Ruby master - Feature #10208

Passing block to Enumerable#to_h

09/06/2014 04:25 AM - yhara (Yutaka HARA)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	

Description

Now that we can convert 'a list of [key, value] pairs' into a hash with Enumerable#to_h,
how about make it take a block to specify 'how to convert each element into a [key, value] pair'?

Example:

```
# Convert users into an {id => name} hash
users.map{|u| [u.id, u.name]}.to_h
    ↓
# Convert users into an {id => name} hash
users.to_h{|u| [u.id, u.name]}
```

This could also be a solution for these feature requests:

- Feature #6669 A method like Hash#map but returns hash

  ```
  hsh.apply{|k, v| [k.to_s, v]}
  == hsh.to_h{|k, v| [k.to_s, v]}
  ```

- Feature #7793 New methods on Hash
  Feature #9970 Add Hash#map_keys and Hash#map_values

  ```
  hsh.map_k(&:to_s)
  == hsh.to_h{|k, v| [k.to_s, v]}
  hsh.map_v(&:to_i)
  == hsh.to_h{|k, v| [k, v.to_i]}
  hsh.map_kv(&block)
  == hsh.to_h(&block)
  ```

Related issues:

Related to Ruby master - Feature #12512: Import Hash#transform_values and its...	**Closed**
Has duplicate Ruby master - Feature #15143: Extend `Enumerable#to_h`	**Closed**

History

#1 - 09/06/2014 11:16 AM - nobu (Nobuyoshi Nakada)

The name to_h doesn't feel nice for it, IMHO.

#2 - 09/06/2014 12:14 PM - matz (Yukihiro Matsumoto)

I agree with Nobu.

Matz.

#3 - 09/10/2014 01:21 PM - marcandre (Marc-Andre Lafortune)

I agree to_h isn't the right method to do this.

I completely agree that we need new methods to do this.

This would also be a solution for #4151, and the multiple other proposals related to it.

#4 - 09/11/2014 05:29 AM - sawa (Tsuyoshi Sawada)

I think the creation of intermediate arrays for each pair is waste of resource. Can we use combination of two methods? For example, something like:

```
users.with_keys(&:id).with_values(&:name)
```

or

```
users.with_values(&:name).with_keys(&:id)
```

Order of application of the two methods with_keys and with_values should not matter. When only one of them has applied, the return value should be something like an enumerator. As soon as both methods have applied, a hash should be returned.

#5 - 09/11/2014 04:19 PM - yhara (Yutaka HARA)

Marc-Andre Lafortune wrote:

> I completely agree that we need new methods to do this.

Thanks. One idea is name it Enumerable#hash_by (like max_by, group_by)

```
User = Struct.new(:id, :name)
users = [User.new(1, "Alice"), User.new(2, "Bob")]

users.hash_by{|u| [u.id, u.name]}
# {1 => "Alice", 2 => "Bob"}
```

#6 - 09/11/2014 04:43 PM - matz (Yukihiro Matsumoto)

We have #max and #max_by. When we have #hash and #hash_by, people may expect something else.
But it may not matter much. Or maybe we need to rename #hash to #hashcode (off topic here).

Matz.

#7 - 06/22/2016 01:03 PM - knu (Akinori MUSHA)

I like this proposal. I think it is reasonable for to_h to take an optional block to specify how to make a hash from a given enumerable object because most enumerable objects are not composed of two element arrays. (IIRC that was the reason why Matz didn't like the idea of adding the method when it was first proposed)

The said use case users.map{|u| [u.id, u.name]}.to_h is only as effective as Hash[users.map{|u| [u.id, u.name]}], and that spoils the usefulness of the method defined in Enumerable. users.lazy.map{|u| [u.id, u.name]}.to_h may be cool in that it does not create an intermediate array object, but it still creates a couple of chained lazy enumerable objects and looks far from being concise and straightforward.

What do you guys think? I don't think we need a new name for this.

#8 - 07/20/2016 02:20 AM - shyouhei (Shyouhei Urabe)

We looked at this issue at yesterday's developer meeting and had consensus that there is no other example of to_* method that takes a block. Introducing such new concept seems too risky.

#9 - 10/24/2016 05:24 PM - knu (Akinori MUSHA)

So, we need a different name for this.

Here's some candidates I can think of:

- hash_by (proposed above)
- to_h_by
- hash_map
- map_h
- map_to_h

#10 - 10/25/2016 01:36 AM - shyouhei (Shyouhei Urabe)

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

#11 - 10/25/2016 01:42 AM - shyouhei (Shyouhei Urabe)

- Description updated

Not a direct translation of the proposed method, but Hash#transform_values is now available. https://bugs.ruby-lang.org/issues/12512

#12 - 12/07/2016 04:28 AM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Feedback

FYI if there are people who need this functionality: so far we have not found an appropriately sounding name of this method. It should be a method of Enumerable (not only Hash), that takes a block, and returns a Hash.

#13 - 09/20/2018 02:47 AM - nobu (Nobuyoshi Nakada)

- Has duplicate Feature #15143: Extend `Enumerable#to_h` added

#14 - 09/20/2018 02:00 PM - knu (Akinori MUSHA)

So, exactly the same proposal linked above got accepted. The name wasn't actually a problem!

Well, congratulations anyway! □□

#15 - 09/21/2018 02:17 PM - yhara (Yutaka HARA)

- Status changed from Feedback to Closed