

Ruby master - Feature #9970

Add `Hash#map_keys` and `Hash#map_values`

06/21/2014 09:35 PM - seantheprogrammer (Sean Griffin)

Status: Open	
Priority: Normal	
Assignee: nobu (Nobuyoshi Nakada)	
Target version:	
Description These methods simplify two common patterns when working with hashes, and transforming the data. Without map_keys: <pre>Hash[{ a: 1, b: 2 }.map { key, value [key.to_s, value] }]</pre> With map_keys: <pre>{ a: 1, b: 2 }.map_keys(&:to_s)</pre> Without map_values: <pre>Hash[{ a: '1', b: '2' }.map { key, value [key, value.to_i] }]</pre> With map_values: <pre>{ a: '1', b: '2' }.map_values(&:to_i)</pre> The patch, with tests, is attached.	
Related issues:	
Related to Ruby master - Feature #10552: [PATCH] Add Enumerable#frequencies a...	Open 11/27/2014
Related to Ruby master - Feature #12512: Import Hash#transform_values and its...	Closed
Is duplicate of Ruby master - Feature #7793: New methods on Hash	Assigned

History

#1 - 06/21/2014 11:48 PM - phluid61 (Matthew Kerwin)

Duplicate of [#7793](#) ?

--

Matthew Kerwin

<http://matthew.kerwin.net.au/>

#2 - 06/21/2014 11:51 PM - seantheprogrammer (Sean Griffin)

Matthew Kerwin wrote:

Duplicate of [#7793](#) ?

--

Matthew Kerwin

<http://matthew.kerwin.net.au/>

It looks like that was a feature request, and not a patch?

#3 - 06/29/2014 01:07 AM - seantheprogrammer (Sean Griffin)

Ping

#4 - 06/29/2014 03:36 AM - phluid61 (Matthew Kerwin)

A patch that introduces a new feature is still a feature request, and the feature you'd introducing matches one of the stronger suggestions in [#7793](#).

And considering the other ticket has been sitting for a year now, I suspect this is a low priority. :\

#5 - 06/30/2014 04:22 AM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #7793: New methods on Hash added

#6 - 11/29/2014 03:04 AM - duerst (Martin Dürst)

- Related to Feature #10552: [PATCH] Add Enumerable#frequencies and Enumerable#relative_frequencies added

#7 - 09/18/2015 08:59 AM - matz (Yukihiko Matsumoto)

- Assignee set to nobu (Nobuyoshi Nakada)

Hash#map_keys and Hash#map_values may cause confusion. They may have impression of collection of transformed {keys,values}.

But I agree with usefulness of the proposed feature. According to [#7793](#) Hash#transform_keys and Hash#transform_values seem reasonable. Besides that, ActiveSupport also provides transform_keys and transform_values.

Matz.

#8 - 09/22/2015 01:58 AM - seantheprogrammer (Sean Griffin)

Just to be clear, are you saying you would prefer that I edit the patch so that the methods are named #transform_keys and #transform_values?

#9 - 09/22/2015 02:06 AM - matz (Yukihiko Matsumoto)

No, you don't have to. We will take care. Thank you for the proposal.

Matz.

#10 - 06/22/2016 12:15 AM - shyouhei (Shyouhei Urabe)

- Related to Feature #12512: Import Hash#transform_values and its destructive version from ActiveSupport added

#11 - 05/17/2017 10:37 PM - graywolf (Gray Wolf)

What's the status on this? In [#12512](#), only Hash#transform_values was added, should I bother making patch adding Hash#transform_keys (and maybe Hash#transform_pairs)? Or is this ([#9970](#)) still in the process based on last comment from matz ("We will take care [of it].")?

W.

#12 - 05/18/2017 02:08 AM - shyouhei (Shyouhei Urabe)

This issue has no move so far. I think you need to start discussing about transform_keys if you want that. It has one problem though; what happens when the transformed keys conflict is not intuitive. Is that an exception? Or the latter wins?

```
{ x: 1, y: 2 }.transform_keys {|*| :same_key } # should what happen?
```

#13 - 05/18/2017 08:56 PM - graywolf (Gray Wolf)

I think emulating the behaviour of

```
[2] pry(main)> Hash[{ a: 1, b: 2 }.map { |key, value| [:s, value] }]
=> {:s=>2}
```

is reasonable. It's also the same thing Hash#transform_keys from rails does (afaict).

You said

you need to start discussing about transform_keys if you want that

ok, how do I do that? Here? IRC? Mailing list (which one)?

#14 - 05/19/2017 12:41 AM - shyouhei (Shyouhei Urabe)

graywolf (Gray Wolf) wrote:

You said

you need to start discussing about transform_keys if you want that

ok, how do I do that? Here? IRC? Mailing list (which one)?

Thank you, please file a separate issue that requests this method.

Files

hash-map-keys-and-map-values.diff	5.79 KB	06/21/2014	seantheprogrammer (Sean Griffin)
-----------------------------------	---------	------------	----------------------------------