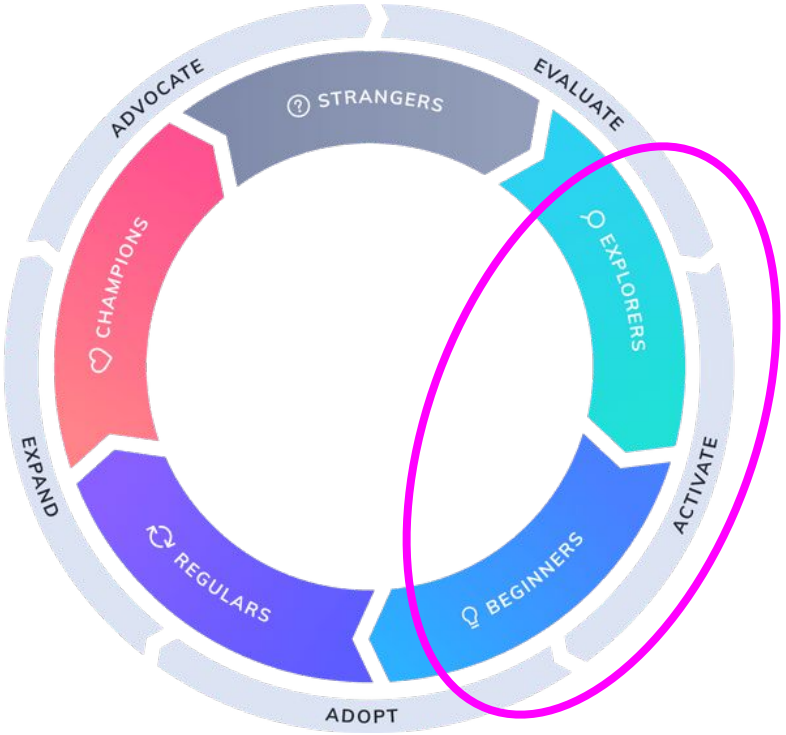
sisense

Onboarding - Research Findings

Growth / Activation

Definition of Onboarding

Ensuring the audience experiences the core value as quickly as possible.



Research

20 developer products were researched that involve data, coding, APIs, deployment & more
Developers onboarded products and notated their experience

Target Audience of Products
Developer, Data Engineer, Data Analyst, CloudOps, QA

Entry Point

All tools provide ***free entry*** to the product, with the most common being Freemium. This is important for developers to be able to try the product.

Important nuance is to give access to try the product before asking for anything else, even email authorization and especially asking for a credit card. Forms can be filled out after initial login. This causes the least amount of friction and provides the smoothest experience.

An outstanding example is CodePen. It gives free, frictionless entry and asks to sign up when you click to save your first project.

Learning outside of product

Marketing Page: Most learning starts on the Marketing Page. Good experience is usually rated by developers if value is clearly shown and explained on the Marketing page. Good example is **Snowflake**, where value is explained without Marketing jargon so users feel confident they want to try the tool.

Product Preview: It is good practice to show the UI of the product on the Marketing page. This is also the preliminary step to onboarding, as the user is already getting familiar with the user interface.

Intro Video of UI & value: If the video is well made, with ***good graphics***, developers will watch to understand the product to understand how the product can ***provide value*** to them.

Docs: Documentation was mentioned in every tool that was researched. Docs need an easy-to-follow QuickStart section with ***code samples*** for a good onboarding experience.

Tutorials: If the tool is complex, developers will benefit from tutorials, which are often placed in documentation.

Community: Community gives developers the feeling that all their questions will be answered.

Dev friend: This is a great indication that a tool brings value, whether it has better or worse UI.

In-app Onboarding Tools – Recommended

Self-explanatory UI: Best approach is to present minimalist self-explanatory UI that developers can get to know by using, building, coding.

Non-intrusive Tooltips: Much better than wizards. Important that these are relevant to the user.

