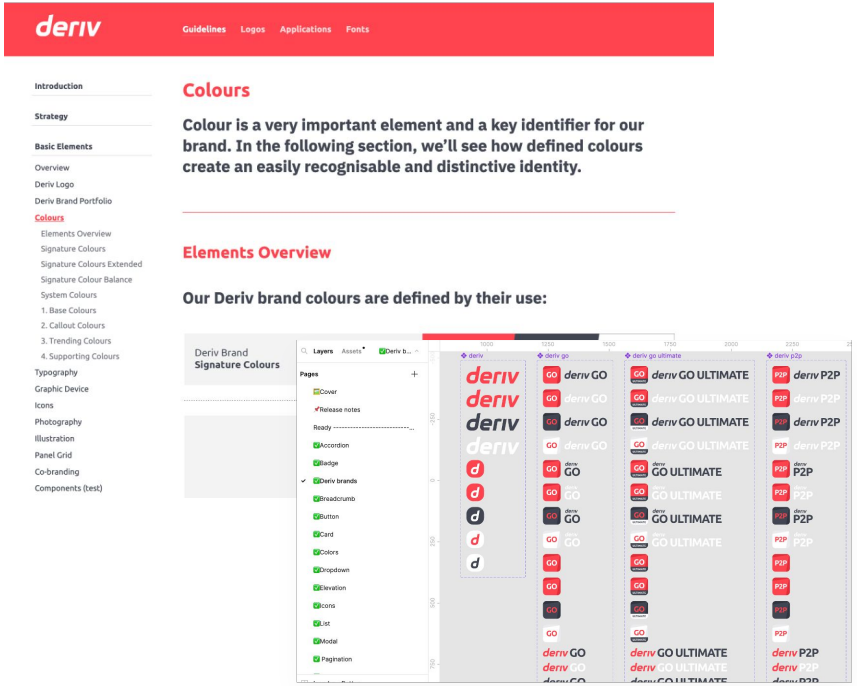


Build a library with design systems

To build a new component:

- Use the style guide and branding from Deriv Brand Guideline. E.g.:
  - Text sizes and text styles
  - Colors

- Use one or more atom-level components from design systems to build new components in the UI library. E.g.:
  - Icons
  - Buttons
  - Text fields



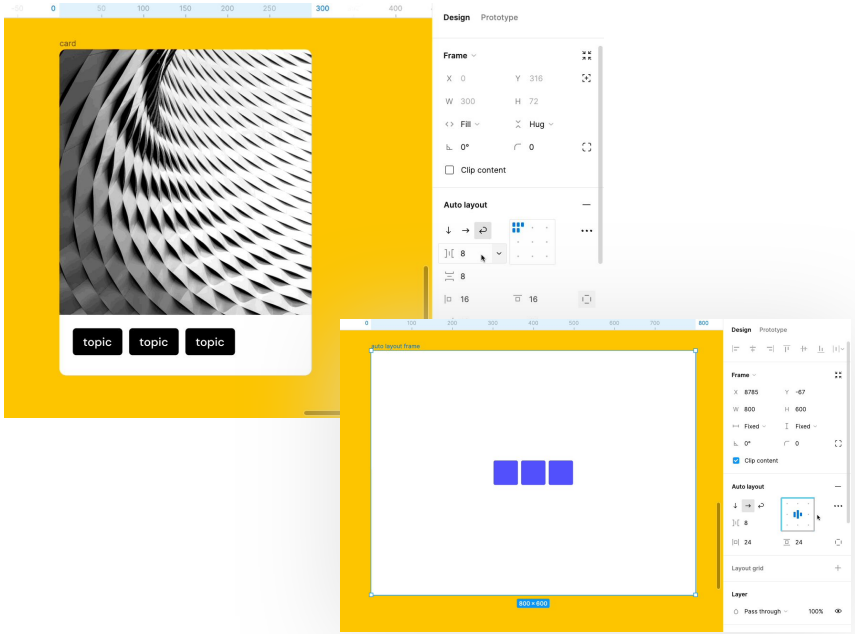
How do I build new components?

Build components with Figma using “component properties”:

- Auto layout
- Section

Learn more through the Figma guide:

- Article: Using auto layout
- Article: Explore auto layout properties
- Video: What’s new in Auto layout
- Article: Update old auto layout frames
- Article: Organize your canvas with sections
- Article: Groups and Frames



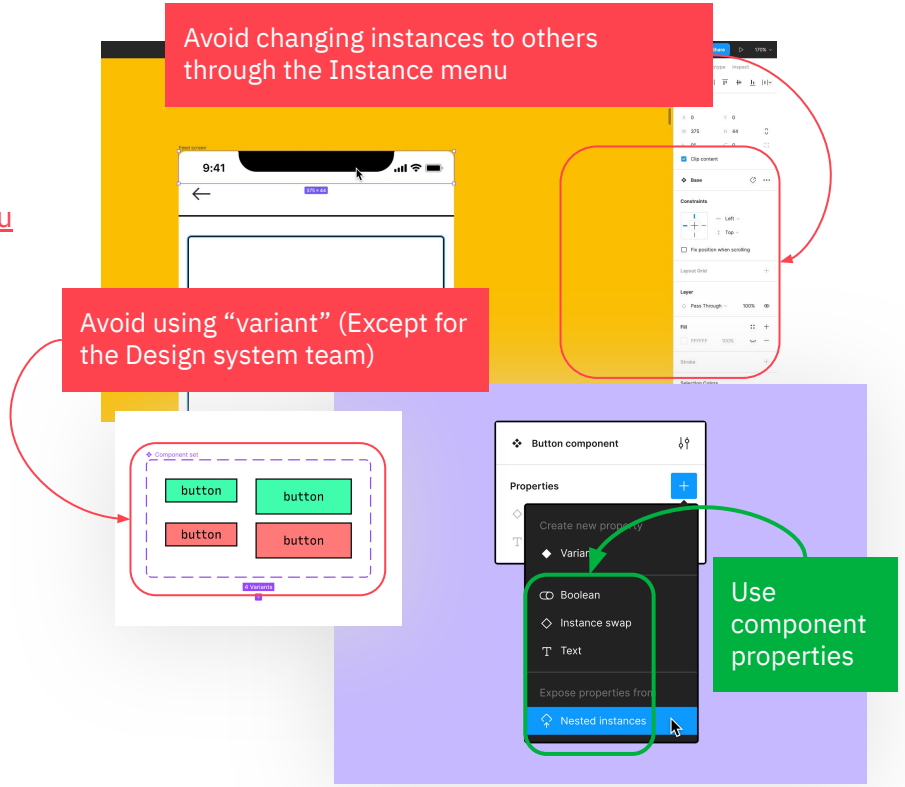
How do I build new components?

Build components with Figma using “component properties” such as Boolean, Instant swap, Text, and Nested instance:

- ***Avoid*** changing instances to others through the Instance menu
- ***Avoid*** using “variant” (Except for the Design system team)

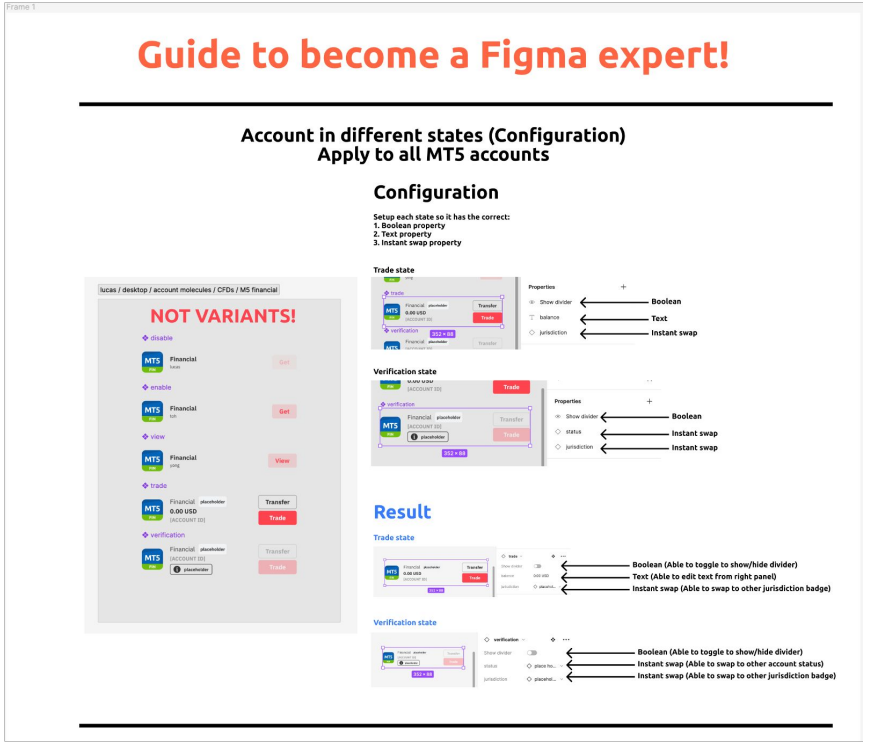
Learn more through the Figma guide:

- Article: Explore component properties
- Video: Component properties
- Article: Create and manage component properties
- Video: Training with Figma’s designer



How to use those properties together?

