Chassi Code Kata

Your task for this kata is to perform a ReactJs/Redux lifecycle that queries an api for a list of fisherman's catches and displays it to the screen using a Material-UI (beta) Table to form a leaderboard of competing anglers. This leaderboard should be thought of a standings in a tournament of big fish such that the top angler is displayed first, followed by the second top angler and so forth.

You will have 4 hours to perform this task, and unlike a typical story where how and what to display are well defined, in this kata we will leave that up to you to choose.

Table of Contents

Libraries required to complete the kata

Here is my solutions dependencies list:

```
"dependencies": {
    "axios": "^0.17.1",
    "axios-mock-adapter": "^1.10.0",
    "material-ui": "^1.0.0-beta.21",
    "react": "^16.1.1",
    "react-dom": "^16.1.1",
    "react-redux": "^5.0.6",
    "redux": "^3.7.2",
    "redux-promise-middleware": "^5.0.0"
  }
```

Public API

The root URL for the task is `https://868r1t0.restletmocks.net`.

There is only one resource available for consumption:

- GET /catfish

I have included 2 .pdf files in the repository which are print outs of the restlet studio artifacts, but I will cover what is on them here in detail.

DTO

There exists one Data Transfer Object (DTO). In this case, the DTO is the format of the data being sent back by the server and it looks like the following:

```
{
    id: [integer],
    species: [enum],
    length: [number],
    girth: [number],
    anglerName: [string]
}
```

The possible enumerations for `species` are: `blue`, `channel`, and `flathead`. Length and Girth are always positive real numbers.

GET /catfish

The resource at `GET /catfish` returns a hardcoded list of 100 catfish DTOs. The id's are sequential, the species a random selection of the enumerated values, the length and girth are random numbers between 30 and 80, and the angler name is one of 5 possible names drawn at random.

For brevity, here are the first 3 DTOs of the list

```
[{
    "id": 1,
    "angler": "Ricky",
    "length": 49.12864320641791,
    "girth": 35.66288053927431,
    "species": "blue"
},{
    "id": 2,
    "angler": "Anthony",
    "length": 50.95331832904824,
    "girth": 40.51948182204999,
    "species": "blue"
},{
    "id": 3,
    "angler": "Anthony",
    "length": 57.35317111720916,
    "girth": 48.40535114101511,
    "species": "blue"
},{
    ...
}]
```

Kata Instructions

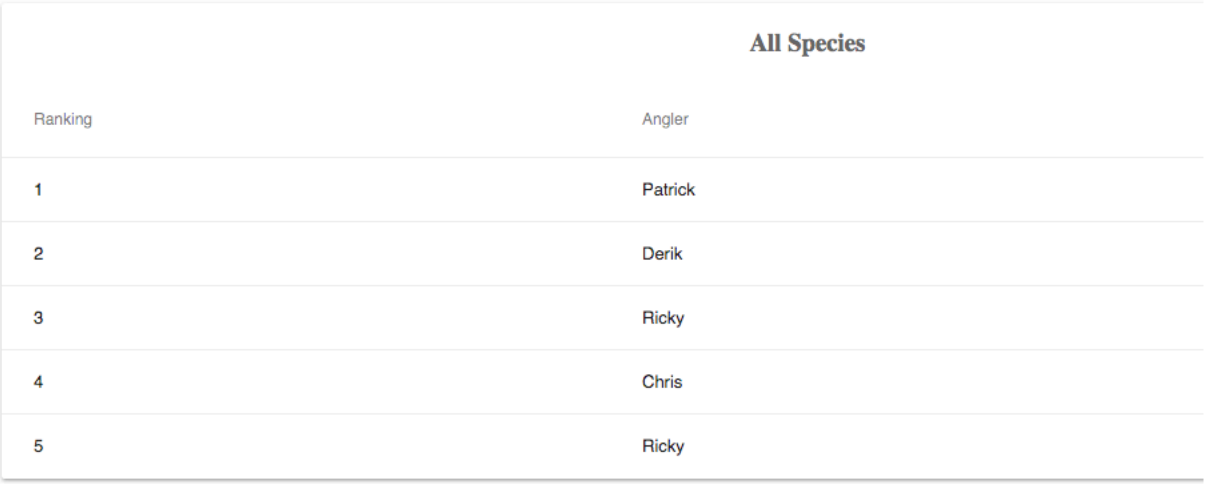
Now that we have covered the technical side of the kata, let's cover the goal to success. I want you to query the given resource and display the data. Pretty easy. The data should be displayed in one of three ways of your choosing.