Sample code & commented out instructions: Amazing to provide code samples that developers can explore at their own pace and style!

AI Assistance: AI is a great nice to have, but should consider the flow of work of a developer.

Empty-state of product: Empty states should be accompanied with instructions how to create the first asset.

Docs: Docs should not be placed inside the product. Tooltips and empty states can explain relevant snippets of information. Links to relevant doc articles should always be provided throughout the product and at high-level as a help feature.

Create 1st asset guided: Creates feeling of ownership and explanations don't feel tedious. Should have the goal of showing value of using the product with creation of the first asset.

Progress bar with steps to complete: Should not contain more than 5 simple onboarding steps to take the user to a value point. Use only if necessary.

Getting Started: Developers prefer a clean Getting Started page with max of 3 options. Include only if product is complex. Otherwise, it is better to land the user closer to "aha moments".

In-app Onboarding Tools - Not Recommended

Getting Started: A page with many links to resources may seem like a good idea, but it creates a feeling of cluttered design and developers are presented with too many options. This creates an overwhelming onboarding experience, in which each user may click on a different link.

In App Guide/Wizard: Not recommended. Use only when absolutely necessary. Wizards take away the feeling of freedom, flexibility and working at my own pace.

Chat: Developers would only use a chatbot as a last resource, if UI or Docs were not helpful, only when they have a very specific case to ask about, such as in a bank app.

In app docs: The general expectation is that documentation is always on a web page, different from the product.

Onboarding Proposal

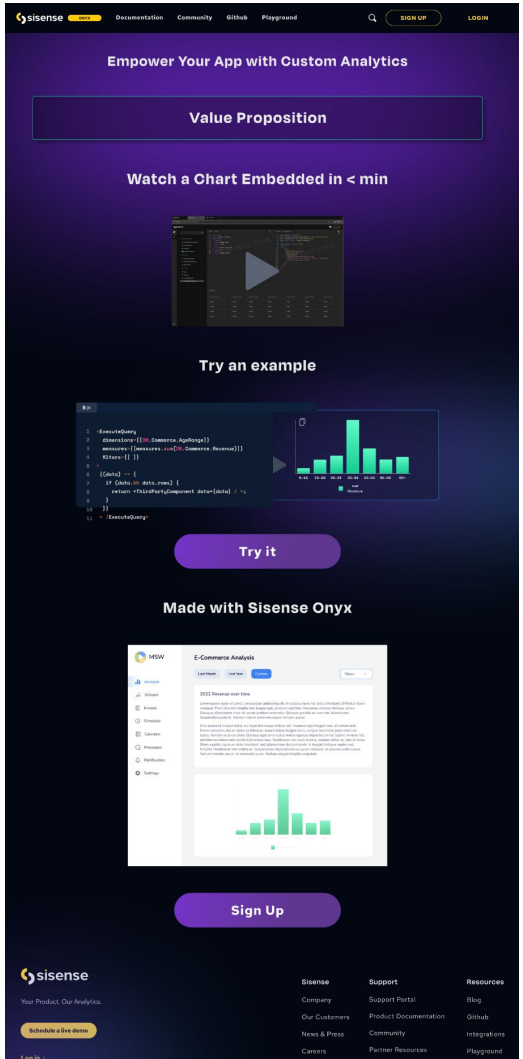
Onyx

Marketing Page

Marketing page should show why a developer would benefit from the product and prime user by showing inspiring examples. A developer should be able to explain the value of Sisense to their colleague in a sentence after skimming the website.

