Ruby master - Feature #14371

New option "recursive: true" for Hash#transform_keys!

01/18/2018 05:29 AM - tagomoris (Satoshi TAGOMORI)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		

Description

Hash#transform_keys! is available when we want to symbolize hash keys.
But in some/many cases (for example, receiver hash object is nested configuration tree loaded from any files), hash object are nested object, which has hashes or arrays of hashes as values.
It's super useful if we can transform keys of such nested values.

My proposal is Hash#transform_keys!(recursive: true, &:to_sym). Pseudo code is:

```
def transform_keys!(recursive: false, &block)
  # do original transform_keys! here
  values.each do |v|
    if v.respond_to?(:each)
      v.each{|i| i.transform_keys!(recursive: true, &block) if i.respond_to?(:transform_keys!) }
    else v.respond_to?(:transform_keys!)
      v.transform_keys!(recursive: true, &block)
    end
  end if recursive
end
```

The most major usage example is: config = MyAwesomeFormat.load(file); config.transform_keys!(recursive: true, &:to_sym).

History

#1 - 01/18/2018 05:32 AM - naruse (Yui NARUSE)

- Description updated

#2 - 01/18/2018 05:56 AM - nobu (Nobuyoshi Nakada)

- Description updated

tagomoris (Satoshi TAGOMORI) wrote:

```
    if v.respond_to?(:each)
    else v.respond_to?(:transform_keys!)
```

Why prefer each over transform_keys!?
And probably you wanted to write elsif.

#3 - 01/18/2018 05:59 AM - tagomoris (Satoshi TAGOMORI)

> Why prefer each over transform_keys!?
> And probably you wanted to write elsif.

Your points are correct. The ideal pseudo code is here:

```
def transform_keys!(recursive: false, &block)
  # do original transform_keys! here
  values.each do |v|
    if v.respond_to?(:transform_keys!)
      v.transform_keys!(recursive: true, &block)
    elsif v.respond_to?(:each)
      v.each{|i| i.transform_keys!(recursive: true, &block) if i.respond_to?(:transform_keys!) }
    end
  end if recursive
```

```
end
```

#4 - 01/18/2018 12:19 PM - shevegen (Robert A. Heiler)

I agree with the proposal.

If it is approved, please do not forget meaningful documentation + perhaps
at the least one short example for using it (I refer to the recursive: true
option; http://ruby-doc.org/core-2.5.0/Hash.html#method-i-transform_keys-21).

#5 - 01/19/2018 02:55 PM - nobu (Nobuyoshi Nakada)

https://github.com/nobu/ruby/tree/feature/14371-transform_keys-recursive

#6 - 01/24/2018 06:19 AM - matz (Yukihiro Matsumoto)

I don't think this proposal is a good idea for following reasons:

- transform_keys method to modify value part of Hash
- recursive option to change the behavior this much
- it is much harder to implement transform_keys(recursive:true) (without bang)

I understand the intention behind the proposal. Probably you want to process JSON-like Array-Hash structures. I feel the recent demand to handle those data structures in Ruby. So we might need to add some utility methods in Ruby. I am not sure how yet. Maybe by adding methods to JSON module, or adding them both Array and Hash (like dig).

Matz.

#7 - 01/24/2018 08:39 AM - sakuro (Sakuro OZAWA)

FYI, ActiveSupport implemented these functionalities as separate methods:

- Hash#deep_transform_keys
- Hash#deep_transform_keys!

#8 - 01/25/2018 05:59 AM - tagomoris (Satoshi TAGOMORI)

- Status changed from Open to Rejected

I see. Thank you!