1. A straight list sorted in descending weight.
2. A list of one species only, sorted in descending weight.
3. An aggregation of the three species so that 3 tables exist, each sorted in descending weights.

Let me repeat, you should choose only one of the three ways to display the data.

I have provided examples of how the data should look for each table. Since this is a leaderboard, we should only display the top 5 fish from any angler. This means an angler may have caught multiple placing fish and each fish should be displayed. Each row of the table should have a `Rank`, `Angler`, and `Weight`.

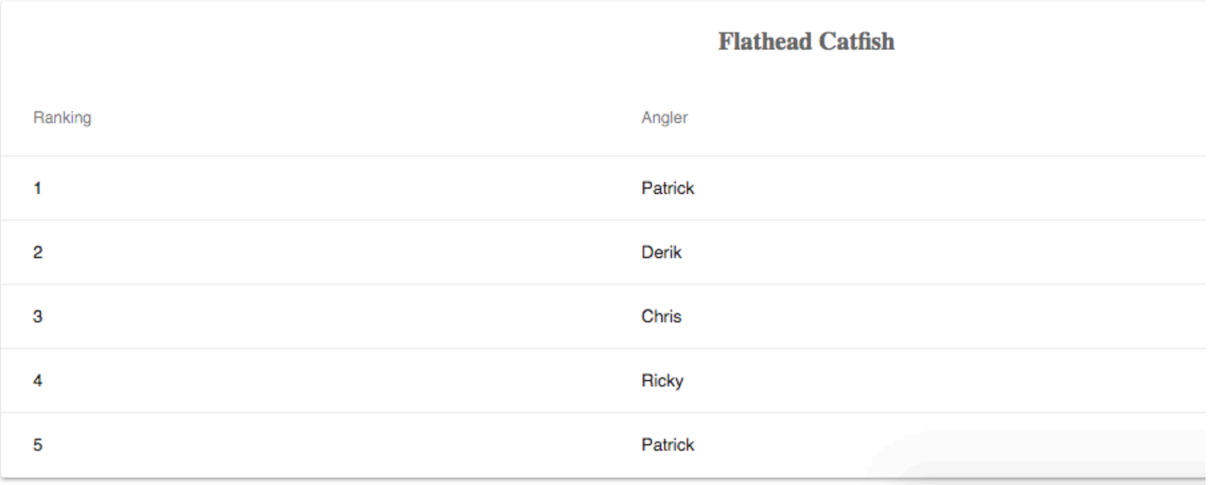
The first example is "All Species" option. If you choose this option, this is the expected outcome.



