

npmjs.com/package/json-server

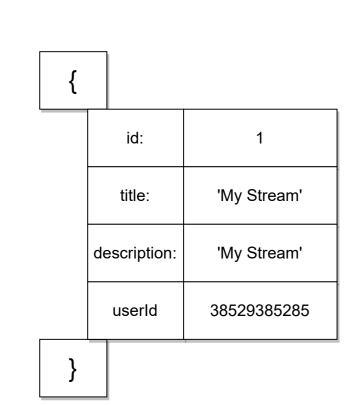
REST Conventions

Standardized system for designing API's

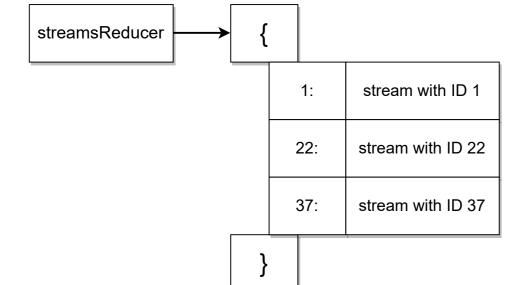
Action	Method	Route	
List all records	GET	/streams	
Get one particular record	GET	/streams/:id	
Create record	POST	/streams	
Update a record	PUT	/streams/:id	
Delete a record	DELETE	/streams/:id	

Reducer Structure Option #1

streamsReducer stream stream stream ]



## Reducer Structure Option #2



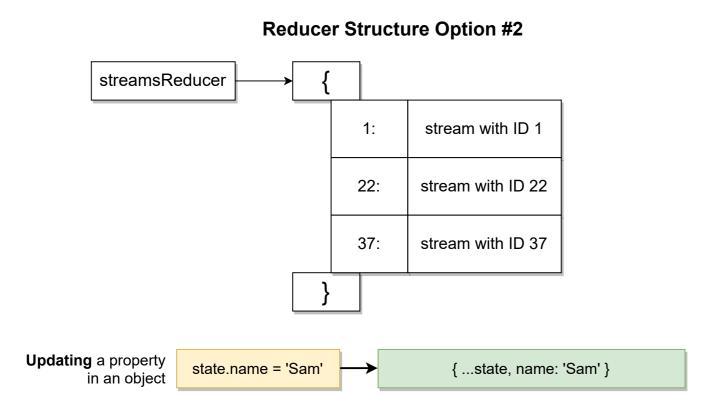
Reducer Structure Option #1

streamsReducer

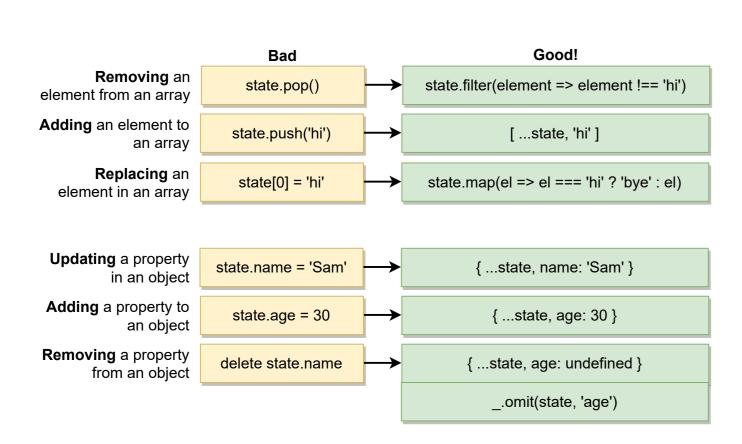
[ stream stream stream stream ]

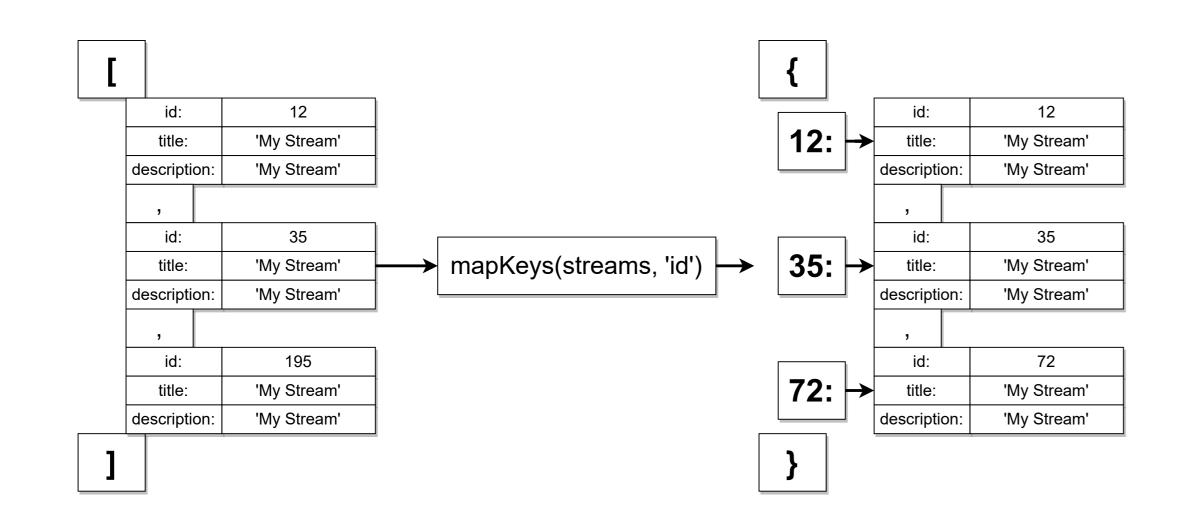
Replacing an element in an array

state.map(newStream => el === 'hi' ? 'bye' : el)



Action	Method	Route	Response
List all records	GET	/streams	Array of records
Get one particular record	GET	/streams/:id	Single record
Create record	POST	/streams	Single record
Update ALL properties of a record	PUT	/streams/:id	Single record
Update SOME properties of a record	PATCH	/streams/:id	Single record
Delete a record	DELETE	/streams/:id	Nothing





Time

User submits the form

We make request to backend API to create stream

We navigate user back to the list of streams

Time passes....

