

README

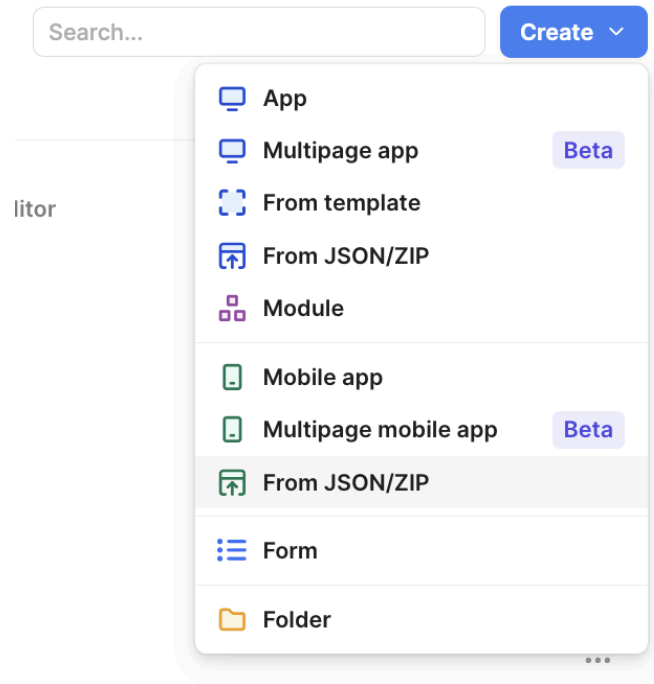
This document is now deprecated. Read our new docs:

<https://docs.retool.com/apps/guides/multipage>

Multipage Private Beta Program

TL;DR

- Thank you for being part of our Multipage Private Beta program! Any and all feedback is appreciated – **please share any feedback via [this form](#)**.
- **You can build production-ready applications with Multipage.** Our goal is to not introduce any backwards incompatible changes unless necessary. In the event that we must make one, we will provide specific instructions on how to migrate your app to be compatible with the change. More on expectations [here](#).
- When the feature flag is on for your organization, you will have the ability to create a Multipage app via this dropdown on your Home page! **Your existing apps will be untouched, so you don't need to worry about those.**



Expectations with Private Beta program

Though we are super eager to have you test out our Multipage feature, we want to set the following expectations in order to prevent any miscommunication:

- The below feature set and how it works is subject to change.
 - Though unlikely, if there is a backwards incompatible change, we will always provide a migration path for your application and ample time, documentation, and help to migrate your application.
 - We have high confidence that we've designed a great solution, but the point of the Private Beta program is to gather feedback from builders like yourselves so that we can make sure we are designing the best solution! We'll be soliciting feedback throughout this program and will take it into consideration as we iterate on Multipage.
- We encourage you to build production-ready apps with Multipage.
 - We want to see real-world use cases of multipage! To do so, we encourage building production-ready apps with Multipage.
 - Do remember that this is still a Private Beta which means that there may be bugs that we are still fixing with the feature.
- We see this as a strong partnership and would really love your feedback!
 - Not all feedback might be taken immediately (or at all), but we will surely evaluate it holistically with others' feedback to design the best solution.

Why Multipage?

Multipage solves numerous problems Retool builders have faced over the past few years:

- Performance of large apps
 - Read more about [performance boosts here](#)
- Flashing when navigating between apps
 - Since all pages live in one app, there shouldn't be flashes when navigating within the same app (if there is, please email us at multipage-team@retool.com, videos and screenshots would be very helpful!)
- Having to edit multiple apps in multiple browser tabs
- App sprawl on the home page
- Creating list detail views
- Sharing data between apps
 - Read more about [passing data here](#)