Please refer to the guide in this Figma link



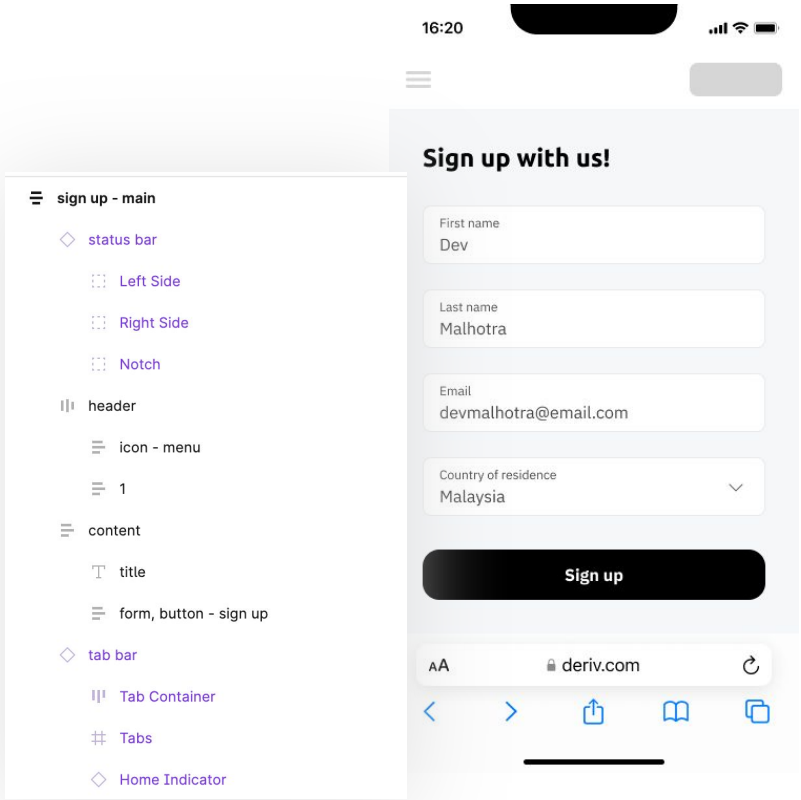
Name layers properly

Proper layer naming helps us to identify the layer and understand the structure easily. It reduces confusion, maintains coherence, and improves workflow. Here are the rules we should follow:

- Use dash “-” with space on both side. E.g., sign up - main, dropdown - country of residence.
- Only use capital letters to describe Currency, products, platforms, regions etc. E.g., EUR, GBP, CR, MF, EU, ROW, etc.
- Refer to this playground to know how we do it in Deriv.

Learn more through the Figma guide:

- Article: Name and organize components
- Article: Rename Layers
- Article: Guide: Layer naming conventions (future plan)

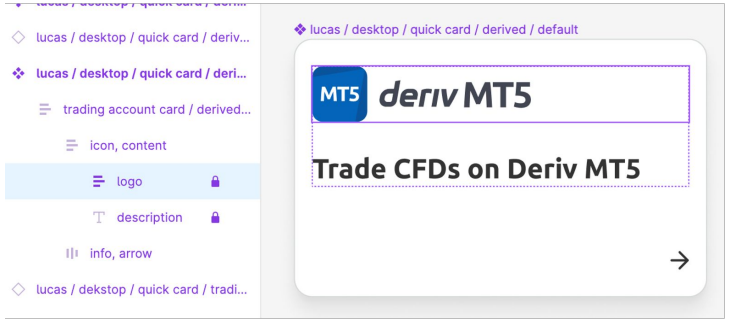


Why lock layers in components?

Lock the layer(s) of a component that is not intend for edit to avoid direct overwrite changes in layers of a component.

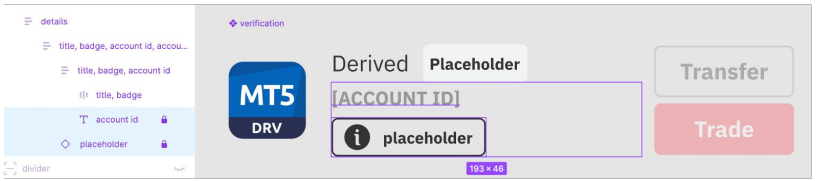
Example 1 – Derived MT5 account quick card:

This card itself is meant for the Derived MT5 account. Changing the logo or the description is wrong hence they are both locked.



Example 2 – Derived MT5 account verification state:

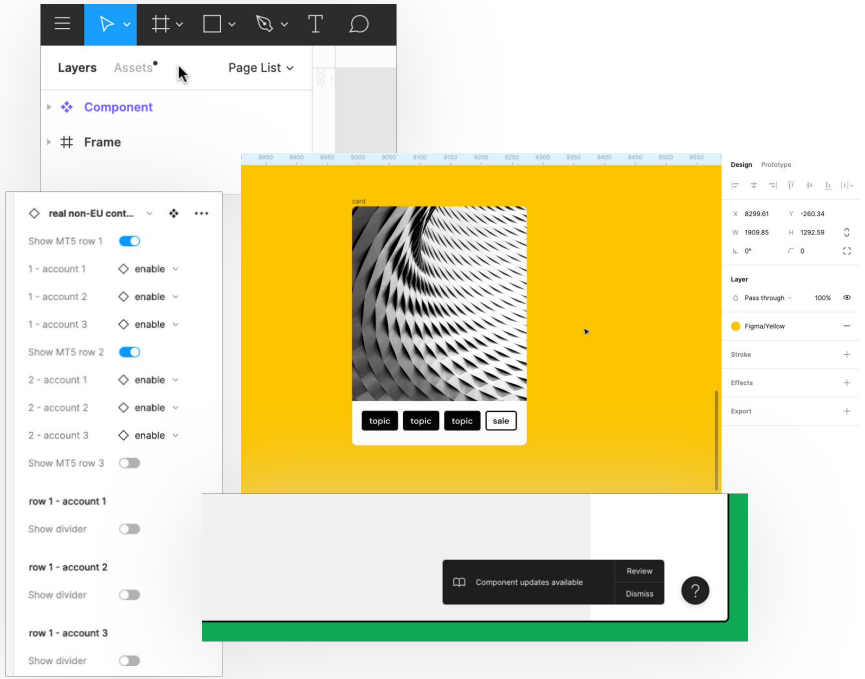
This is how the MT5 Derived account will look when in the “verification” state. The account ID layer is locked and text properties have been applied. The Placeholder layer is locked, and instant swap properties have been applied.



What is the result I should see?

If you do it right, then other designers and you will be able to:

- Find components easily from the asset panel as a result of proper naming when building flow in the release file
- Adjust component properties from the design tab (right sidebar) instead of applying changes directly on a component
- Resize a component, and it grows to fill or shrinks to fit
- Pull updates from the UI library quickly and easily
- Get updates UI library reflected on the release file without the need to reset overrides components

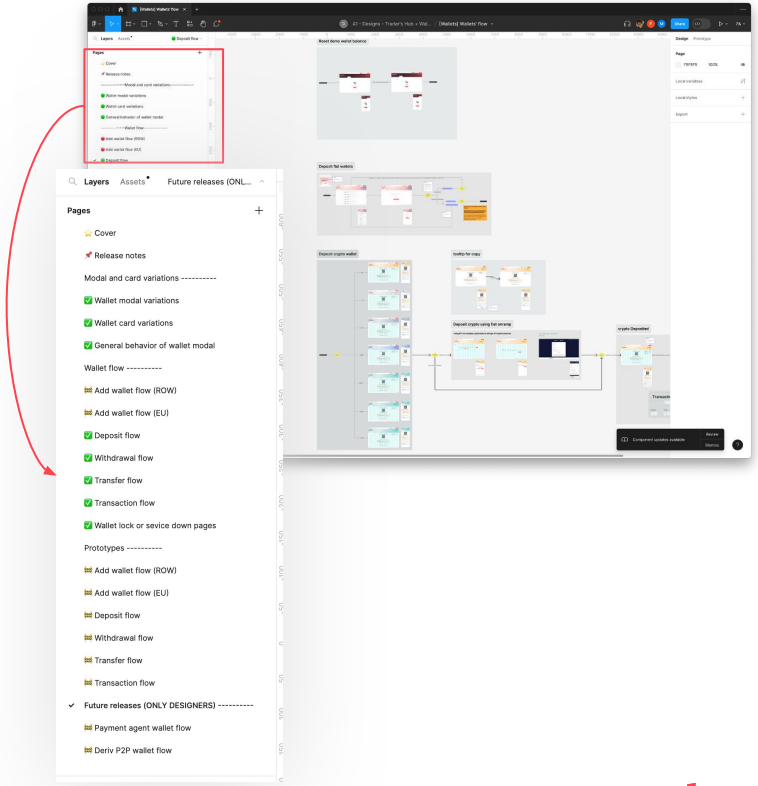


Prepare pages with the proper structure

Pages with proper structure

Each design file must always have the same structure on the pages following the specific order as below:

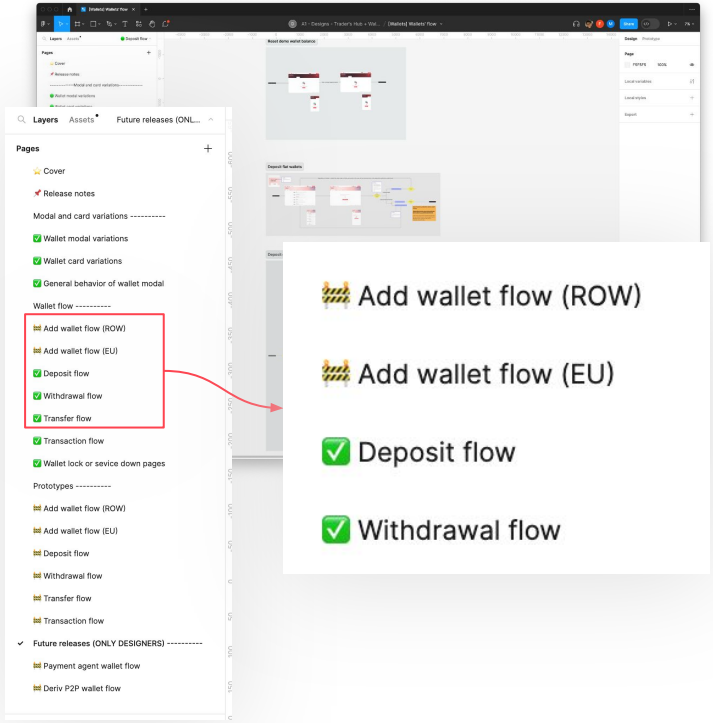
1. ⭐ Cover: The cover design you see in the folder.
2. 📌 Release notes: The release notes that inform the changes over time.
3. Name ----------: The divider that tells what the category of the following pages belongs to.
4. Prototypes ----------: pages below this divider are where we kept the prototypes.
5. Future releases (ONLY DESIGNERS) ----------: pages below this divider are where we kept designs for future releases and not to be shared with any developers.