It should include:

- Value Proposition
- Video/gifs UI to showcase value
- Examples of Embedded Charts to click & try them



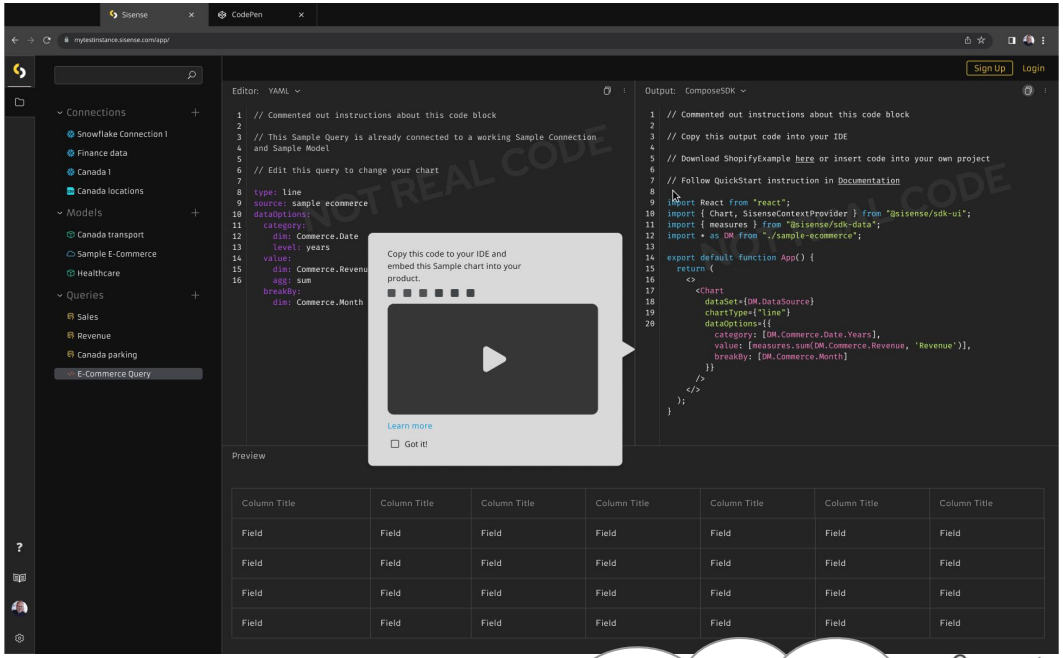
Looks like this product does what I am looking for and seeing examples of end results is inspiring. **Let me try!**

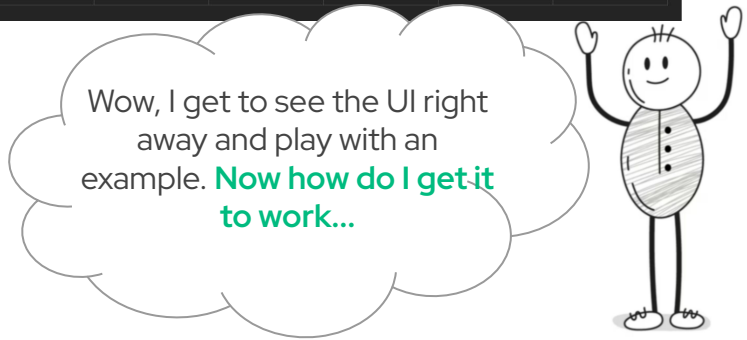
Easy Entry, Quick Sample

Developers are eager to try a product and are sceptical until value is proven.

By clicking on an example on the marketing page, they are taken to the query with Compose SDK Output. Simple instructions should take the developer through a few steps to:

- Copy output code
- Insert it into a project (a sample we provide or their own)
- View a chart embedded