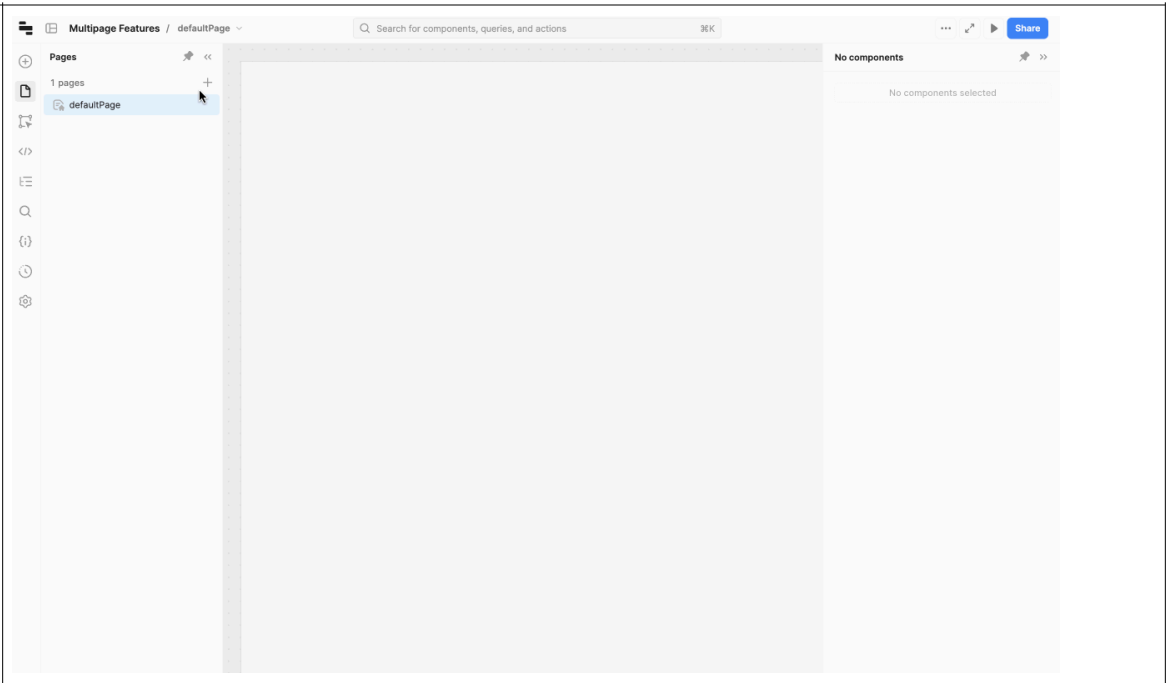
Feature Documentation

When the feature flag is on for your organization, it means that you will have the ability to create a Multipage app via this dropdown on your Home page! **Your existing apps will be untouched**, so you don't need to worry about those. You can also continue to make normal single-page apps. If you would only like to turn on this experience for specific users so that it doesn't disrupt the app building experience for all users, we can also do that for specific users, just send us the emails.

Basics

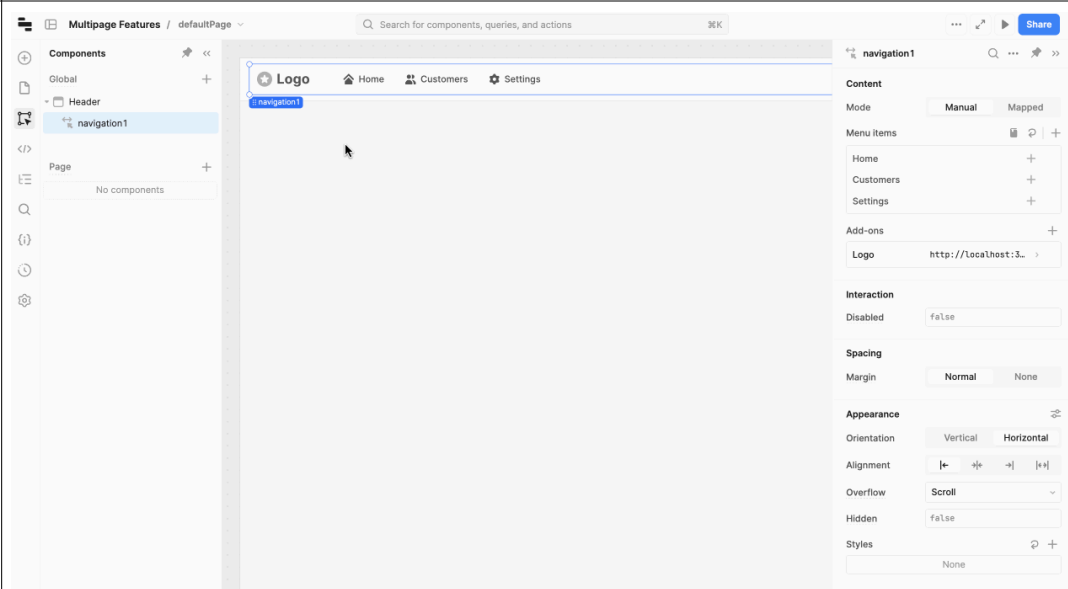
Adding pages to your app

- Ability to configure:
 - Page title
 - Page URL
 - Page URL search & hash parameters
 - Page styles
 - Default page
 - Right click a page and click *Set as default page*



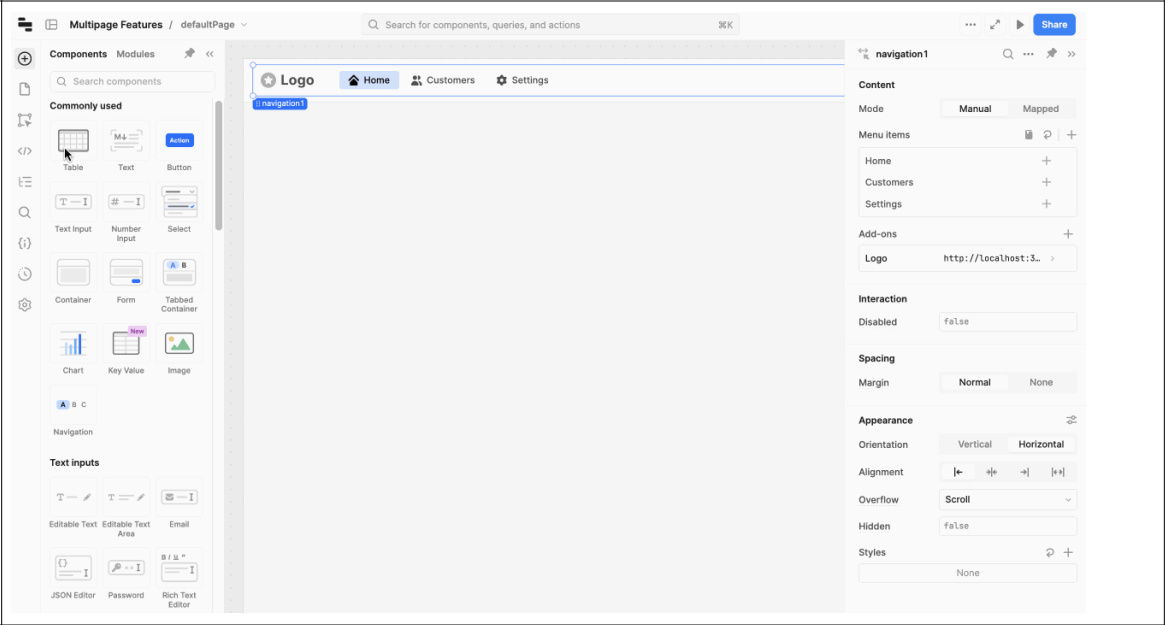
Adding navigation to your app

- Header or Sidebar with navigation
- Menu Items can be configured to go to a page
- You can add a Header/Sidebar by clicking the Global '+' icon in the Component Tree