Use emoji on pages

Indicate the completion of design

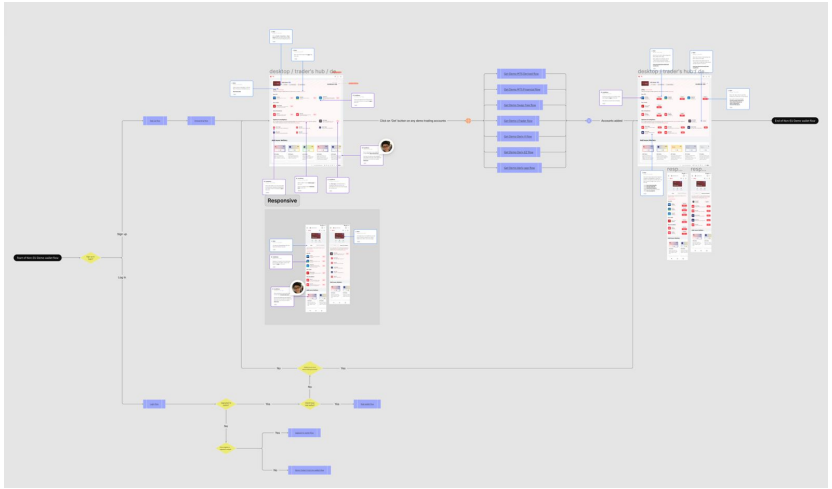
1. Pages must come with an indication of completeness as per the image on the right.
2. Designers purely use these indications rather than to be used to communicate with developers.
3. There are 2 types of indications:
    a. The ✅(Design ready icon): Everything on this page is done, including the design of each screen, the flow, and the instruction guides, for which no further changes are required.
    b. The 🚧(construction icon): some things are not done.



Build a user flow

Flow navigation

- Each page contains a specific flow of a platform.
- We use the ISO flowchart standard to build a flow.
- Below are the standard symbols we use:



Start of Non-EU Demo wallet flow

“Terminator” symbol: Indicates the beginning and ending of a process.

Sign up or login?

“Decision” symbol: Shows a conditional operation that determines which one of the two paths the process will take. The operation is commonly a yes/no question.

Sign up flow

“Predefine process” symbol: Shows named process which is defined elsewhere. It is represented as a rectangle with double-struck vertical edges. Click to further explore the process on another page or design file.

“Or” symbol: Indicates that the process flow continues in three or more branches.

“Summoning junction” symbol: Used to converge multiple branches back into a single process.

After delay (300ms)

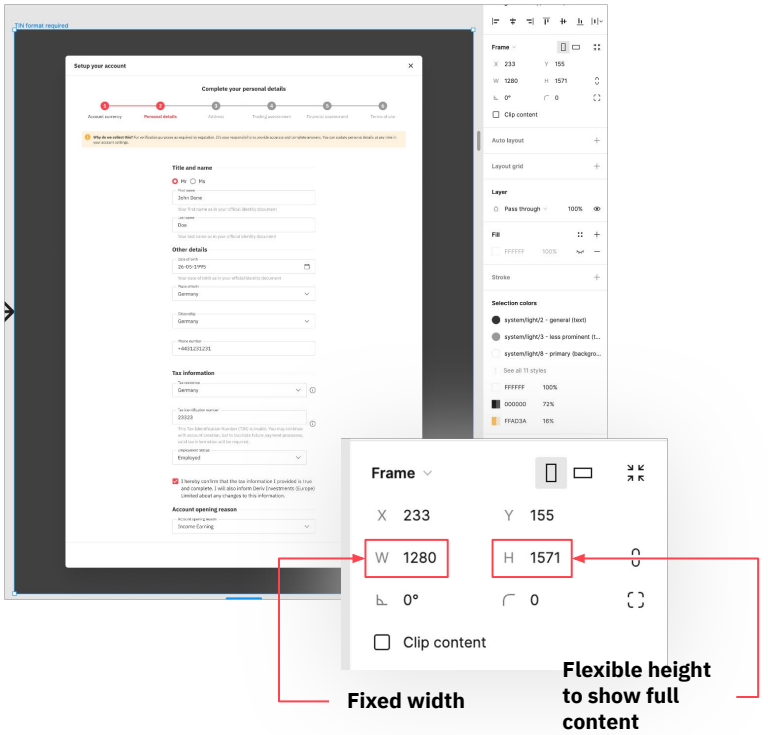
“Delay” symbol: Represent any delay periods that will be part of the process.

Navigate back to Deriv GO

“Outside control” symbol: Represent any process outside of Deriv environment that require users to operate.

Do not fix height

- Make sure each design shows its full content:
- Do not fix the height of a design frame
- Fixed the width to 1280px but show the design outside the frame if necessary.



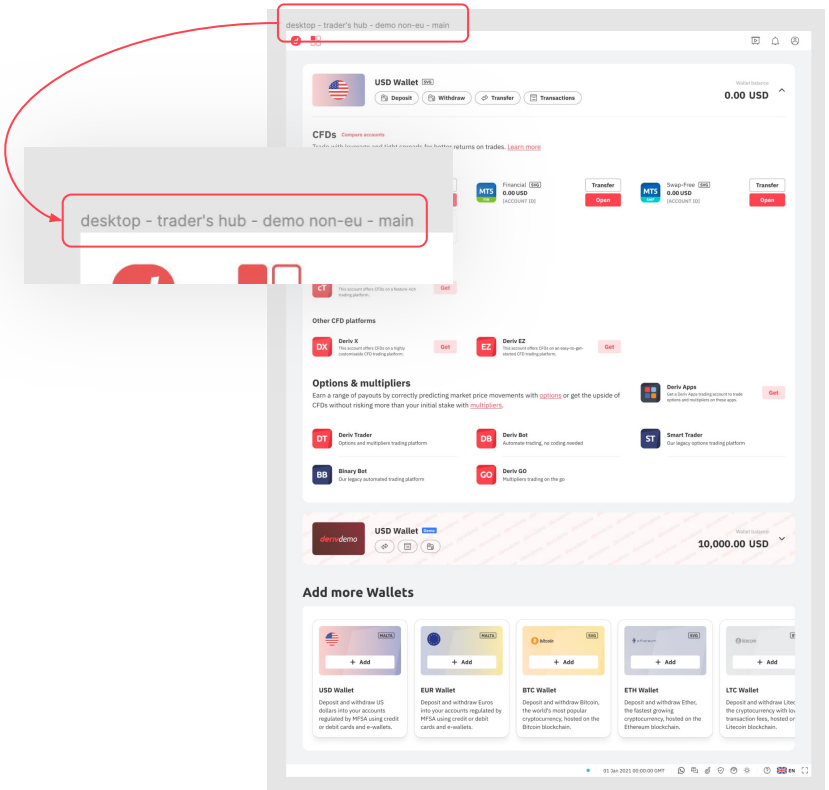
Name screens properly

Proper naming on screen helps with us to understand the flow better. Here are the rules we should follow:

1. Use slash “-” with space on both side. E.g., desktop - trader’s hub - real row - poi poa failed
2. Only use capital letters to describe Currency, products, platforms, regions etc. E.g., EUR, GBP, CR, MF, EU, ROW, etc.
3. Refer to this playground to know how we do it in Deriv.

Learn more through the Figma guide:

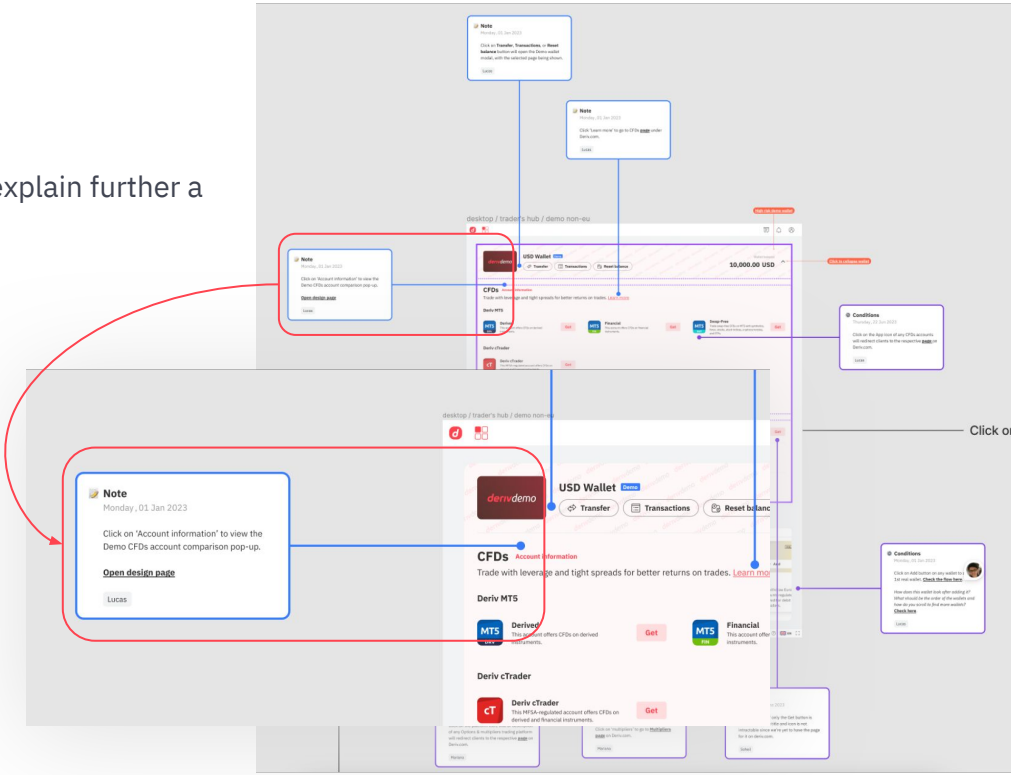
- Article: Rename Layers



Build instruction guide

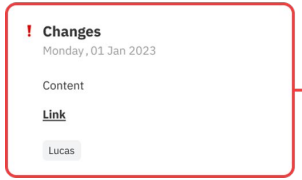
Instruction guide

- There are instruction guides on some designs which explain further a particular element, such as a button.
- There are 4 types of notes:
  1. New
  2. Changes
  3. Conditions
  4. Note

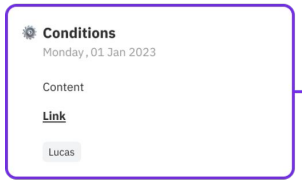


Instruction guide (Cont.)

Indicate new elements or flow.



Indicate elements or flow that have changed since the last version.

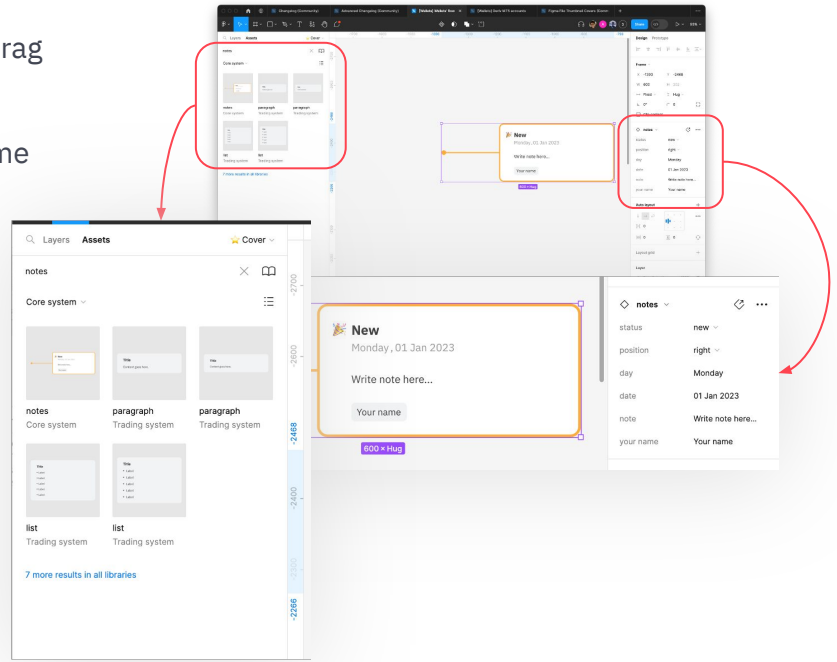


Indicate elements or flow that will change depending on user actions.

To further explain an element or flow for a better context.

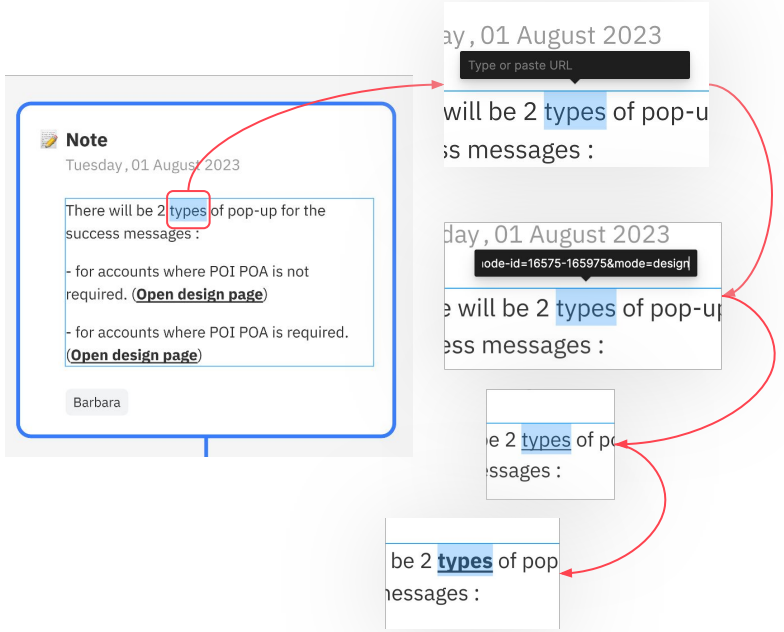
Adding Instruction guide

- You can search for “Instruction guide” from the assets panel and drag it into your work area to start using it.
- You can also change the type of the instruction guide, position, name etc, from the right panel.



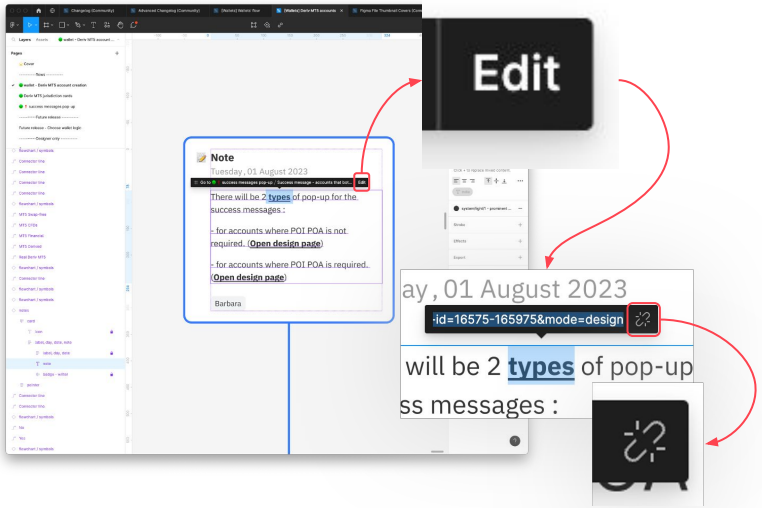
Attach link to Instruction guide

1. You can attach a link to a selected sentence by:
    a. Clicking the “Create link” button on the toolbar
    b. Press “Cmd+K” on your Macbook.
2. You should see an insert field show up. Paste the link into it and press enter.
3. The highlighted sentence will be underlined. It indicates a link has been attached to it.
4. Lastly, with the sentence highlighted, press “Cmd+B” on your keyboard to bold it.



Detach link to Instruction guide

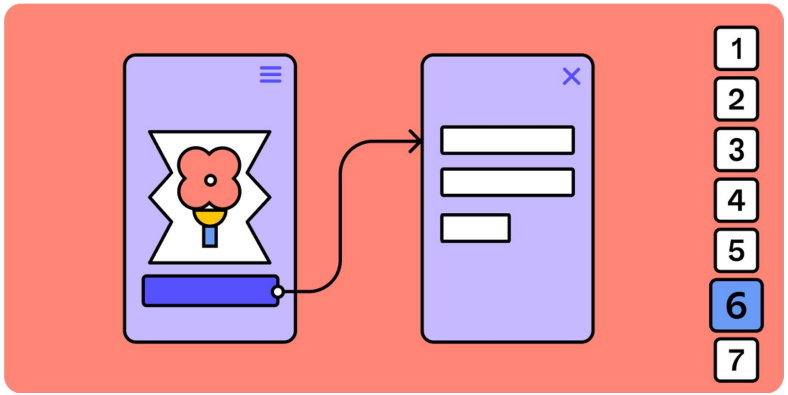
1. To detach a link, hover and hold for 1 second over the sentence with a link.
2. A tooltip will show up with an “Edit’ button at the end. Click on the “Edit” button.
3. You can now edit the link or remove it by clicking on the “unchain” icon.



Build prototype

Purpose of prototype

- Figma’s prototype features allow you to create interactive flows that explore how a user may interact with your designs.
- It is a fantastic way to preview interactions and user flows. Share and iterate on ideas.
- Building a prototype close to how we want the website to be is possible by learning through the video tutorial below in the exact order:
    - Tutorials: Prototype while you design
    - Figma for Education: Prototyping with Figma 101
    - Figma tutorial: Prototype with variables (new features)
    - Figma tip: Sections and prototyping (new features)
    - Figma for Edu: Advanced prototyping workshop
    - Extra article: Guide to prototyping in Figma



Preparation for the “Hand-Shake”

- **Dev-Ready Marking**
  Mark designs as Dev-Ready for developers to help them translate designs into code, faster.