All Species

Ranking	Angler
1	Patrick
2	Derik
3	Ricky
4	Chris
5	Ricky

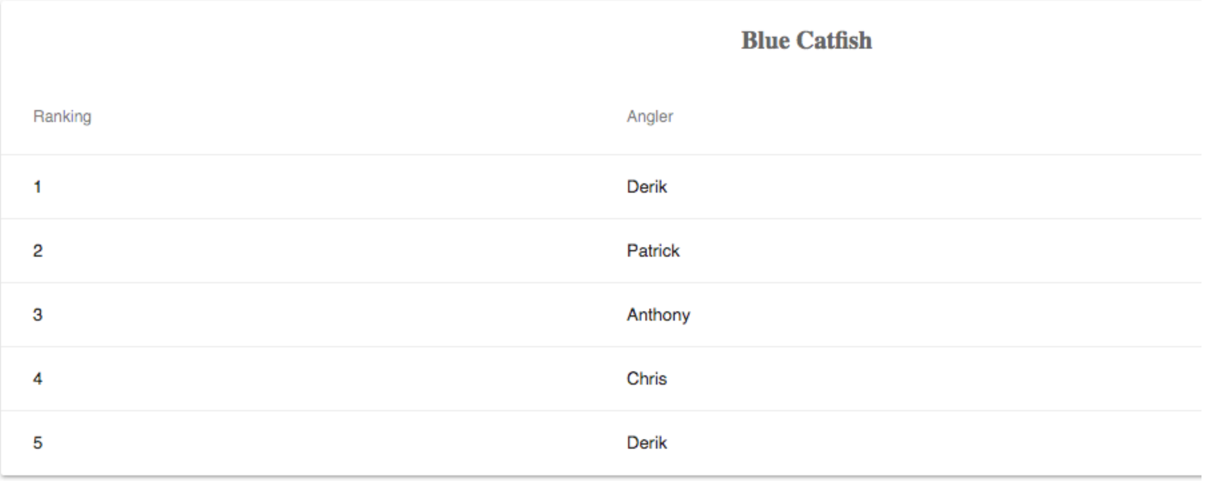
The second example is a singular species. Since flatheads are my favorite, I will show that. Again, this example is the expected outcome.



Flathead Catfish

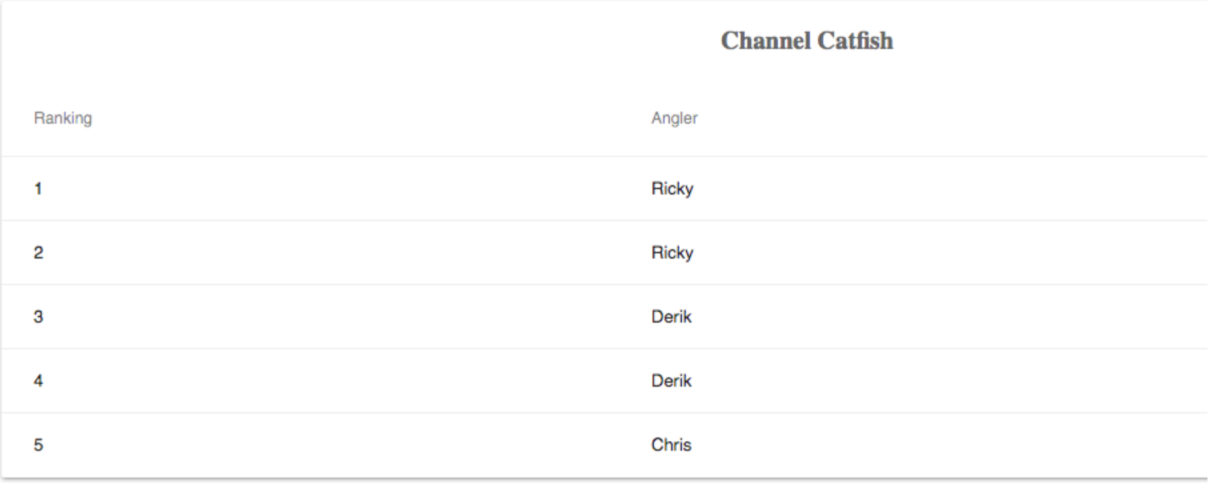
Ranking	Angler
1	Patrick
2	Derik
3	Chris
4	Ricky
5	Patrick

If you choose the third option, you can display them on one page. For the example, I had to screen shot each one of them individually since there was scrolling. But here is the expected result.



