MICHAEL SHELELMAN

Senior Technical Writer

(408) 607-2116 · Redding, California · sheleman.m@gmail.com · LinkedIn

PORTFOLIO ARTICLES

- <u>Conceptual Article</u>
- Reference Guide
- Procedural Guide
- Documentation Site

Preface

As a startup, Unqork requires its technical writers to produce high-quality, high-volume content at a fast pace to support five development teams. The company's no-code platform simplifies web development for complex applications used by fintech, government, and insurance organizations around the world.

Since 2021, I've researched, created, and managed documentation for the Unqork SaaS platform. My role has grown from making minor edits to owning the documentation for approximately 90% of all new platform features.

I specialize in creating procedural guides, conceptual overviews, reference documentation, FAQs, and API docs (REST and GraphQL). I also manage the design, structure, and deployment of the public-facing documentation site.

Beyond my core responsibilities, I've co-led an employee resource group and pursued technical learning in Python, CSS, XML, and JavaScript. Most recently, I built and deployed my personal website using Docusaurus, a docs-as-code static site generator.

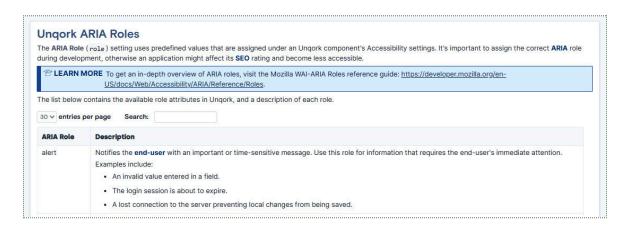
Below are four recent, public-facing projects representing conceptual, reference and procedural documentation, as well as the design and layout of the Unqork Documentation Hub.

Conceptual Article



Title	Bring Your Own (BYO) Component SDK
Article Link	https://docs.unqork.io/Content/A00-Unqork_Platform/A005 06-BYO-Component.htm
Article Type	Conceptual
Target Audience	Experienced JavaScript Developers
Platform	Unqork software platform, a SaaS (Software as a Service) product designed to create no-code web applications for financial technology companies.
About	This article details a new feature that enables javascript developers to create their own components for use in the Module Builder tool.
Key Responsibilities	 - Research: Evaluated and compiled internal and external documentation. - Development: Created layout, deisgn, styling, and language. - Collaboration: Worked cross-functionally with product manager and two engineers, to create accurate documentation for end-users. - Implementation: Uploaded, cross-linked, and promoted article on Unqork Doc Hub.
Outcomes	 Provied documenation for new feature to over a thousand Unqork end-users. Reduced workload for customer support and support engineers.
Authoring Tool	Madcap Flare

Reference Guide



Title	ARIA Reference Guide
Article Link	https://docs.unqork.io/Content/Q-References/Q014-ARIA_Accessibility.htm
Article Type	Reference Table
Target Audience	Unqork Platform Users
Platform	Unqork software platform, a SaaS (Software as a Service) product designed to create no-code web applications for financial technology companies.
About	Unqork's recent push into accessibility lead to a need for documentation on ARIA (Accessible Rich Internet Applications) settings and how to implement them into an Unqork application.
Key Responsibilities	 - Research: Evaluated and compiled internal and external documentation. - Development: Created layout, deisgn, styling, and language. - Collaboration: Worked cross-functionally with product manager and two engineers, to create accurate documentation for end-users. - Implementation: Uploaded, cross-linked, and promoted article on Unqork Doc Hub.
Outcomes	 Provied documenation for new feature to over a thousand Unqork end-users. Reduced workload for customer support and support engineers.
Authoring Tool	Madcap Flare

Procedural Guide

Adding a Get Operator

For this example, create a <u>Data Table</u> component that stores a list of countries and their capitals. Then, set up a <u>Data Workflow</u> component with two Get operators. One Get operator retrieves an entire row of data, and the other retrieves a single value.

NOTE These instructions assume you have a new module open, saved, and with a title.

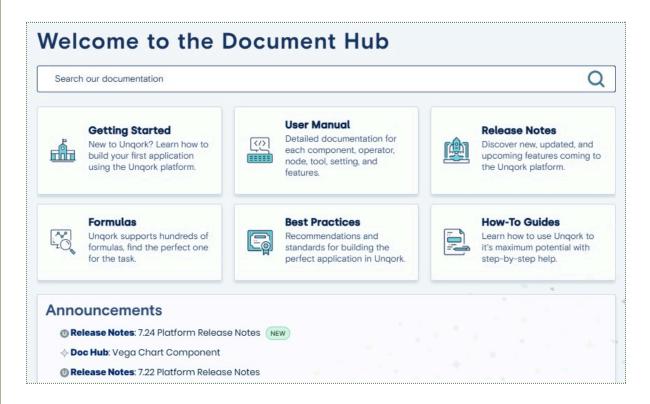
Configure the Data Table Component

First, you'll add a Data Table component to store some sample data.

- 1. In the Module Builder, drag and drop a 🖽 Data Table component onto your canvas.
- 2. In the Property ID field, enter dtCountries.
- 3. In the data table, enter the following:

Title	Get Operator
Article Link	https://docs.unqork.io/Content/A06-ModEditor-Data_Workflow/A060 5-Array/A0605002-Get_Operator.htm
Article Type	Software Tool Procedural Guide
Target Audience	Unqork Platform Users
Platform	Unqork software platform, a SaaS (Software as a Service) product designed to create no-code web applications for financial technology companies.
About	The Unqork platform enables it's users to create complex logic operations using a tool called the "Data Workflow Component". One of these operations, the Get Operator, lets users select a single field's value from a data set. This article presents a simple procedural guide for using the Get Operator.
Key Responsibilities	 - Research: Evaluated and compiled internal and external documentation. - Development: Created layout, deisgn, styling, and language. - Collaboration: Worked cross-functionally with product manager and two engineers, to create accurate documentation for end-users. - Implementation: Uploaded, cross-linked, and promoted article on Unqork Doc Hub.
Outcomes	 Provied documenation for new feature to over a thousand Unqork end-users. Reduced workload for customer support and support engineers.
Authoring Tool	Madcap Flare

Documentation Site



Title	Unqork Documentation Hub
Article Link	https://docs.unqork.io/
Article Type	Documentation Website
Target Audience	Unqork Platform Users
Platform	Unqork software platform, a SaaS (Software as a Service) product designed to create no-code web applications for financial technology companies.
About	Designed, implemented, and authored the public-facing documentation site for the Unqork platform used by web developers. The goal was to provide clear, searchable, and scalable product documentation to improve user onboarding, reduce support tickets, and enhance product transparency.

Key Responsibilities	 - Site Design & Layout: Built a responsive and accessible documentation site using MadCap Flare, focusing on ease of navigation, clean information hierarchy, and consistent branding. - Implementation: Configured site structure, versioning, and custom components to support both internal teams and external users. Used Git and CI/CD pipelines to publish updates. - Content Strategy and Language: Developed documentation tone and style aligned with the company's voice. Wrote task-based, user-centric help content, onboarding guides, and API references. - Collaboration: Worked cross-functionally with product managers, engineers, and UX designers to ensure documentation matched real-world user workflows and product functionality.
Outcomes	 Simplified navigation by reducing the average amount of clicks needed to reach an article from six to four. Increased time on page for users by ~45%. Documentation oragnic traffic increased by ~650% via improved SEO (search engine discoverability).
Authoring Tool	Madcap Flare