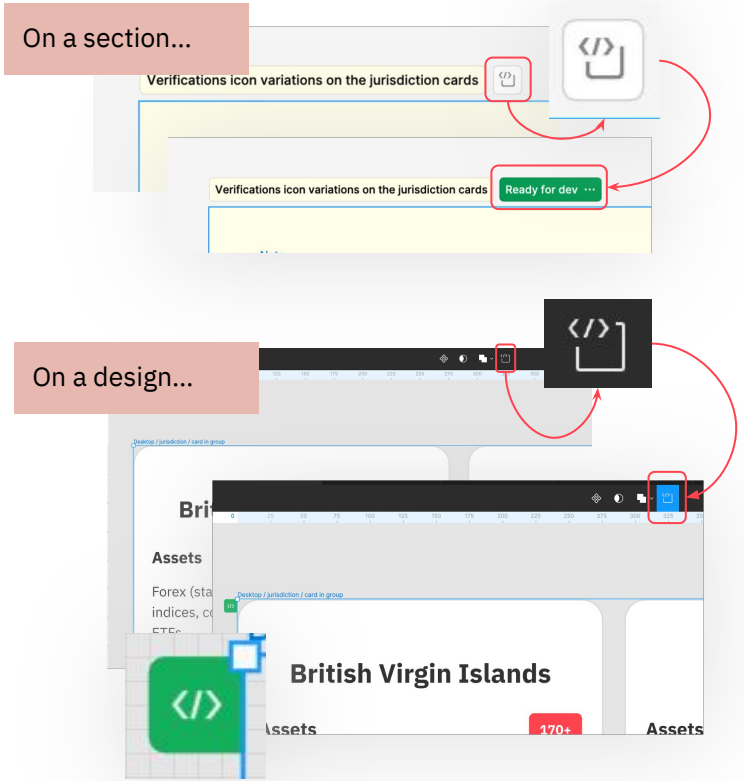
- **Release Notes**
  Summarising changes, new features, improvements, and fixes.

- **Version numbers**
  Maintaining different versions of a design file with a unique number to keep track of changes made to the design over time.

- **Design File Cover**
  Updating thumbnail that shows a preview of a file's contents wherever the file appears in the file browser.

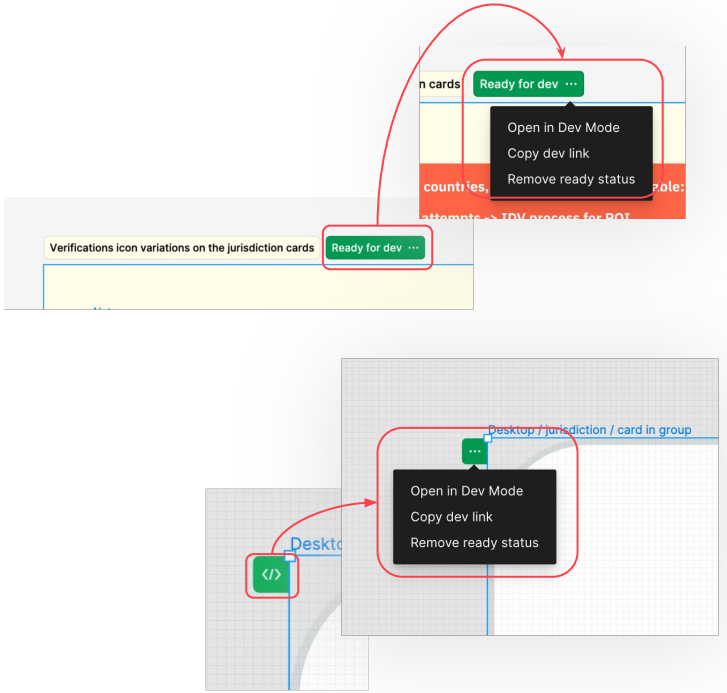
Dev-ready marking on sections or designs

1. Once designs are completed in a branch, mark sections or specific designs with “Dev ready” status to indicate designs requiring engineers to make changes.
2. To mark the status on a section, select the section and click on the ‘Mark as ready for dev” right after the section name.
3. To mark the status on a design, select the design and click on the ‘Mark as ready for dev” on the toolbar.
4. If a green marking with “Ready for dev” or “</>” shows up, it means you have successfully marked a section or a design to make it ”Dev Ready”.



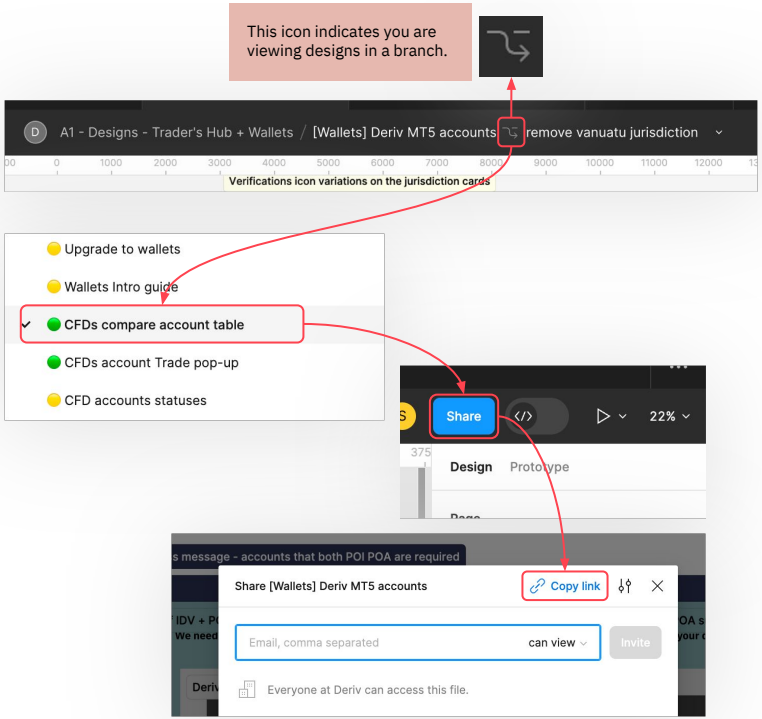
Copy Dev-Ready links to specific changes

1. On sections or designs that are “Dev-Ready”, click on the “Ready for Dev” badge, then click “Copy dev link” to add this link to a clickup card.
2. A developer will click on these links and see precisely the section or design they need to work on.



Copy link of specific flow

1. Look for the flow you want to share on a branch, click on it to view it, and click the “Share” button on the top navigation menu.
2. Click on the “Copy link” button to copy the link.
3. When someone clicks on the link, it will open this branch and default to the flow you have chosen to share.



Preparation for the “Hand-Shake”

- **Dev-Ready Marking**
  Mark designs as Dev-Ready for developers to help them translate designs into code, faster.

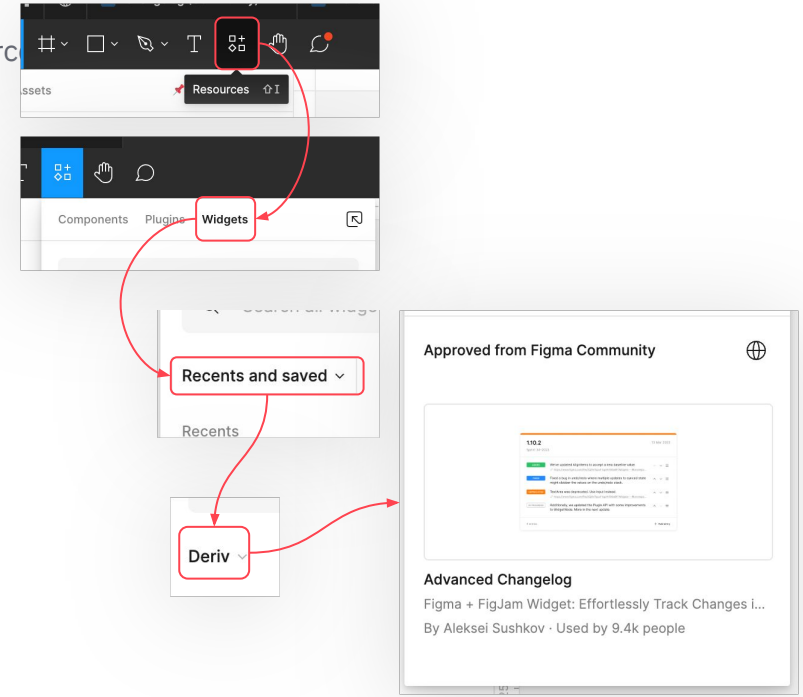
- **Release Notes**
  Summarising changes, new features, improvements, and fixes.

- **Version numbers**
  Maintaining different versions of a design file with a unique number to keep track of changes made to the design over time.

- **Design File Cover**
  Updating thumbnail that shows a preview of a file's contents wherever the file appears in the file browser.