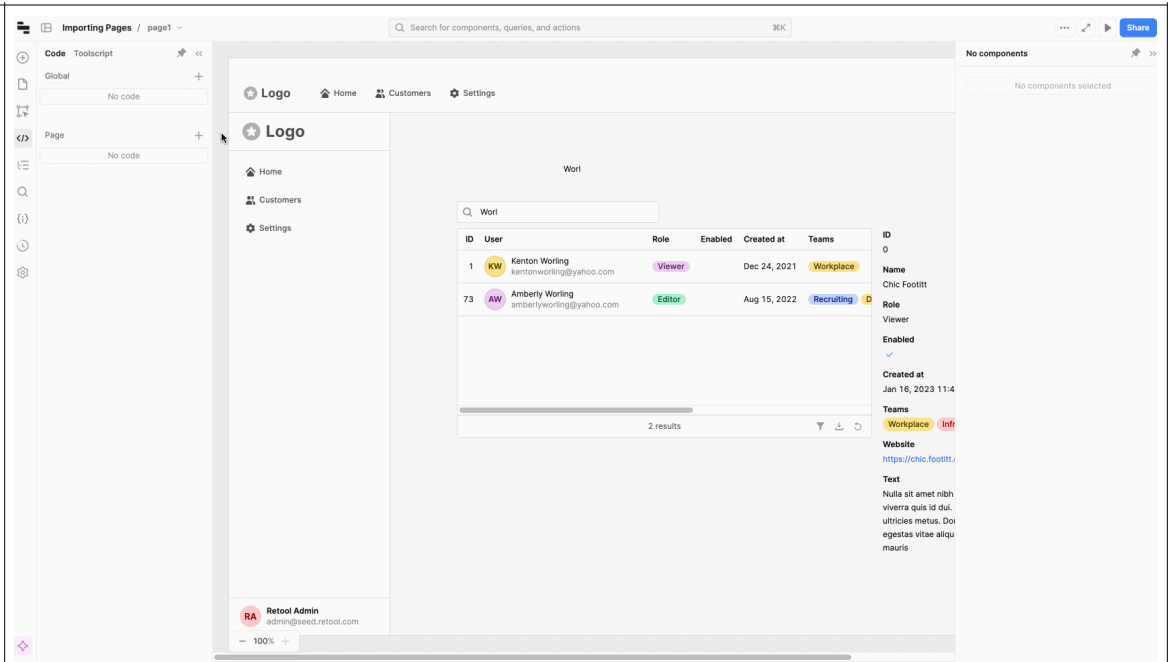
Adding components to pages

- Same as before go to "Add" tab in your IDE and drag & drop components onto the canvas!
- Components will only show up on that page, unless added to Header or Sidebar



Adding code to your pages

- Same as before, go to "Code" tab in your IDE and use the + icon to add new queries, variables, etc.
- Two types: page code & global code (learn more in the Scope section)



Scope and performance

Multipage introduces two types of scope for components and code: Global and Page Scope

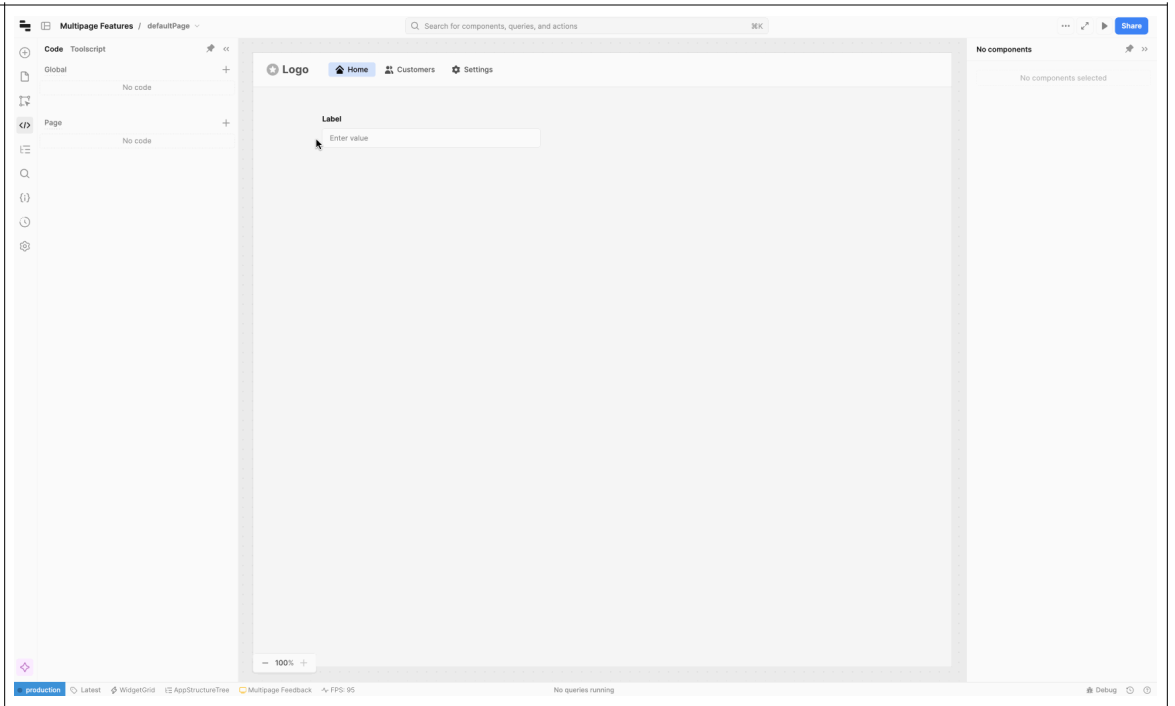
- Global scope
 - Any component or code in global scope can only reference other components/code in global scope.
 - Ex: a global query can read data from another global query, but cannot read data from a specific page's query
- Page scope
 - Any component or code in page scope can reference other components/code on its same page *or* in global scope.
 - Ex: a page query can read the data from a global variable.