API responds with error, stream wasn't created and user doesn't know!

Time

User submits the form

We make request to backend API to create the stream

Time passes....

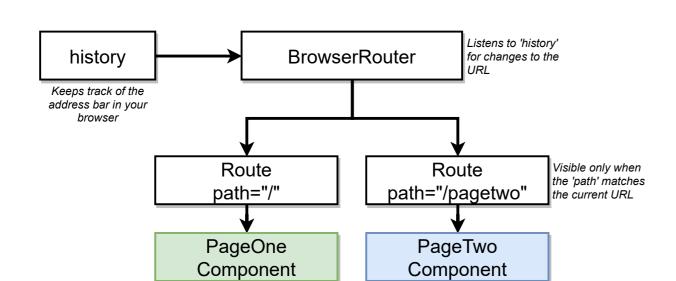
API responds with success or error

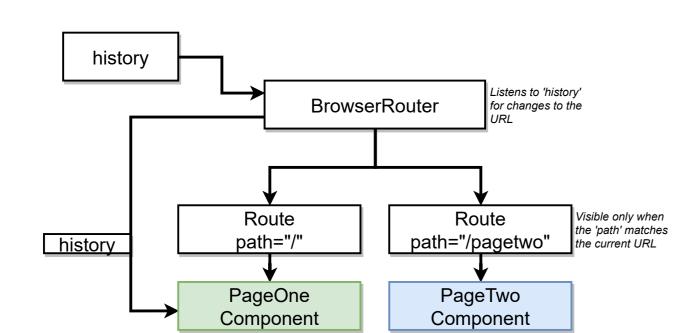
We either show error to the user or navigate them back to list of streams

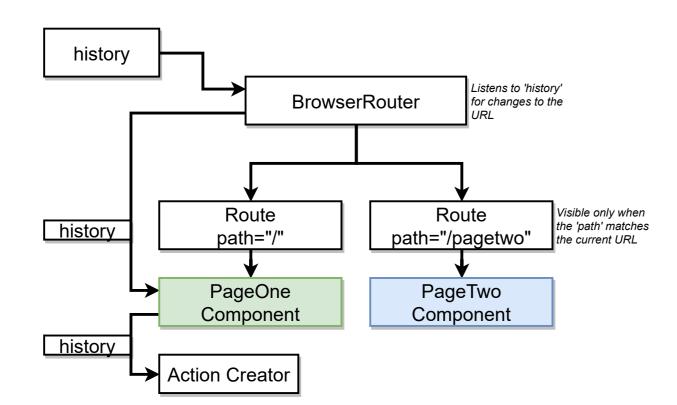
Intentional Navigation

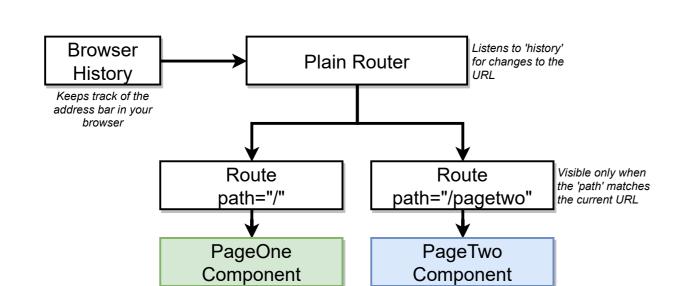
tion Programmatic Navigation

User clicks on a 'Link' component We run code to forcibly navigate the user through our app









Selection Reducer

When a user clicks on a stream to edit it, use a 'selectionReducer' to record what stream is being edited

Put the ID of the stream being edited in the URL

Path	Component	
1	StreamList	
/streams/new	StreamCreate	
/streams/edit	StreamEdit	
/streams/delete	StreamDelete	
/streams/show	StreamShow	

Path	Component	
1	<b>├</b>	StreamList
/streams/new	<b></b>	StreamCreate
/streams/edit/:id	<b></b>	StreamEdit
/streams/delete/:id	<b></b>	StreamDelete
/streams/:id	<b></b>	StreamShow

Selection Reducer

Keeps selection 'state' in Redux, but tough for a user to navigate to a page and see correct content

URL-based selection

Works well with navigation-enabled apps

User types in /streams/edit/3 to address bar and hits enter

User loads up our app

Redux state object is empty!

We try to select stream with id '3' from state

No streams were loaded, so we get undefined!

We navigated to '/'

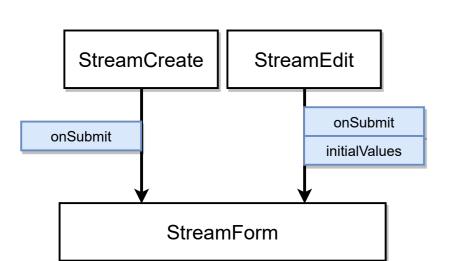
StreamList fetches all of our streams, updates Redux state

We navigate back to '/streams/edit/3'

We select stream with id of 3

Data is now in redux store, so we see the appropriate stream

With React-Router, each component needs to be designed to work in isolation (fetch its own data!)



```
onSubmit
initialValues for 'title' and 'description'

ReduxForm

StreamForm
```