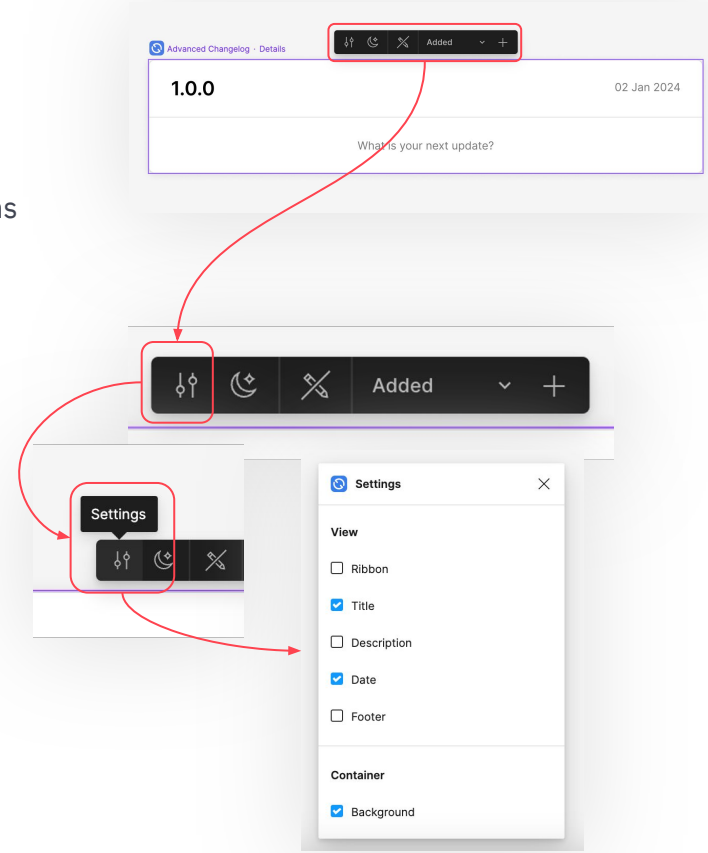
Create Release note for new file

1. To create the first release note for a new design file, click the resourc button on the toolbar and switch to the “Widgets” tab.
2. Switch from “Recent and saved” to “Deriv”, then look for a widget called “Advanced Changelog” under “Approved from Figma Community”.
3. Click on the widget and it will be added into you design space.



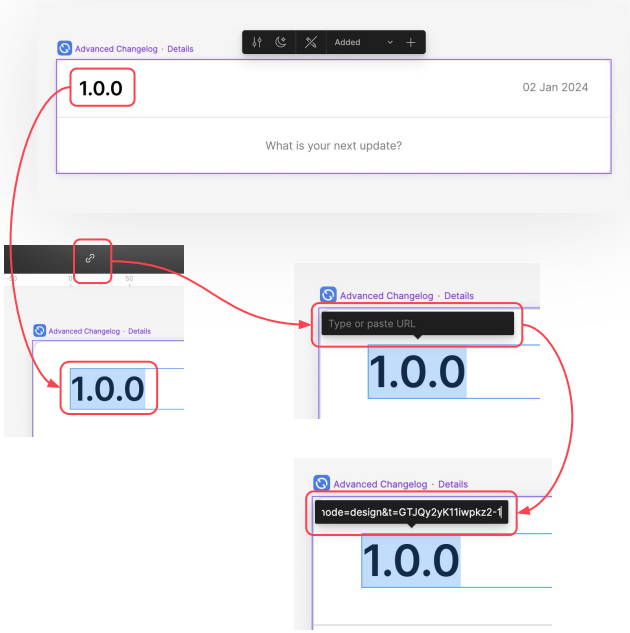
Setup release notes

- A floating toolbar will be shown if the widget is selected.
- From the floating toolbar, always check and make sure these 3 items from below are unchecked from the Settings:
  - Ribbon
  - Description
  - Footer



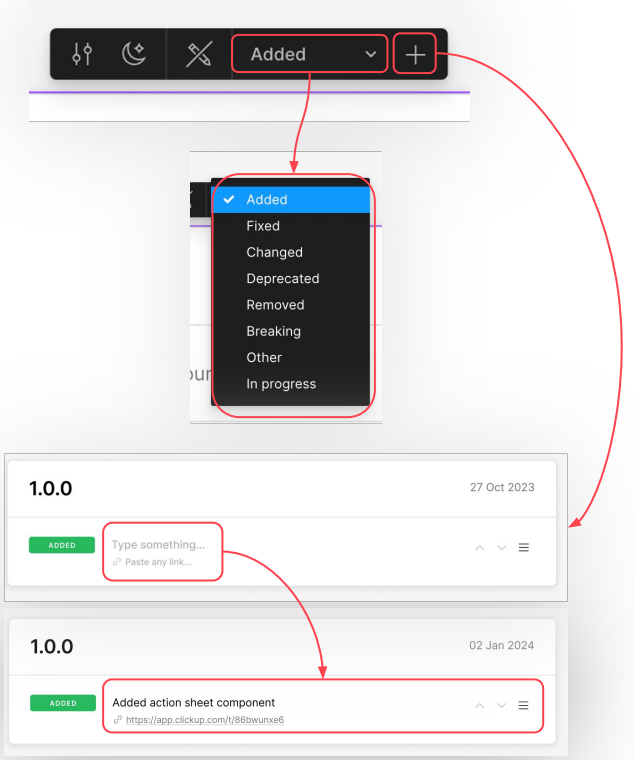
Update version number

- Make sure the version number is correct.
- The Figma link of this specific branch should be attached to it.
- you can attach a link by:
  - Highlight the version number
  - Click the “Create link” button on the toolbar or press “Cmd+K” on your Macbook.
  - An insert field show up. Paste the link into it and press enter.
  - Finally, click on anywhere outside the field to keep the changes.



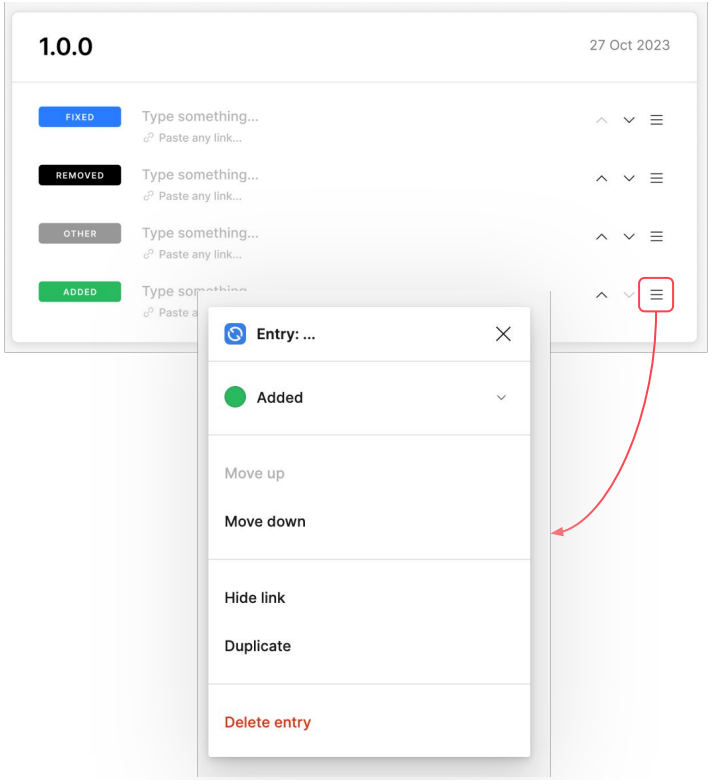
Add entry to release notes

- The “Added” dropdown on the floating toolbar allows you to set a default status to an entry every time you add an update.
- You can add lists of designs you have done by clicking the “+” button on the floating toolbar.
- With an entry registered, write a short description to explain what you did, and add the ClickUp card link to the “Paste any link...” field.
- To edit the link, click on the description field first to bring out the text edit mode. Then, click on the link. Now you can edit the link.



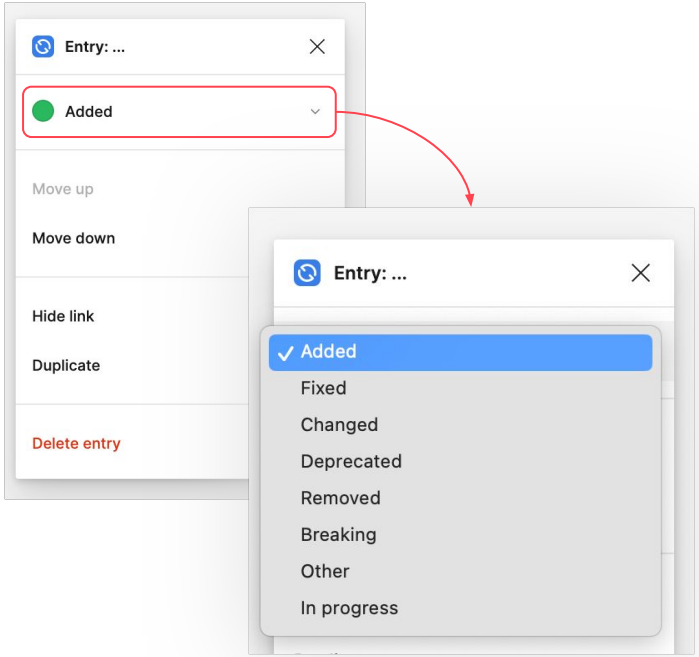
Manage entries

- To manage an entry, click the Hamburger menu on the entry you want to make changes.
- You can then change the status, move the entry up or down, hide the link, duplicate it, or delete it.



There are 8 type of entry statuses

- **Added**
  A newly added component or design
- **Fixed**
  Fix mistake spacing, font sizes, colour, etc
- **Changed**
  The design changed over an existing component or design
- **Deprecated**
  A component is no longer supported, but usage is allowed (Design system team only)
- **Removed**
  An element has been removed
- **Breaking**
  A major change has been made, and the major number of this release's version number must be changed
- **Other**
  A component has been moved to other places (Design system team only)
- **In progress**
  Do not use



Preparation for the “Hand-Shake”

- **Dev-Ready Marking**
  Mark designs as Dev-Ready for developers to help them translate designs into code, faster.

- **Release Notes**
  Summarising changes, new features, improvements, and fixes.

- **Version numbers**
  Maintaining different versions of a design file with a unique number to keep track of changes made to the design over time.

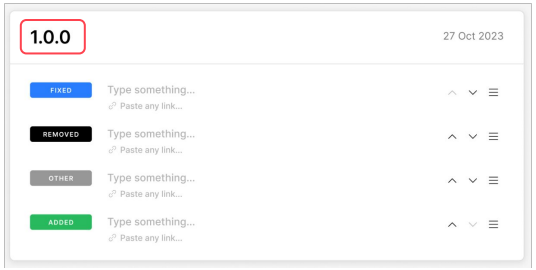
- **Design File Cover**
  Updating thumbnail that shows a preview of a file's contents wherever the file appears in the file browser.