Multipage is fast because scope enables deferred evaluation

- Deferred evaluation - when Retool only evaluates the code and components on the current page.
- This means one page's size won't affect another page's performance.

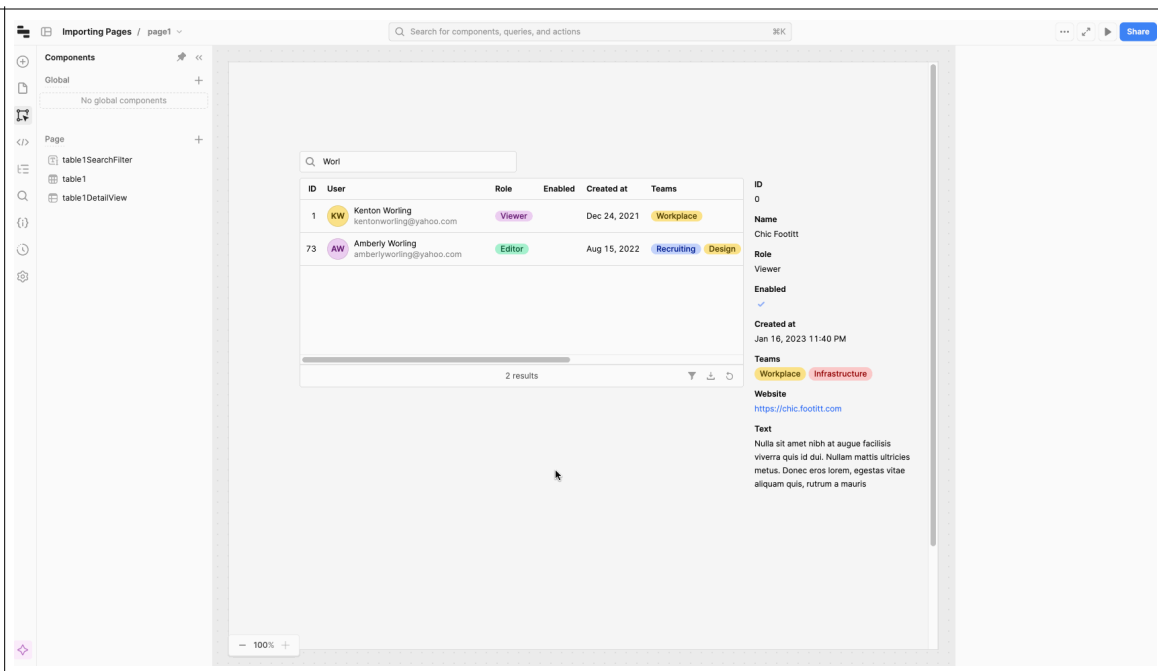
Global / Page Code

- Page code only evaluates when the page is loaded
- Global code evaluates when the app is loaded
- Global code is meant to be shared code you evaluate once and use across your pages



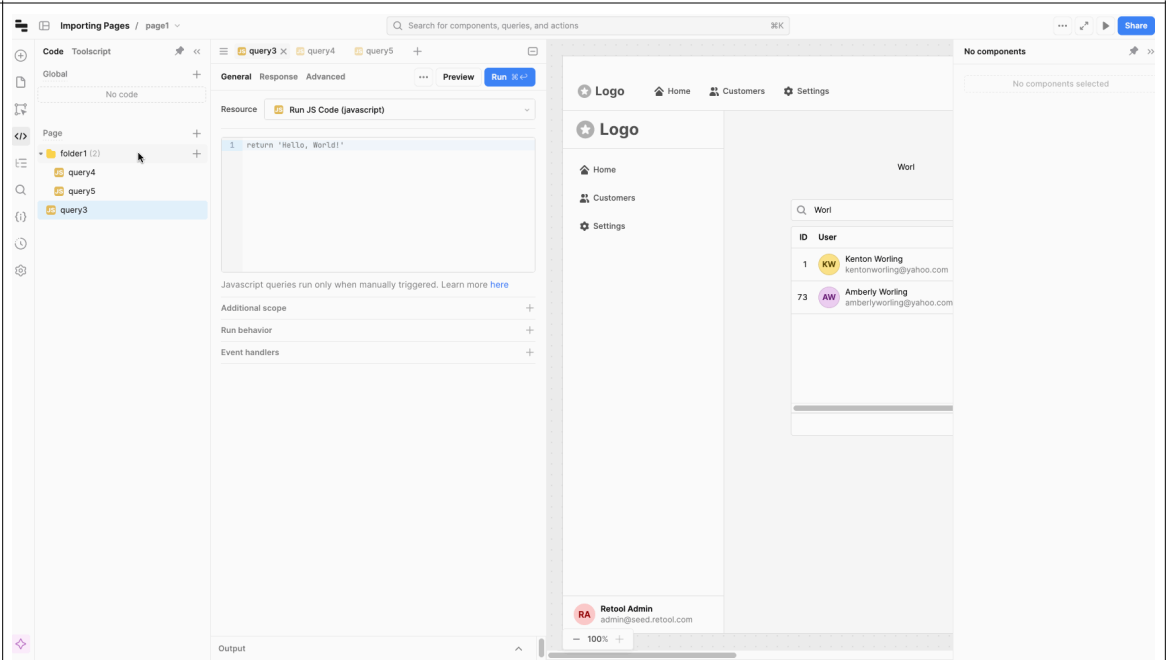
Global / Page Components

- Header and Sidebar frames are global scope. They appear on every page.
- All components on a page's main canvas is page scoped
- Modals and Drawers are page-scoped
- Drag & dropping components across the Header, Sidebar, and main canvas will change the component's scope accordingly (gif for reference)



