



QUESTIONS

RESPONSES 97

97 responses

SUMMARY

INDIVIDUAL

Get email notifications for new responses

Select response destination

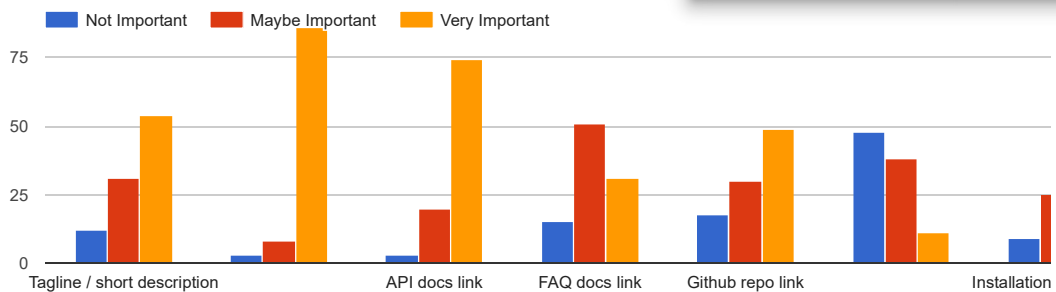
Unlink form

Download responses (.csv)

Print all responses

Delete all responses

What content should a library docs site front page have? What are the most important priority descriptions, examples, info, and docs links that should be shown away?



Other suggestions for front page content:

15 responses

Short animation off all the moving piece

IMO copy-pasteable examples are more important than runnable ones

GIVE ME MORE REAL WORLD COMPLEX EXAMPLES!!! I'M HUNGRY, you guys already did an awesome job with docs and site.

Typescript. There is hardly any documentation about using redux or react redux with typescript. It is understandable as when all this happened, typescript wasn't even a thing. But now, you should have it IMO

Diagrams of data flow. Examples of how redux prevents bugs and aids debugging.

Redux Docs look good!

I want to know "Why, What, How"

A search bar

Keep it as short as possible. Redux docs in general are pretty hard to grasp since there is so much to read. As a non native English reader this is even harder, I found my self couple of times never finishing reading a page.

An ideal site that's 100% perfect would have a small example, a longer getting started guide, a cookbook for common topics, and then detailed API docs

easy to navigate / ctrl + f

It would be nice to have a visual (diagrams, tables, arrows) overview of the Redux work flow

Nice to have: Cool graphics!

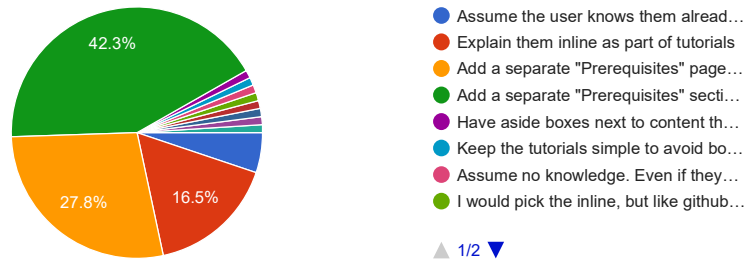


QUESTIONS

RESPONSES 97

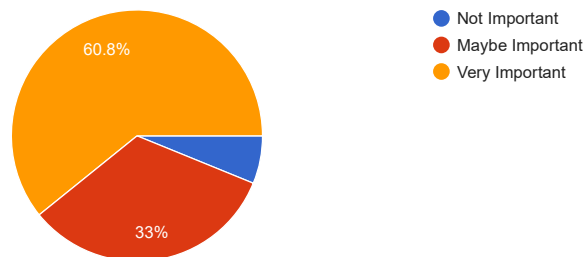
necessary for understanding it? How should those be structured page-wise? Integrated right into a tutorial? Separate "Prerequisites" page? Detailed explanations, or just links to more reading? (Specifically for Redux, "prerequisites" would be things like immutability, `Array.reduce`, event emitters, and maybe higher-order functions and other array methods.)

97 responses



How valuable would it be to have guided "learning paths" for beginners, people who want to understand how it works, or people who want to use it in production?

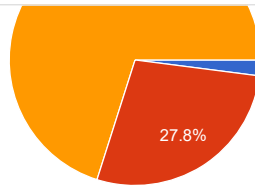
97 responses





QUESTIONS

RESPONSES 97



What topics would most benefit from diagrams and examples? What are the best ways to include those in the flow of the docs structure?