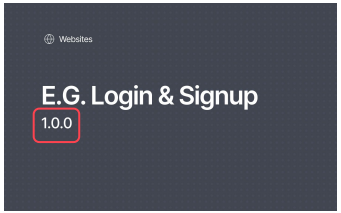
Update version numbers

There are 3 places where you need to update the version numbers:

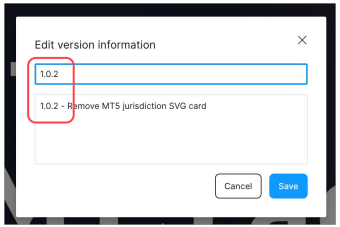
- Release note
- Design file cover (Refer to Update Design File Cover section)
- Branch title and description after merge (Done by Team Lead only)

Release note





Design file cover



Branch title and description after merge (Done by Team Lead only)

How design team do versioning?

- The **Major version** number can be incremented when significant changes to the overall flow and UI may not be backwards-compatible with previous versions.
  Flow changes: *Complete overhaul of the flow (e.g., wallet project)*
  UI changes: *Drastic visual changes on UI (e.g., rebranding)*

- The **Minor version** number can be incremented when new features or components are added/removed to the flow that is backwards-compatible with previous versions.
  Flow changes: *Adding/removing new steps but not entirely different (e.g., MT5 Vanuatu account can now use IDV)*
  UI changes: *Select another instance for specific components only on a few screens.*

- The **Patch version** number can be incremented when small improvements are made to the design that does not affect compatibility.
  Exclusively UI changes: *on content, fixing a typo, color update, adjusting spacing, deleting or changing small elements such as icons, dividers, which does not affect the flow at any level.*

How design system team do versioning?

- The **Major version** number can be incremented when significant changes to the UI redesign or big changes to the overall design may not be backwards-compatible with previous versions.

  Example: implementing design tokens

- The **Minor version** number can be incremented when components are improved that are backwards-compatible with previous versions.

  Example: adding new properties, variants, or minor changes on the button's border-radius and input field's label size)

- The **Patch version** number can be incremented when bugs are fixed, or small improvements are made to the design that does not affect compatibility.

  Example: wrong font size or property is not working.

Preparation for the “Hand-Shake”

- **Dev-Ready Marking**
  Mark designs as Dev-Ready for developers to help them translate designs into code, faster.

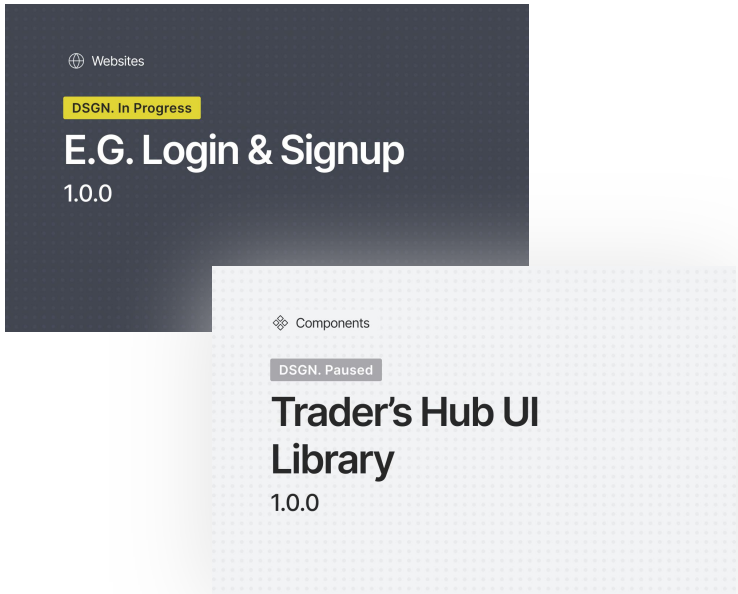
- **Release Notes**
  Summarising changes, new features, improvements, and fixes.

- **Version numbers**
  Maintaining different versions of a design file with a unique number to keep track of changes made to the design over time.

- **Design File Cover**
  Updating thumbnail that shows a preview of a file's contents wherever the file appears in the file browser.

How to add and types of design file cover

To add a cover, search for “Cover” from the “Core” or “Quill” designs system. On the properties panel that is located on your right, you will see a list of change you able to make, the first on the list is the **File Type.**



There are 2 file types of cover you can choose from:

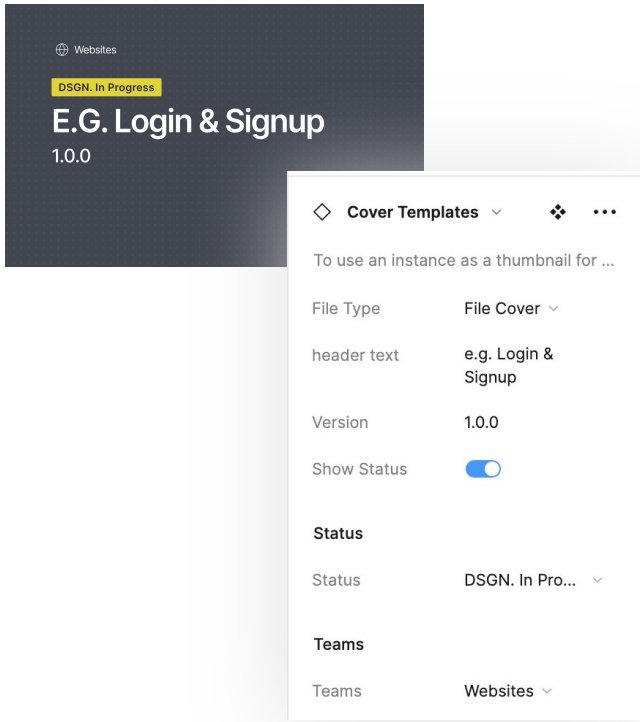
1. **File Cover (for design files) in black:**
   Design file cover that is for any design file.

2. **Library Cover (For design system file and UI Library file) in gray**
   Library file cover that is for design system file and UI Library file.

Maintain a Design File Cover

With **File Cover (for design files) in black** selected, the rest of the properties is shown as below:

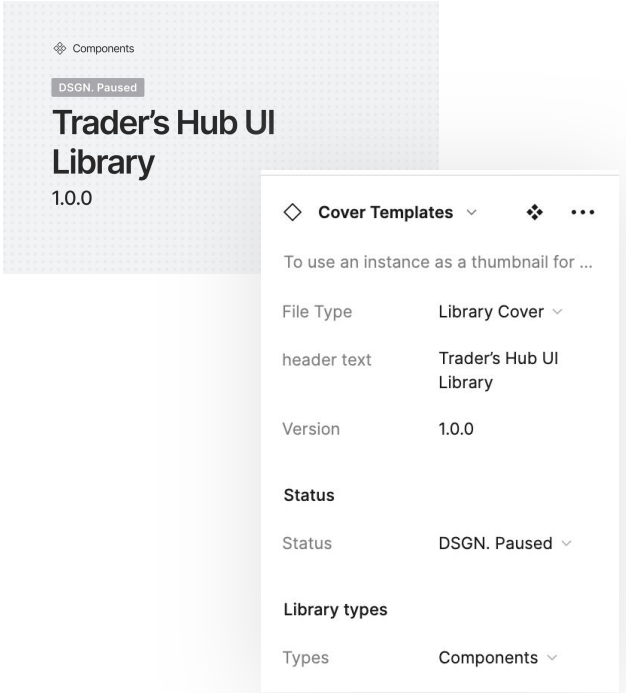
- **Header text:** Add the title of this file
- **Version:** Update the version of this file
- **Show status:** Toggle this off so the status is not show. This must be done before merging a branch.
- **Status:** Change status of the design file between “DSGN. In Progress”, “DSGN. Paused”, “DSGN. Blocked”, or “ARCHIVE”.
- **Teams:** Choose the team you are report to such as “Websites”, Mobile Native, Responsive, Design system, etc.



Maintain a Library file Cover

With **Library Cover (For design system file and UI Library file) in gray** selected, There will be one more properties for you to change compare to the design file cover, which is the **Types** of the file.

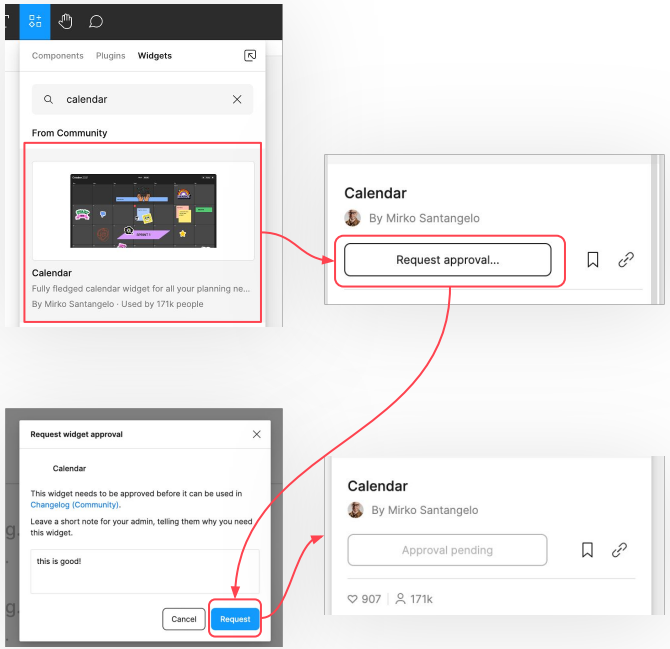
- **Types:** Choose the type of this design system file or UI Library file. It can be “Master”, “Guidelines”, “Components”, “Patterns”, “Working file”, “Templates”, or “Foundation”.