Changing code scope

- You can drag and drop code between the Global and Page scoped trees (1st gif)
- Code context menu also has the option to move code to different pages (2nd gif)



The screenshot shows a web development IDE with a code editor on the left and a preview window on the right. The code editor displays a JavaScript query:

```
1 return 'Hello, World!'
```

The preview window shows a user list table with the following data:

ID	User
0	CF Chic Footitt chic.footitt@yahoo.com
1	KW Kenton Worling kentonworling@yahoo.com
2	EF Evelina Fender efender@outlook.com
3	LS Lexis Speers lexisspeers@icloud.com
4	KA Kenton Antonioni kenton_antonioni@icloud.com

URL Params

Configuring URL parameters

- Each page can have its own URL parameters
- URL parameters can be set up to update when component state updates
- Components on the page can reference the initial URL parameters state so they can default to that value from the start

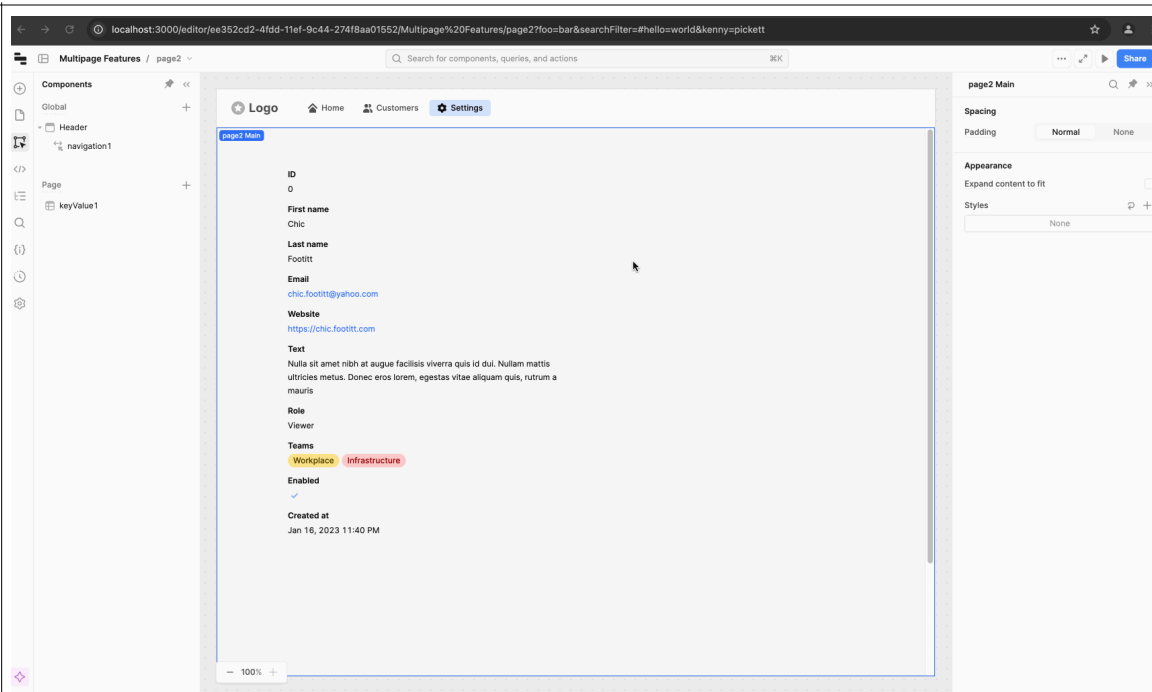
The screenshot shows a web application editor interface. The main content area displays a user management page with a table of users. The table has columns for ID, User, Role, Enabled, Created at, and Teams. The data is as follows:

ID	User	Role	Enabled	Created at	Teams
0	Chic Footitt chic.footitt@yahoo.com	Viewer	✓	Jan 16, 2023	Workplace, Infrastru
1	Kenton Worling kentonworling@yahoo.com	Viewer	✓	Dec 24, 2021	Workplace
2	Evelina Fender efender@outlook.com	Editor	✓	Jan 3, 2022	Product, Sales
3	Lexis Speers lexispeers@icloud.com	Admin	✓	Dec 13, 2022	Infrastructure, Desig
4	Kenton Antonioni kenton_antonioni@icloud.com	Admin	✓	Apr 15, 2022	Success, Recruiting

The sidebar on the left shows a list of pages: defaultPage, page1, and page2. The right sidebar shows configuration options for the defaultPage, including Page title, Page URL, URL Preview, URL search params, URL hash params, and Appearance.