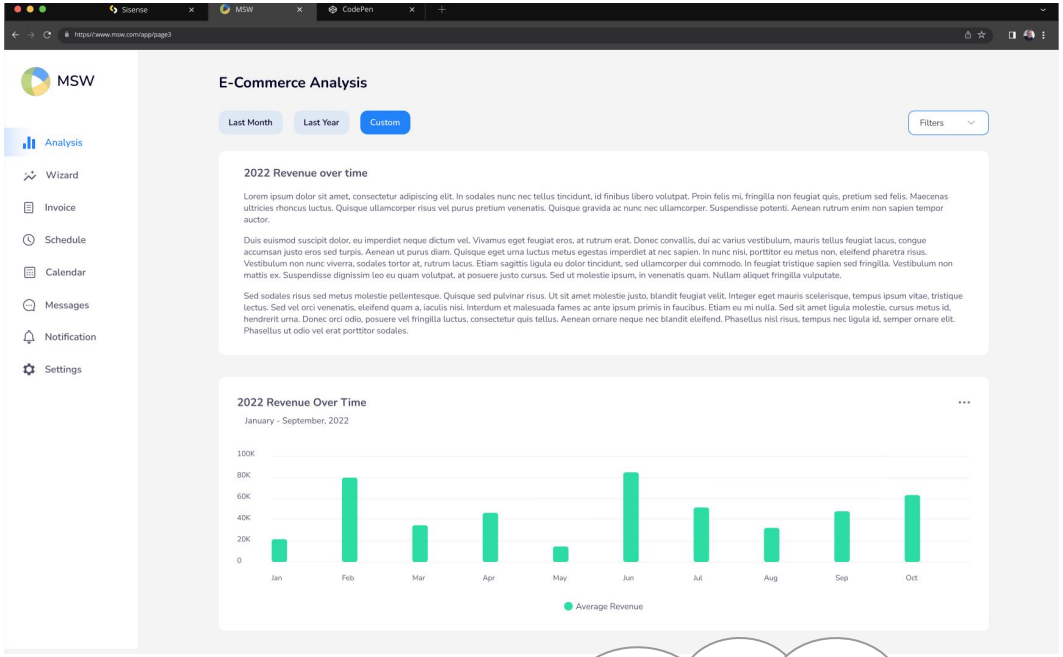


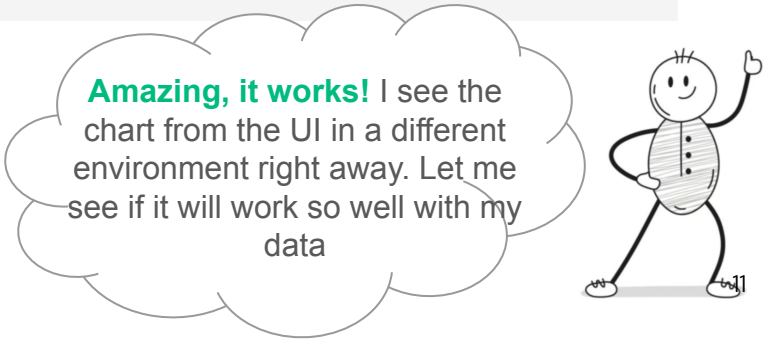


sisense

Quick win, it works!

Seeing a working example is a strong **Aha Moment**. Trust is built and the user is more psyched to sign-up.