Plugins and widgets approval

About plugins and widgets approval

- Plugins and widgets are third-party applications that run in Figma design or FigJam.
- Since plugins and widgets have access to information about members and files, designers can only use plugins and widgets that have been approved by an organization admin.
- You can request approval when you find plugins or widgets in the Figma Community you want to use.



Request for branch review

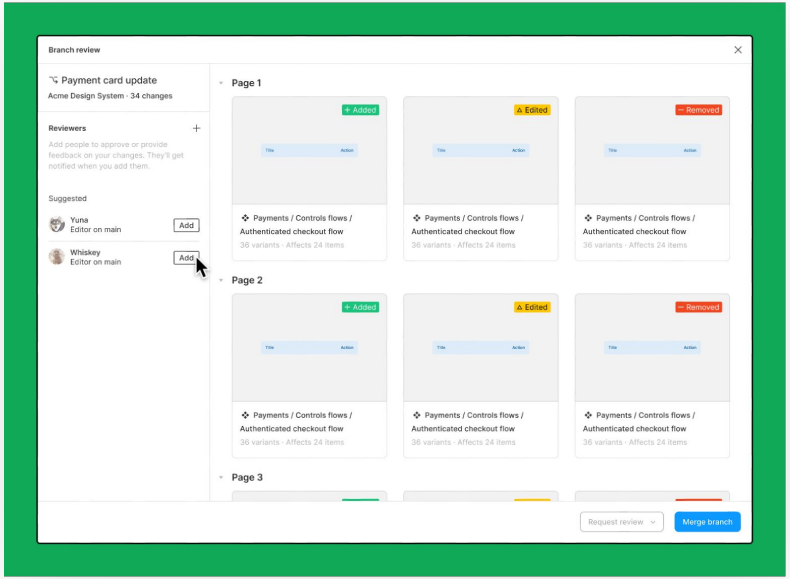
Checklist before review request

Before requesting a review, please make sure:

1. All required designs and flow are completed.
2. Instruction guides have been added.
3. Designs are marked with “Dev-ready”.
4. Release notes have been updated.
5. Prototypes are completed if required.
6. The file cover has been updated to the version of this branch.

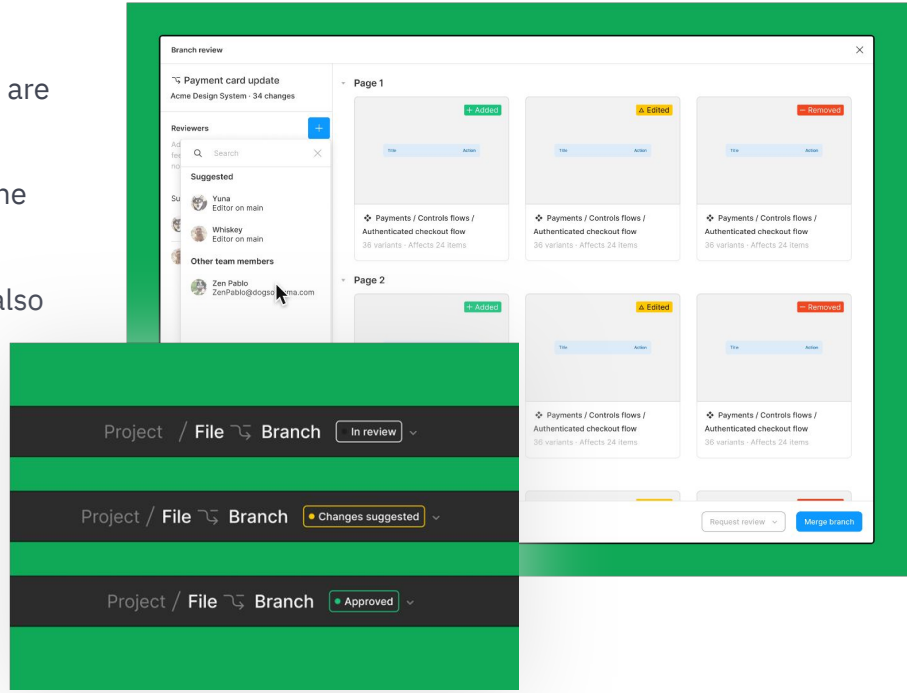
To find out how, please refer to the article and video guide below:

- Article: Request review
- Video guide



Who should I get review and approval from?

- Request a branch review from your team lead
- He/she will leave feedback, help and guide you if the changes are required.
- He/she will approve the changes and decide when to merge the branch and merge it him/herself.
- This process applies to branches belonging to a product and also branches belonging to a UI library.



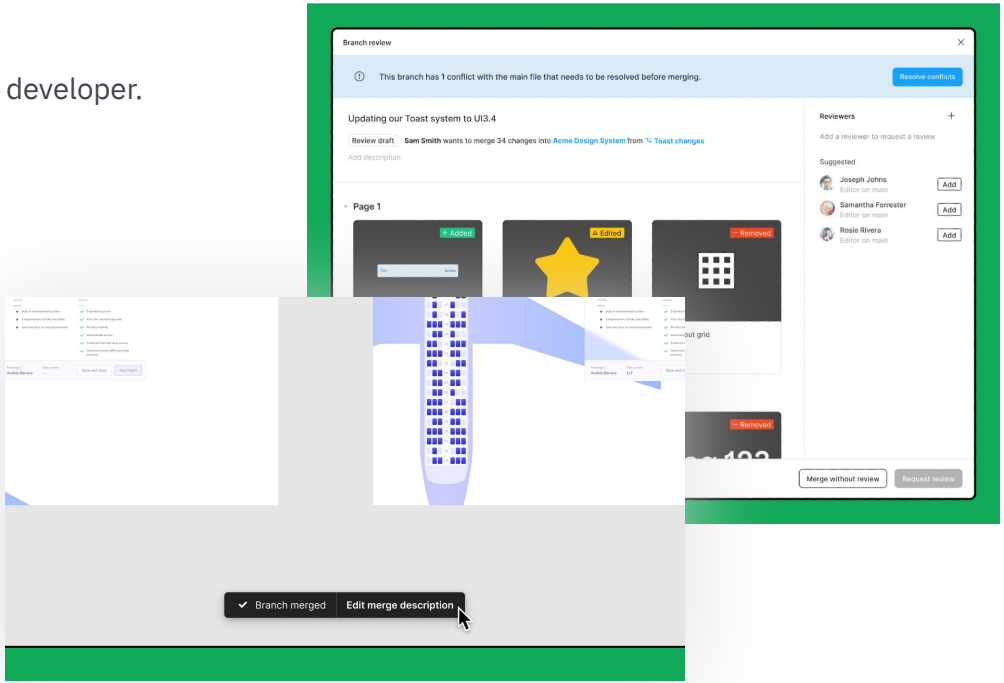
Merging a branch

Merging a branch to its main file

Your TL will merge the branch after the design is ready for the developer.

Once merged, we can re-access this branch.

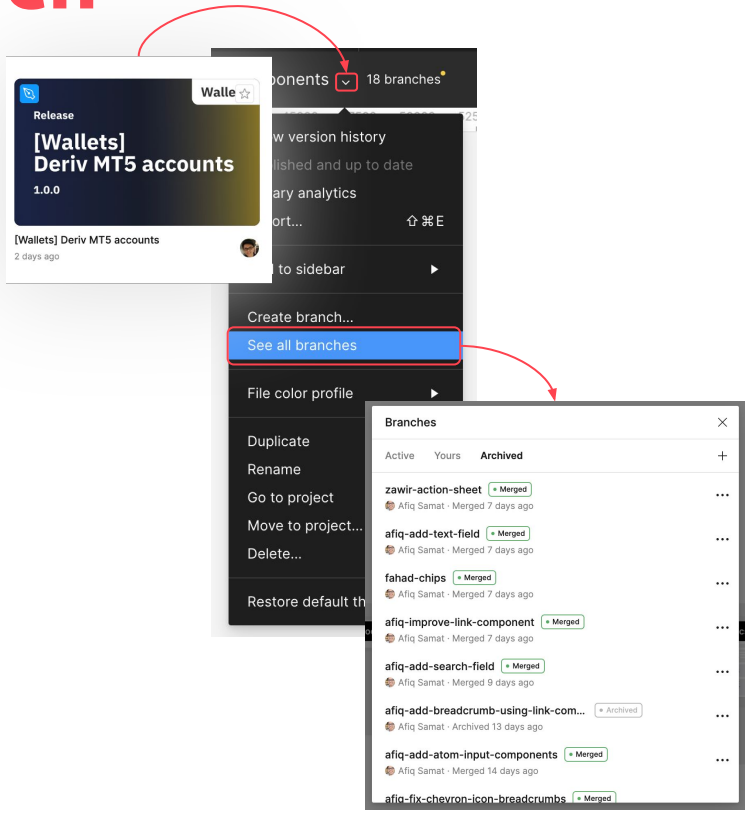
- How to merge branch
- Video guide



Access a merged branch

Access to merged branch

1. A merged branch is part of the master design file.
2. To access, open the design file you wish to explore its history.
3. Once the file is open, click the chevron right after the file name on the toolbar.
4. In the dropdown, click on “See all branches”.
5. A “Branches” pop-up window will appear.
6. Browse through branches under the “Archived” category to see all merged branches.



Thank you!