Accessing URL Parameters

- Use `url` to access the search and hash params and the href



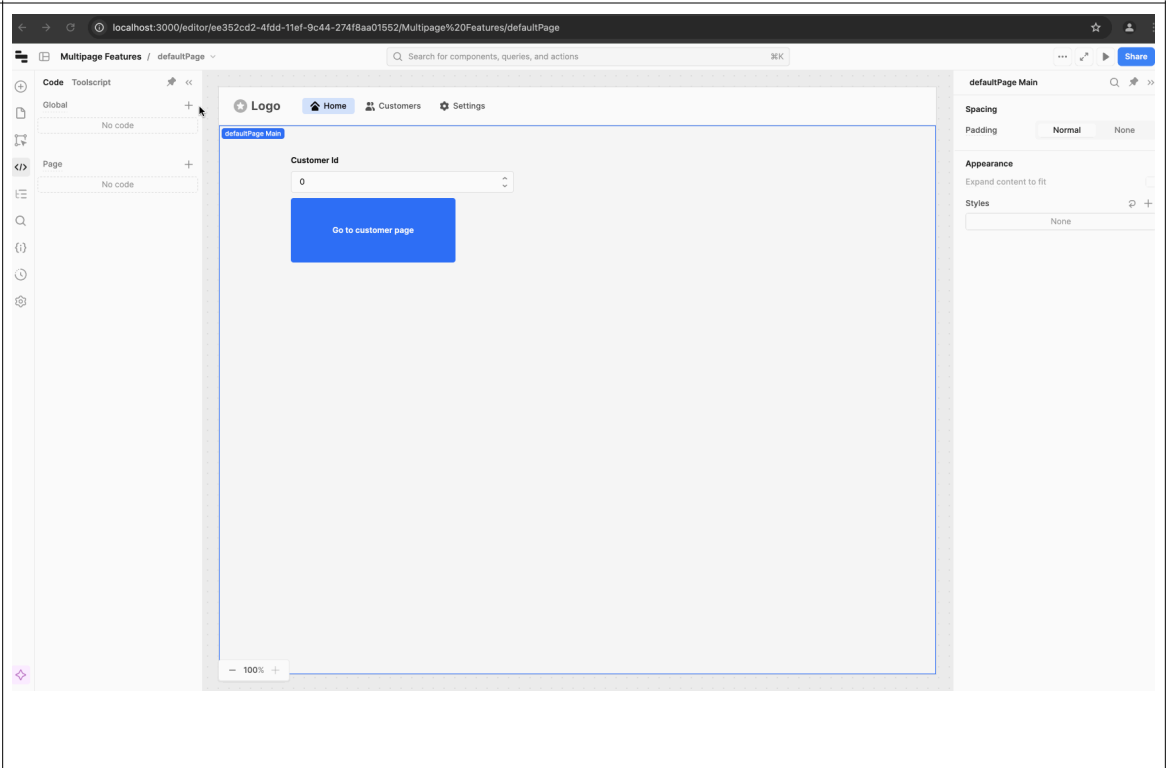
Passing data between pages

There are 3 ways you can pass data between pages:

- Using a global variable (1st gif)
- Using URL parameters (2nd gif)
- Using local storage (3rd gif)

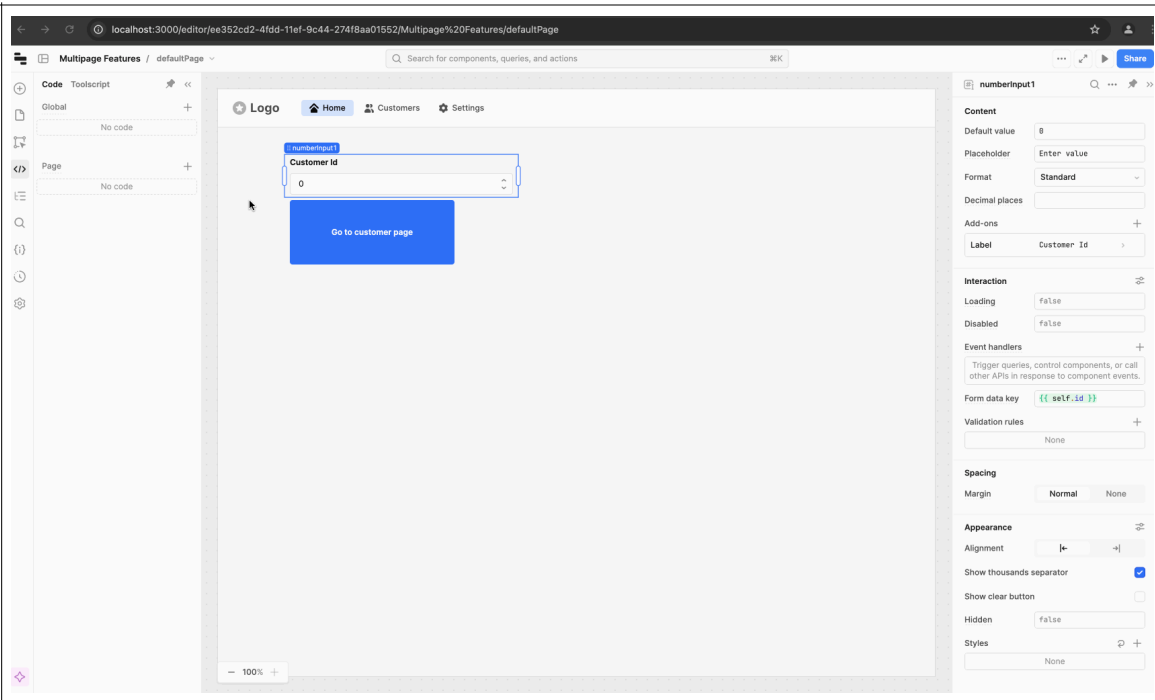
Using global variables

- We found global variables to be the best relevant for most data passing uses.
- Use the *Set variable* event handler to set its value.
- Since the variable is in global scope, it can be accessed from any page and any global component.



