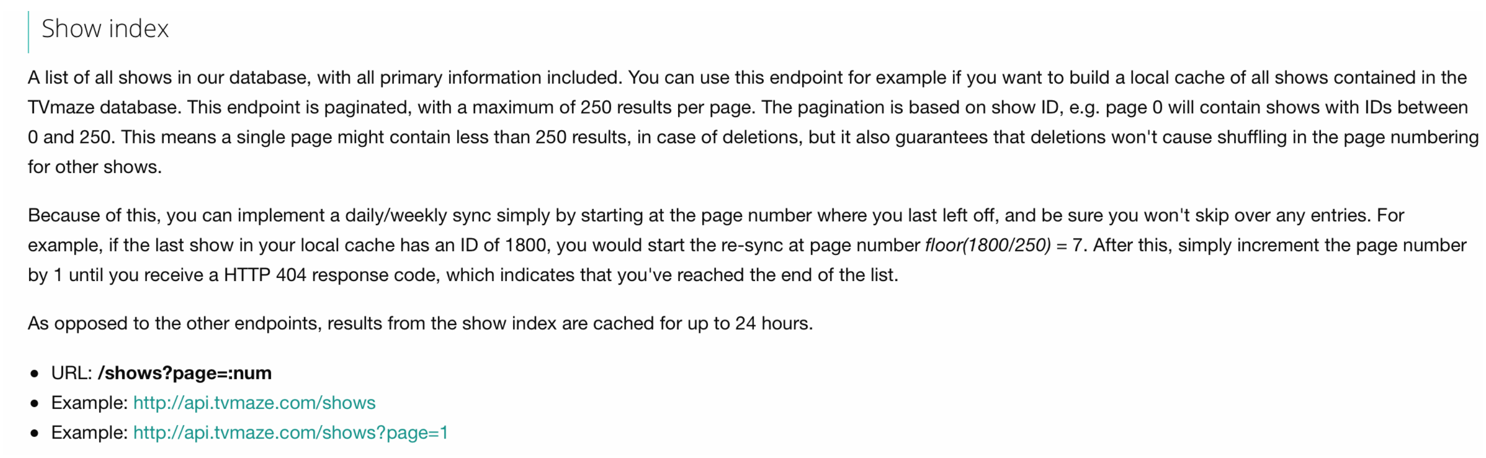
React Exercise

Due Mar 7 by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip

For this exercise you are going to take the updated TV Maze code found on GitHub: **https://github.com/stevens-cs546-cs554/CS-554/tree/new-tv-maze-functions/react_second_lecture/tv-maze-function-components (https://github.com/stevens-cs546-cs554/CS-554/tree/new-tv-maze-functions/react_second_lecture/tv-maze-function-components)**

Review the TV Maze API documentation: **http://www.tvmaze.com/api#show-main-information (http://www.tvmaze.com/api#show-main-information)**

Specifically the part on pagination for the show list:



Show index

A list of all shows in our database, with all primary information included. You can use this endpoint for example if you want to build a local cache of all shows contained in the TVmaze database. This endpoint is paginated, with a maximum of 250 results per page. The pagination is based on show ID, e.g. page 0 will contain shows with IDs between 0 and 250. This means a single page might contain less than 250 results, in case of deletions, but it also guarantees that deletions won't cause shuffling in the page numbering for other shows.

Because of this, you can implement a daily/weekly sync simply by starting at the page number where you last left off, and be sure you won't skip over any entries. For example, if the last show in your local cache has an ID of 1800, you would start the re-sync at page number *floor(1800/250)* = 7. After this, simply increment the page number by 1 until you receive a HTTP 404 response code, which indicates that you've reached the end of the list.

As opposed to the other endpoints, results from the show index are cached for up to 24 hours.

- URL: **/shows?page=:num**
- Example: http://api.tvmaze.com/shows
- Example: http://api.tvmaze.com/shows?page=1

We have seen the Shows TV maze url before: **http://api.tvmaze.com/shows (http://api.tvmaze.com/shows)**

We can see from the documentation, that the shows returned at that URL are just the first 250 shows in the dataset.

You will take the code from the course GitHub linked above and then modify it. You will add pagination to the shows list. This will help you when we get to our lab on React as you'll need to implement pagination into that lab so this will be good practice.

You will need to make sure the the URL displayed is the page number for that set of shows. I would recommend perhaps using a different URL pattern like localhost:3000/shows/page/0 Would show the initial list of 250 shows, localhost:3000/shows/page/1 would show the next 250 shows and so on.. Manually changing the URL in the URL bar should also work and trigger the data change.

In the list of shows, you will have a next and previous button/link that will go to the next page. so for example, if you are on localhost:3000/shows/page/0 and click Next, it should show the URL localhost:3000/shows/page/1 in the URL address bar and then fetch the 2nd page of shows.

The Previous link should NOT be visible on the first page since there are no previous results. On the last page of the results, you will not show the Next button/link since we are at the end of the results.

For the pagination, you'll want to use a useEffect hook that depends on : `props.match.params.pagenum` where `pagenum` is the name of the parameter you define. (you can see this in the show details component, I watch for changes to the id in the URL)

Here is a video going over the changes that were made to the TV Maze Codebase, and I also demo the pagination from the older lab 4 on React so you can see how it should work.

React Ex.mp4 (https://sit.instructure.com/courses/39041/files/6999185/download?download_frd=1)