sisense

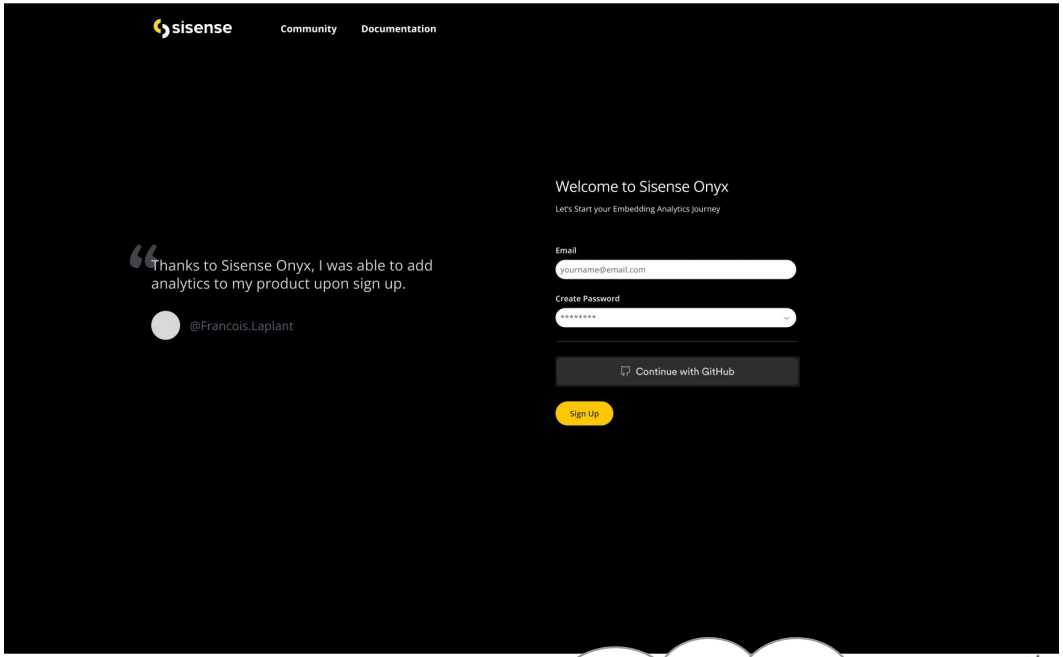
Simple Sign-up & Login

Less is more.

Developers don't have access to the company's credit card, so there should be no payment info required.

After value has been showcased and trust won, it is much easier to ask for forms to be filled out.

- Email
- Create Password
- One-click sign-up options, such as GitHub
- Reviews from developers on the side