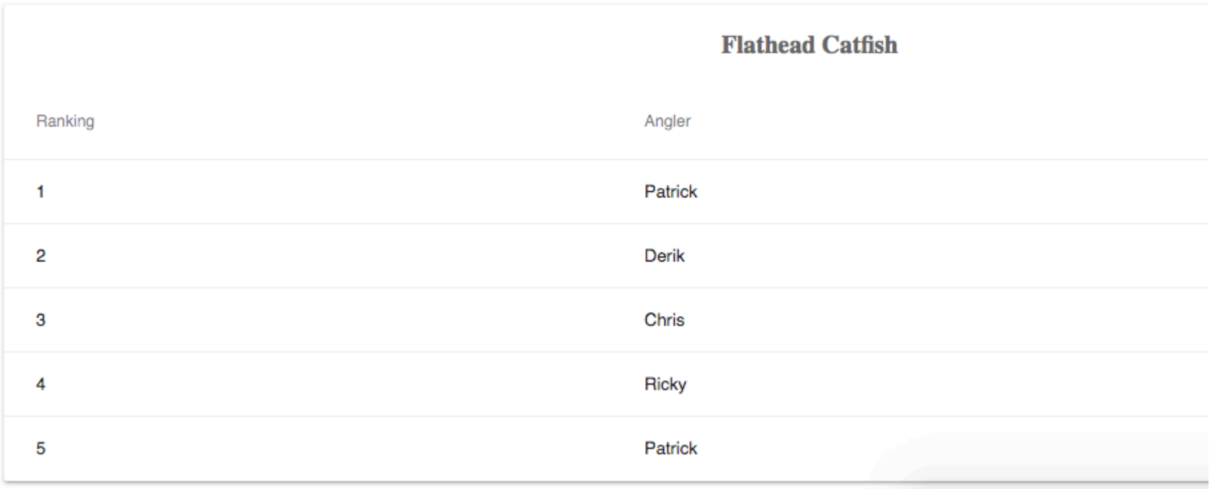
Blue Catfish

Ranking	Angler
1	Derik
2	Patrick
3	Anthony
4	Chris
5	Derik



Channel Catfish

Ranking	Angler
1	Ricky
2	Ricky
3	Derik
4	Derik
5	Chris



Flathead Catfish

Ranking	Angler
1	Patrick
2	Derik
3	Chris
4	Ricky
5	Patrick

Figuring out the weight

To handle the weight of the fish there is a great estimation formula. Its is

```
const weight = length * girth * girth / 800;
```

You can simply use that to solve the problem. *Hint,* `weight` *is a real number and I am only using integers. Coerce the output into an integer.*

Grading

Your kata will be judged on what you do compared to your peers. There is no grade of A, B, etc. *There is the best kata and ones that are not as good as the best, what you choose to submit may reflect well or poorly depending on what work your peers submitted.* What I will be looking for, past your personal interview, is for SOLID principles, OOP pillars, HOC patterns, an understanding of data structures and how well you use them, software engineering principles, etc. Basically all those things you learned at the Galvanize class applied to the React/Redux flow.

Submitting Work

I fully expect you to push your work to a repository. Make sure to collaborate with me so that I have access to your repo if private or confirm I am following it if public. There are three major divisions in this exercise: the action, the reducer, the view. I expect individual commits on those three topics as well as the initial commit that builds the app (`create-react-app code-kata`). Make sure to tag me with `@anthonykulis` clearing explaining your commit. Push to your repository after each topic commit. How well you communicate via the SDLC flow is as important as your code.

Asking Questions

You may ask questions as needed. During normal work, any question is reasonable, but there is a fine line with the kata since part of it's purpose is to demonstrate your skill set. Do not be afraid to ask a question on how something works or why something doesn't compile and run if you are stumped, but at the same time please don't ask me things like what is SOLID and what is not SOLID when you have google available to you. If you do ask a question that I think you should be able to solve on your own, I will tell you. Asking a question does not count against you, so again, do not be afraid to ask. The worst thing that could happen asking a question is me responding with a link to "Let me google that for you".