34 responses

Data flow

The overall idea and data > event flow

Architecture

Diagrams are really great on "concepts" pages, getting started, "thinking in redux" etc

Dispatchers, how state is mapped to props in react

thunks

The idea of Multireducer, Redux setup (middlewares, Provider, connect boilerplate...), action-reducer-state relationship and workflow

How an action moves from a dispatch (whether in a mapDispatchToProps call, or in an action creator) through the reducer, into the store, and updates the state view mapStateToProps.

I also believe that I only 'understood' redux when I saw the problem it was trying to solve -> Having to pass through setState handlers to children as props, in order to update parent state. Perhaps a comparison of how vanilla react vs redux react would be better ingested in visual form.

I believe keeping the diagrams inline, or as a potential popup on hover / toggle would be the best way.

The Gist of redux(with each "piece" being an animal or human :P), it would look like: Actions going to reducers, reducers going to state, state going to "view layer", view layer going to actions, with a tiny code snippet for each(actions: {type: 'INCREMENT'}, reducer: 'return state + 1')

Async actions with thunks or sagas.

Great question!

Simple app with Redux (maybe Todo). I want some "Hello World" moment or "Eureka" moment after writing it myself and get convinced to use Redux

Actions and thunking

Data-Flow

Dispatch, action, reducer digram.

Explaining internal concepts like actions-reducer flow, middlewares etc would be easier to understand with diagrams and could be part of the API documentation. A component diagram might help explain a particular solution for a common problem (API calls).

reducer examples and bullet proof core docs set up (the provider, the store etc)

example that can read from up to down in order to easier undstanding

Give production ready example with best practices in tutorial



middleware - no suggestion how best to integrate

dispatch subscribe flow.

How to integrate Redux with the React from scratch rather than using CRA.
Helping to understand the core fundamental of the redux and more importantly Why they are used in such a way.

Throw some stuff in the global state and then take it out

Reasons to use it / problem that it solves

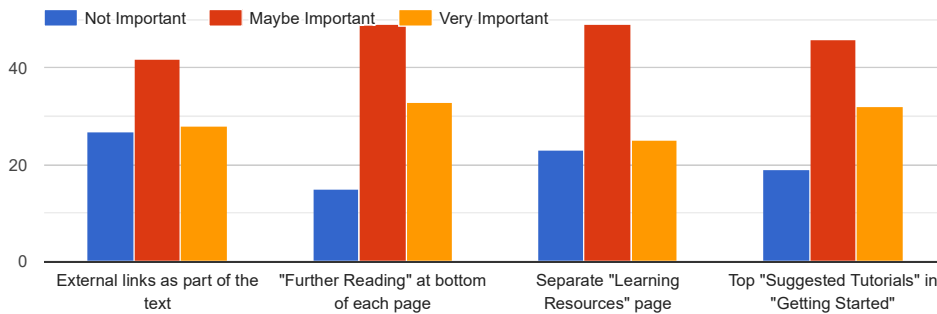
explaining redux workflow

Most topic around abstracted concept in the as well as workflows of the library

Right way of making API calls, I still can't figure it out.

Do best

How helpful is it for docs pages to link to other external tutorials and articles on a topic? Does having a "Further Reading" links section at the end of each page seem beneficial? How should other learning resources be integrated / linked from the main docs?



In general, what topics do the current Redux docs cover well? (Please link to specific pages if possible)

23 responses

3 principles

Language is too difficult.

czvzcvzcvzcv

Reason to use redux, main 4 pillars of redux(store, actions, dispatch, subscribe) all are covered in easy to understand manner but i think topics like middlewares needs some more examples and theory.

-



One area that would be nice to see is an advanced section demonstrating non-trivial use with various patterns.

The API references are done quite well

More on async, reusable redux modules / namespacing

"async actions" section took me 3 passes + reading blog posts to understand what is it and how to use it.

I actually found the current Redux docs some of the most useful I've ever seen.

How to use Redux in a large app is not clear at the beginning. I wonder to see a true example with a big use of middleware pop out

Motivation page is too long. It takes a lot of time in general to get to full example, although prior art is cool

The whole advanced section is nice. Same for recipes.