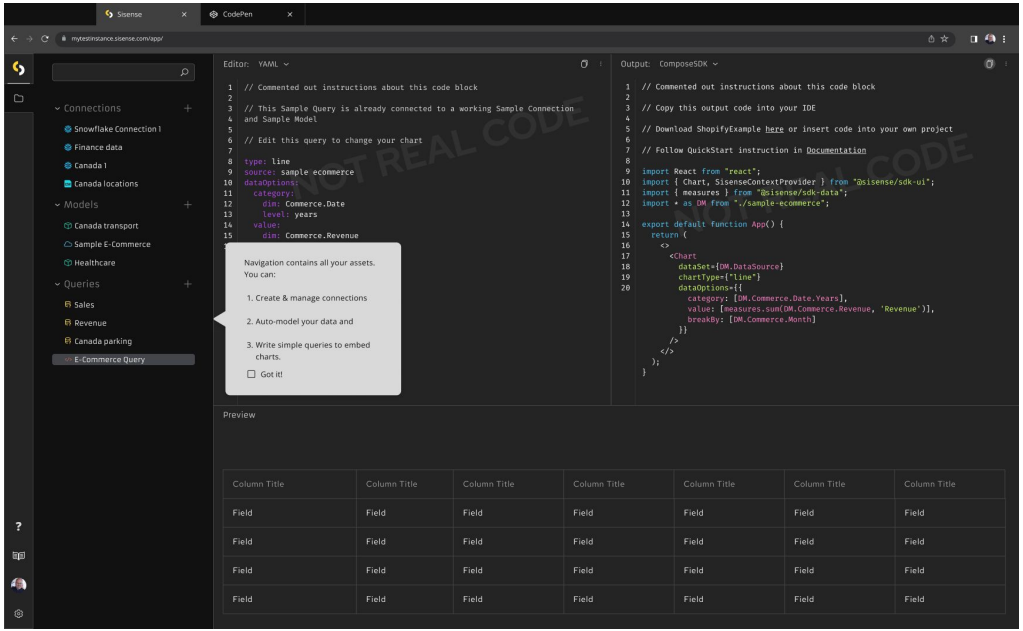


sisense

In-Product Onboarding

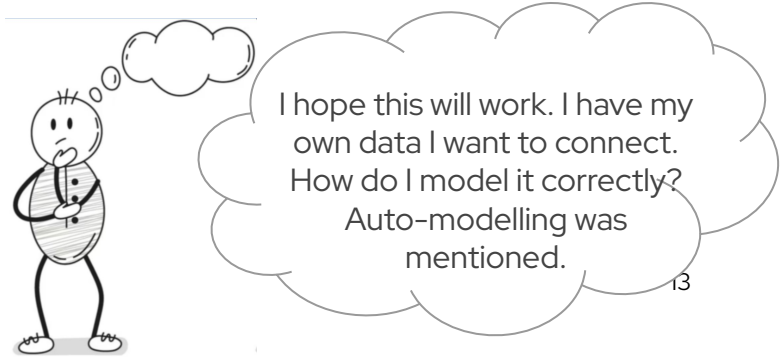
Now the developer is ready to build their own assets. Developers don't like chats, restrictive wizards or long mandatory bubbles (3 max).

The rest of the interface can be explained via tooltips and a concise QuickStart Documentation with short videos/images & code samples. The rest of the documentation should be very thorough.



Tools:

- Max 3 onboarding bubbles situating the user and get to value point
- Non-invasive tooltips throughout the product
- Commented out instructions for code samples
- Empty states contain parts of documentation and action points, helping create assets



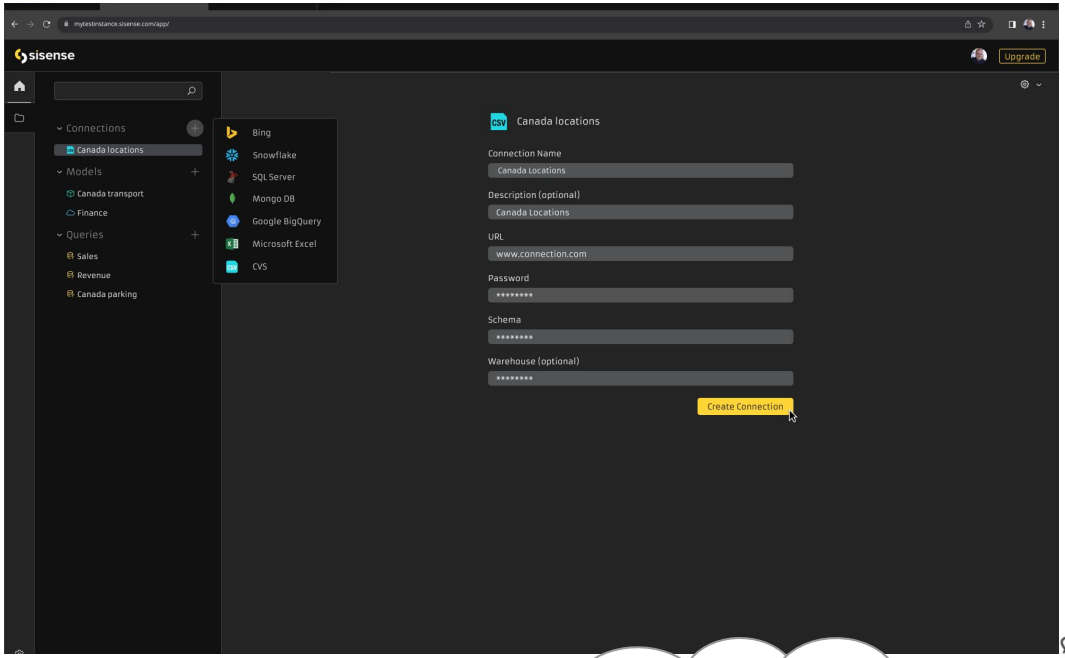
sisense

First assets

Aha Moment: I use my own data to Embed a Chart/Analytics into my project

Users should be able to freely explore the product and build their first assets, with the help of non-invasive tooltips, empty states explanations and documentation. This exploratory stage includes:

- Create my own Connection
- Edit/Create a Model
- Utilize Auto-modelling (aha moment)
- Create a new Query
- Insert Compose SDK code into my project in an IDE to embed