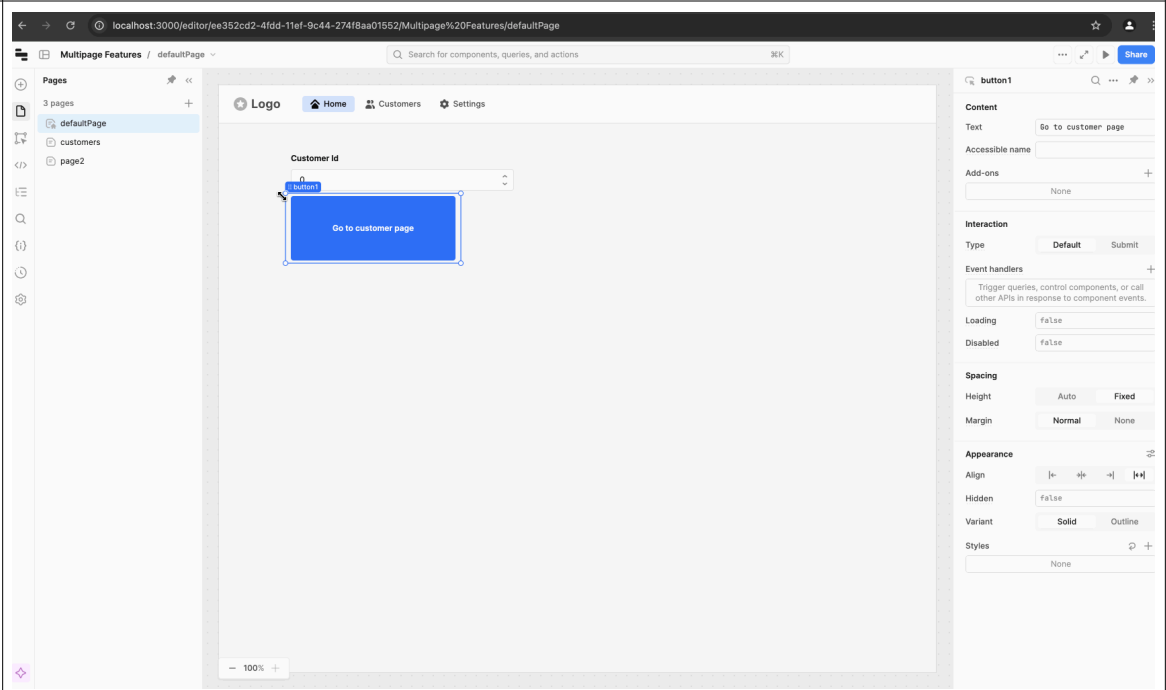
Using URL parameters

- You might want to use **URL parameters** if you want your app's state to be reflected in your URL so you can deep link and share the URL with others
- The *Go to page* event handler allows you to set the search/hash url params for the page you are navigating to.
- **Note:** You may not want to pass sensitive data via url params due to privacy concerns



Using local storage

- You might want to use **local storage** if you want to persist data over sessions
- Use the *Set local storage* and *localStorage* variable to read and write to local storage.



Migrating single page apps to multipage

Duplicating page as Multipage

- Use the application dropdown option 'Duplicate to multipage' to convert a single-page app to multipage
- All code in the original app will be page scoped.
- If the duplicated app contains a header and/or sidebar, those will also be in the new app.
- App settings in the duplicated app will also be in the new app
- Some app settings that are currently unsupported in multipage may not work in the new app. The builder needs to review and fix these if applicable.

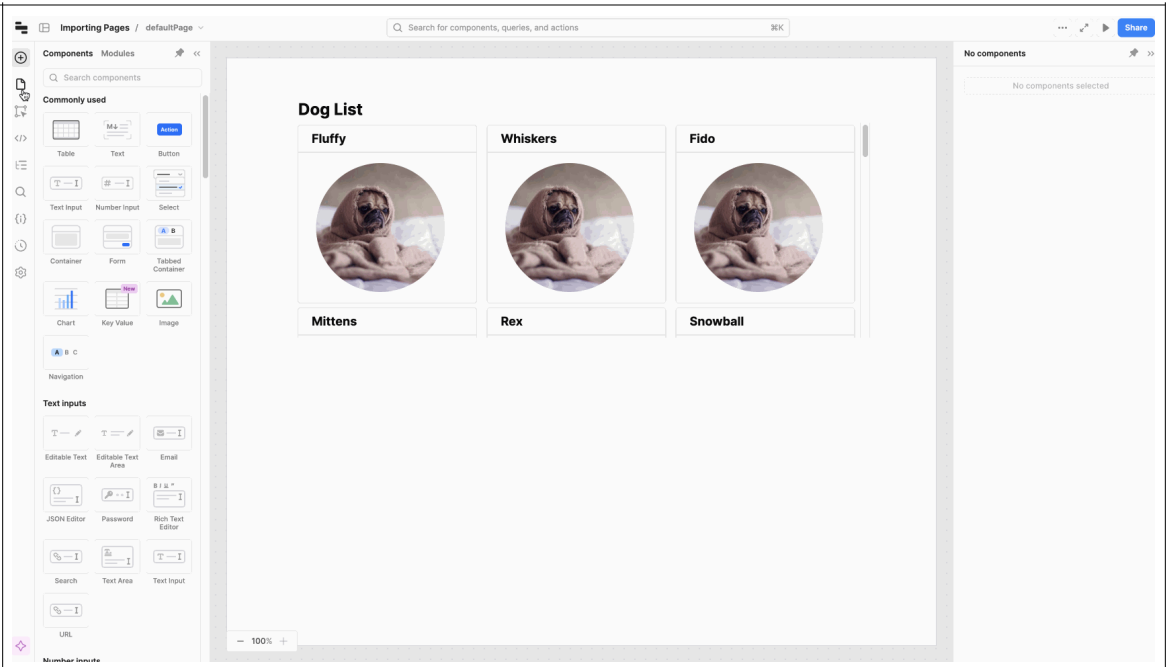
The screenshot displays the Retool dashboard. At the top, there is a navigation bar with 'Retool' and menu items: 'Apps', 'Resources', 'Query Library', 'Grid', and 'Workflows'. A yellow warning banner at the top reads: 'Weak Password Detected. Your password has appeared in a data breach at a different website and is likely to be compromised. To secure your account, change your password to one that is unique. Change Password'. Below the banner, the main content area is titled 'Welcome, Retool' and includes a search bar and a 'Create' button. A table lists applications:

