Paul Cowan

Nomadic cattle rustler and inventor of the electric lasso

- RSS
- Email

Search

Navigate…

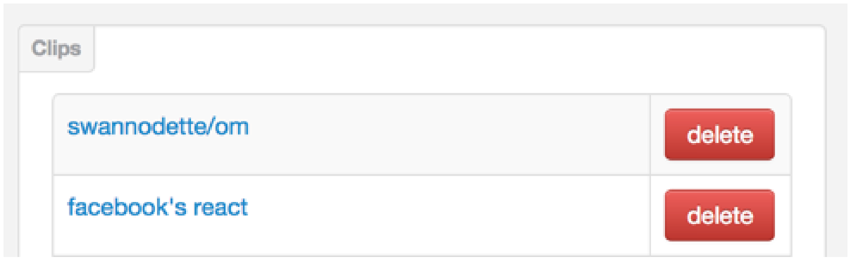
- Blog
- Archives

Clojurescript - Om - Dynamic Components and Unique Keys

Nov 6th, 2014 6:51 am

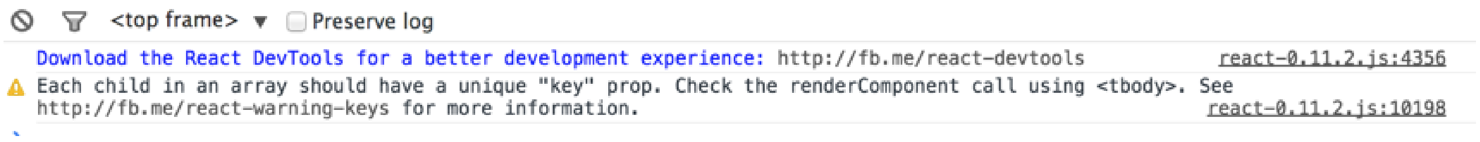
As I get older my tolerance for javascript seems to be getting worse so I've been employing clojurescript as a shield from the true horrors of javascript. I've been playing around wih om which acts as an interface to facebook's react.

While developing with om, I kept getting the same javascript error after rendering a dynamic list like below:



If you have done any developing with om, then I am confident in saying that you will have come across this warning after rendering such a list:

> Each child in an array should have a unique "key" prop. Check the renderComponent call using <tbody>. See http://fb.me/react-warning-keys for more information.



```
Download the React DevTools for a better development experience: http://fb.me/react-devtools          react-0.11.2.js:4356
Each child in an array should have a unique "key" prop. Check the renderComponent call using <tbody>. See
http://fb.me/react-warning-keys for more information.                                                react-0.11.2.js:10198
```

Here is the code I was using to render such a list:

old.cljs

```
(defn clips-view [{:keys [clips]} owner]
  (reify
    om/IRender
    (render [this]
      (html/html
        [:div.well
         [:table.table-bordered.table-hover.table-striped
          [:tbody
           (if (empty? clips)
             [:tr
              [:td.text-center {:colSpan "2"} "No Clips!"]]
             (om/build-all clip-view clips))]]]))))
```

Line 12 in the above is the villain of this piece as there is no way that I could find of passing a func that will set the key property that react uses to identify each dynamic child.

I have been trying to ignore this warning as it did not cause any code to stop executing but I kept seeing this question pop up on the irc channel and I could not find a good answer on the google.

It also turns out that not supplying a react key for each dynamic item could lead to some very unexpected behaviour.

After consulting the om docs I discovered that om/build can take a third `:opts` argument and one of the allowed keys is a `:react-key` which is one of the solutions to the problem.

Armed with this information, I refactored the above code to the following:

refactor.cljs

```
1  (defn clips-view [{:keys [clips]} owner]
2    (reify
3      om/IRender
4      (render [this]
5        (html/html
6          [:div.well
7           [:table.table.table-bordered.table-hover.table-striped
8            [:tbody
9             (if (empty? clips)
10              [:tr
11               [:td.text-center {:colSpan "2"} "No Clips!"]]
12              (map-indexed #(om/build clip-view %2 {:react-key %1}) clips))]]])))))
```

The only change is on line 12:

```
(map-indexed #(om/build clip-view %2 {:react-key %1}) clips))
```

I have used the simplest case of an index for the `:react-key` but if I was rendering from a list where each item had a unique identifier then I would use the `:key` option to specify an element property that would be bound as the react key and would look something like this:

b.clj

```
(map #(om/build clip-view % {:key :id}) clips))
```

(thanks to Anna Pawlicka) for mentioning this in the comments below.

Or probably the best option of is to pass an `:opts` map containing a `:key` option to `om/build-all`:

a.clj

```
(om/build-all clip-view clips {:key :id})
```

Thanks to Daniel Szmulewicz for mentioning this on twitter.

I can now paste a link to this post when somebody asks how to get rid of this warning on the `clojurescript` irc channel.

Posted by Paul Cowan Nov 6th, 2014 6:51 am clojurescript, om

Tweet Like Share Be the first of your friends to like this.

« Clojure - lazy Sequences for the Masses Ember.js - Rendering Dynamic Content »

Comments

3 Comments **The Software Simpleton** 1 Login

Recommend 1 Share Sort by Best

Join the discussion…

mattiasw • 2 years ago

By adding id + :key, I solved this error:
Uncaught Error: No protocol method IDeref.-deref defined for type cljs.core/PersistentArrayMap:
Thanks!

Reply • Share ›

Anna Pawlicka • 3 years ago

Lack of unique keys may actually cause a problem - this post explains the importance of having keys in React:
https://coderwall.com/p/jdy...

It's worth mentioning that you can also use :key in opt. If you're building components out of a sequence, and each item in this sequence has some unique value, e.g.[{:id 56 :foo "bar"} {:id 67 :foo "bar"} ...], you can use :id as a unique key. Option {:key :id} will be used to lookup :id in your data and use it as a React key.

Reply • Share ›

dagda1 Mod → Anna Pawlicka • 3 years ago

cheers, I've updated the post

Reply • Share ›

ALSO ON **THE SOFTWARE SIMPLETON**

Computed Properties and Promises.....don't
1 comment • 2 years ago•
Devin — Misinformation: "data up and actions down". Ember uses "data DOWN; actions UP" Doing things the other way around would make a simple ...

My Failures With Fast Prime Number Generation
1 comment • 2 years ago•
Shannon Severance — If I understand the problem you don't have to look for a prime beyond the 104th. And you can have multiple Ns in a file. It looks like you ...

The Software Simpleton
5 comments • 2 years ago•
thefluxlife — I know this post is kind of old, but I've watched 2 videos, read 1 other blog post, and still felt like I didn't quite get it. I understood the concept but ...

ember.js - Data Down, Actions up…The End of Two Way Data Binding Hell.
1 comment • 2 years ago•
tommyjr — When I first learned Ember, I also

Subscribe Add Disqus to your siteAdd DisqusAdd Privacy

Paul Cowan

I am a.NET/(J)Ruby/JavaScript etc. polyglot programmer and gun for hire. I am currently based in Glasgow (UK).

Contact Me

Recent Posts

- Instance Reducers
- Streams and Async Await in Nodejs
- Animated 3D Pyramid With CSS3 and SASS
- Animating Paths and Arcs With d3.js
- Animating a Sine Wave With d3.js and MathJax

Latest Tweets

- Twitter's busted

Follow @dagda1