

Ruby master - Feature #7340

'each_with' or 'into' alias for 'each_with_object'

11/13/2012 08:17 AM - nathan.f77 (Nathan Broadbent)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		
Description		
Following on from the discussions at #7297 and #7241 , it is apparent that a shorter alias for 'each_with_object' would be much appreciated.		
Related issues:		
Related to Ruby master - Feature #7384: Rename #each_with_object to #each_with	Open	
Related to Ruby master - Feature #7297: map_to alias for each_with_object	Rejected	11/07/2012
Related to Ruby master - Feature #6687: Enumerable#with	Open	

History

#1 - 11/13/2012 11:06 AM - matz (Yukihiko Matsumoto)

I dislike #into because it may or may not put something into the argument.
I am OK with #each_with.

Matz.

#2 - 11/13/2012 11:24 AM - merborne (kyo endo)

matz (Yukihiko Matsumoto) wrote:

I dislike #into because it may or may not put something into the argument.
I am OK with #each_with.

Matz.

I would appreciate if you could look at [#6687](#). but #each_with is OK for me :-)

#3 - 11/13/2012 08:25 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

The reason I dislike each_with_object and each_with is the "each" on them. "each"'s return value isn't meaningful to me. That's why I would prefer "map_to", "map_into" or just "into".

"into" here doesn't mean putting values "into" the array or hash. It means: "turn/map object into a hash/array/whatever". It is not about putting something into the passed object, but about converting the original object (say numbers) into the object being passed as the initial value like a hash or an array.

It is ok to chain operations as a pattern (like the one used by jQuery) so that an_array.each{...}.sort would justify "each" returning the original "an_array", but only to be able to chain operations and not because "each" implies returning anything meaningful.

In that sense, each_with_object is currently supposed to return some meaningful value. That is why I'd prefer to call it "map_to", "map_into" or just "into".

numbers.map_into({}){...} should read "map numbers into a hash where ...". It would be even shorter if we just abbreviated "map_into" as just "into".

Also, if there is any chance that this wouldn't be an alias to each_with_object, I'd prefer the block's arguments order to be inverted to be symmetric to "inject".

#4 - 11/13/2012 08:53 PM - phluid61 (Matthew Kerwin)

On 13 November 2012 21:25, rosenfeld (Rodrigo Rosenfeld Rosas) <rr.rosas@gmail.com> wrote:

Issue [#7340](#) has been updated by rosenfeld (Rodrigo Rosenfeld Rosas).

The reason I dislike each_with_object and each_with is the "each" on them. "each"'s return value isn't meaningful to me. That's why I would prefer "map_to", "map_into" or just "into".

"into" here doesn't mean putting values "into" the array or hash. It means: "turn/map object into a hash/array/whatever". It is not about putting something into the passed object, but about converting the original object (say numbers) into the object being passed as the initial value like a hash or an array.

It is ok to chain operations as a pattern (like the one used by jQuery) so that `an_array.each {...}.sort` would justify "each" returning the original "an_array", but only to be able to chain operations and not because "each" implies returning anything meaningful.

In that sense, `each_with_object` is currently supposed to return some meaningful value. That is why I'd prefer to call it "map_to", "map_into" or just "into".

`numbers.map_into({}){...}` should read "map numbers into a hash where ...". It would be even shorter if we just abbreviated "map_into" as just "into".

Also, if there is any chance that this wouldn't be an alias to `each_with_object`, I'd prefer the block's arguments order to be inverted to be symmetric to "inject".

I believe, given that explanation, that `#map_to` is a far more appropriate name than `#map_into` . . . and suddenly the reservations I had about this alias start to fade away.

There is a clear semantic distinction between `#each_with`, where the focus is on the action performed in the block, and `#map_to`, where the focus is on the object returned from the block. So the question is: is this alias/method meant to be an analogue for `#each`, or for `#map` ?

--

Matthew Kerwin, B.Sc (CompSci) (Hons)

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

ABN: 59-013-727-651

"You'll never find a programming language that frees you from the burden of clarifying your ideas." - xkcd

#5 - 05/10/2014 03:21 PM - akr (Akira Tanaka)

- Related to Feature #7384: Rename `#each_with_object` to `#each_with_added`

#6 - 05/10/2014 03:22 PM - akr (Akira Tanaka)

- Related to Feature #7297: `map_to` alias for `each_with_object` added

#7 - 05/10/2014 03:23 PM - akr (Akira Tanaka)

- Related to Feature #6687: `Enumerable#with` added

#8 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)