I keep coming back to the Reducing Boilerplate page specifically.

I haven't read yet, my survey is filled based on other technical documentations

The examples of the current docs are useful, but they can be used after a beginner has completely understood the data work flow from a different website. I think this could be improved on the docs.

Really good at explaining advance concepts. Needs a way to have best practices

they are like a full tutorial. I like that.

What is dispatch and why reducers

The page on testing is great.

Was asked to take survey before I could review page, so don't know.

Too much

What topics are currently explained poorly, or not covered at all? (Please link to specific pages if possible)

6 responses

Store and reducer design, maybe with some more real world examples

mapDispatchToProps and the Container Pattern

Selectors (where/when to use them), more on real use cases for middleware

see above

Too much

What content or structure changes would be most beneficial?? (Please link to specific pages if possible)

2 responses



What other docs sites should we take inspiration from?

27 responses

Egghead, RxJS,

Laravel. Symfony. PHP. Mozilla Developer Network.

reactjs.org, lodash, react native,

z<xdcvvcxczv

<https://www.valentinog.com/blog/react-redux-tutorial-beginners/>

Algolia, Spotify

-

React, Gatsby

Rails :)

material-ui has some good docs!

<https://vuejs.org/v2/guide/>

<https://golang.org/pkg/> and <https://clearbit.com/docs?shell#> and https://devdocs.io/javascript/global_objects/array

Vue, React-Router, Formik

<https://docs.scala-lang.org/>

Django! It is well known as the best python doc!

<https://reactjs.org/docs>

I really like the React docs, how they explain concepts and what NOT to do

Redux-form

Angular and React doc sites both have good gems, especially around "concept" pages & guides.

git manual - thank to it my hands stopped shaking while changed in repo (because of explaining how is git exactly working) ;)

I have found really useful this blog post: www.valentinog.com/blog/react-redux-tutorial-beginners/

djangoproject.com

None, nobody knows how to write docs

PHP's "User Contributed Notes"

<https://vuejs.org/>

Too much

Any other suggestions for improving the Redux docs?

24 responses

Illustrations with code snippets

More doc curation of quality, third-party redux libraries, extensions, and resources. More redux team activity in popular open source redux-related third-party libraries?



typescript

They are actually pretty sweet! Maybe more examples at the advanced and super advanced level?

More advanced & complex real word examples.

CheatSheets (like <https://docs.scala-lang.org/cheatsheets/index.html>); Code sandboxes (maybe similar to <https://training.play-with-docker.com/>)

In my option, there is just need to be shorter code examples and shorter paragraphs.

Redux is easy to start with, difficult to master. It applies to react as well. Redux docs are good but it could be even better with practical examples than Todo apps all over the places.

Make it very easy for beginners to understand how redux works in-depth. So beginners can become strong specialists.

Good work guys ! Thx again !!

Need more guidance on what Redux users should do in their projects, even if example with a disclaimer it's not the only good way.

Not something fully charged like Redux Toolkit, but more like: should I really keep the types in a file and the action creators in a file.. etc? Quite often it's said people make Redux more complicated than it needs to be, then: what is enough complication for scalability to stay with?

I'm glad the site mentions thunk for async stuff. I feel toolkit is too much though. How much is "practically" good enough for most redux projects, which mostly are big/complex?

Maybe also some guide around coffee splitting by feature. NgRx does this nicely because it integrates with Angular pipeline for this. What's the recommendation on coffee splitting reducers etc as part of lazy loaded modules?

Finally, maybe a blog, where you can start talking about upcoming things like hooks support.

Nothing more. Just wanna say thanks for the great work on this project and being super helpful in the react/redux community!

Ask a bunch of boot camp kids that have never used redux before if your docs are good

I would love to see a section exposing real world application / multiple good ways to right it.

Add user comments. Rated, with code highlight. The best way you can! Question and answers on the docs pages. User interaction. That's the best way to improve any documentation.

In your todo example when the container VisibleTodoList uses mapDispatchToProps, it binds a method called toggleTodo to a property called toggleTodo. For a beginner, this is mind blowing hard to understand. Why did you think it was a good idea to call two methods the same that do totally different things! Please fix the example! Beginners will never understand your examples if the naming is that bad. Thank you very much

Too much