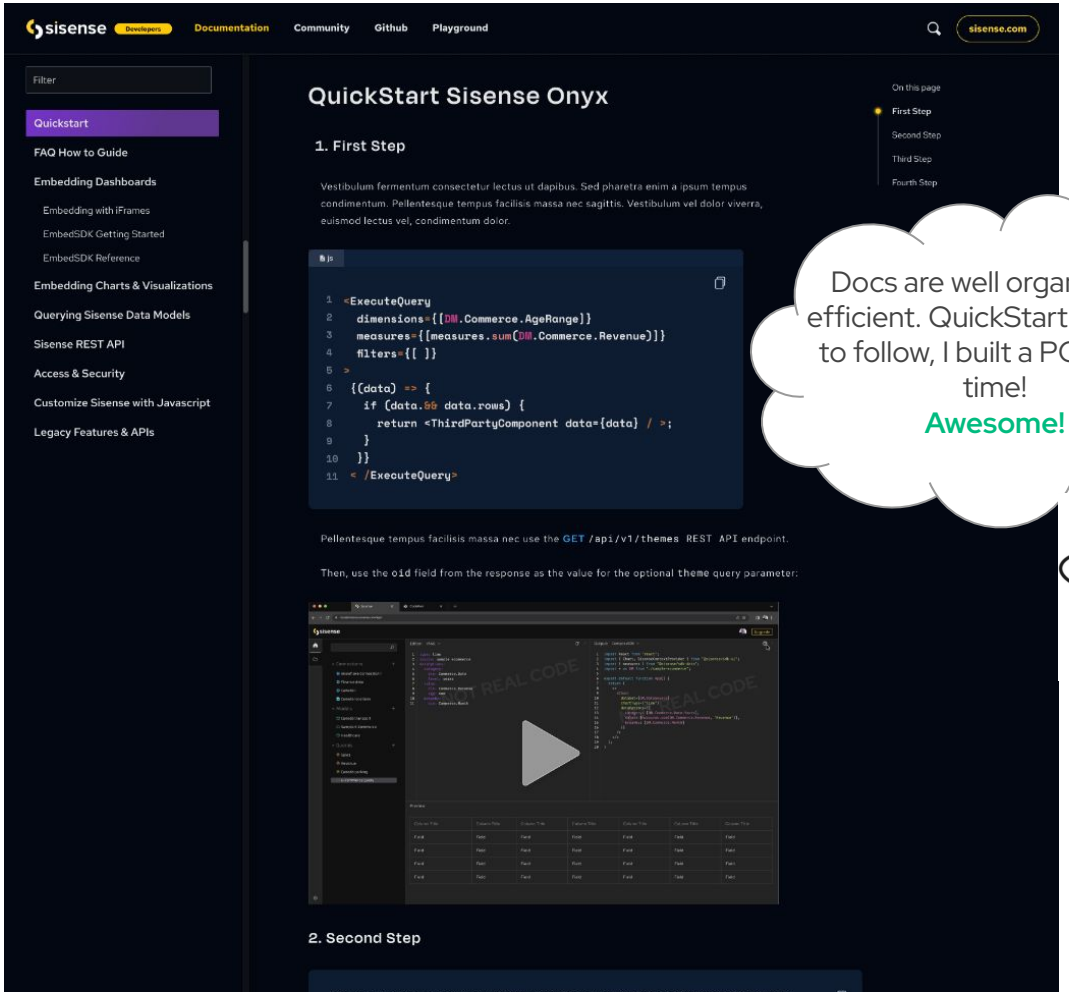
sisense

Documentation

Aha Moment: It is easy to find the QuickStart section in the documentation and from no knowledge, I am able to quickly build a POC by following concise steps that consist of code samples and short videos/images.

The rest of the documentation should be much more verbose and detailed, for when the developer has completed initial onboarding.

- Concise **QuickStart section** with code samples and short videos (can be repeated from Marketing and In-App Onboarding)
- How-to-Guide for FAQs
- *Troubleshooting Support Section*







Thank you

sisense