All	Recent	Starred	Modules	Folders	Trash
<input type="checkbox"/>	Name	Updated	Last Editor		
<input checked="" type="checkbox"/>	Multipage Features	3 minutes ago	Retool Admin		...
<input type="checkbox"/>	Old App	a few seconds ago	Retool Admin	Edit	View

To the right of the table, there is a 'Get Started' section with buttons for 'Web app', 'Mobile app', 'Resource', 'Workflow', and 'Form'. Below that is a 'What's New' section with three cards: 'External apps', 'Workflows', and 'Get started with AI'. A small '2' icon is visible in the bottom right corner of the interface.

Importing single page apps to a Multipage app

- Import existing single pages apps to your multipage via the Add Pages dropdown
- If there is a Global Header/Sidebar conflict, you can choose whether to override or keep the current Header/Sidebar
- Like Duplicate page, all code will be page scoped
- App settings are ignored



Tips and tricks

- If your current app consists of multiple apps connected by a Header/Sidebar module
 1. Create a global header/sidebar
 2. Import each app using "Import app as page"
- If your current app consists of a tabbed container where each tab is a separate page
 1. Use "Duplicate app as multipage" to migrate your entire app to multipage
 2. Create a new page for each tab in the container
 3. Copy paste each tab's components to its own page.
 4. Use code and folder "Move to page" to move code from the original page to the relevant page.
 - You can access this by right-clicking the code/folder
 - If multiple pages rely on the same code, make it global scope
 5. There is likely to be name reference errors. If your queries don't work, check to see if they're referencing the right components/code on the page.
- Import app as page Reference errors
 1. When you replace the current page's header/sidebar with an imported app's header/sidebar, you may find reference errors between the header/sidebar's components and page-scoped components/queries.
 2. In multipage apps, scope prevents references between global header/sidebar components and page-scoped components/queries.

3. To fix, refactor your app:

- You can use global variables or queries to share data between the header/sidebar and page-scoped components.
- Also, you can convert page-scoped queries to global queries to make them accessible to the header/sidebar.

Asks

Feel free to leave any feedback. Below are usability questions we would love to get your feedback on:

- Behavior of the new `url` object for accessing **URL params**. Does this behave as expected?
- **Access restrictions** between screens. Is this easy to understand or annoying to deal with?
- **Global code vs. page code**. Is it clear when to use one or the other? What challenges remain?
- **Performance**. How does the performance of switching between pages feel?
- **Page creation and setup**. Do the interactions in the app structure tab and page inspector make sense? Is anything missing?
- **Navigating between pages**. Is it clear how you can trigger navigating to another page?
- **Passing data from one page to another. Sharing data between pages**. Is it clear how you would pass data from one page to another? Is it clear what logic primitives to use if you'd like to share data between two pages?
- **Shared/Global header & sidebar**: Right now you can create a header & sidebar shared among all pages. Does this make sense? Any other shared/global UI you wish you could build? Does shared/global sidebar & header get in the way of building what you want to build?

Known issues

Below are some known bugs, improvements, and features we plan to implement in the near future. If you notice any of these while you are building, know that we will be fixing these soon! Any other feedback, please feel free to drop in Slack!

Bugs

- Import page
 - Importing an app as a page that had 'Expand to fit content' will not carry over in the multipage app
 - Workaround: Click on the page's main frame and go to inspector and check 'Expand to fit content'

- Split pane
 - You can only create one page-scoped split pane right now.
 - A fix for this is coming out soon.

Compatibility w/ other Features

- Multiplayer
 - Multiplayer is not supported on multipage yet. Workaround: Use source control or use releases to manage changes to your application.
- Theming
 - Styles and themes do not apply on pages outside of the default page
- Public / Embedded Apps
 - Switching pages results in an error in public apps
 - Embedded apps do not render all parts of a multipage app
- Custom Product Pages
 - You cannot currently create multipage apps and set those as your custom product pages.
- Custom CSS
 - You cannot reference elements on pages via `{{ }}` when writing custom CSS.
- Custom Keyboard Shortcuts
 - You cannot reference elements on pages via `{{ }}` when writing custom keyboard shortcuts.
- Performance tab of Debug Tools
 - This is no longer accurately determining the performance of your application. We will fix the scoring for this.

Upcoming features

- Global & page-specific frames (e.g. Drawer, Modal, Split Pane, Sidebar, Header)
- Navigation component connected to pages out-of-the-box
 - When you add a page, a menu item is added to the Navigation component
 - When you have pages in your app and drag out a Navigation component, its already connected to the pages
- Page access based on role
- Multiplayer
- Public apps, Embed, Portals support
- Let us know if you have [any other feature requests here!](#)