

## Ruby master - Feature #15822

### Add Hash#except

05/04/2019 01:50 PM - timoschilling (Timo Schilling)

<b>Status:</b>	Open	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Target version:</b>		
<b>Description</b>		
According to Hash#slice, I would like to add Hash#except. Same for ENV.except.		
<b>Related issues:</b>		
Related to Ruby master - Feature #13563: Implement Hash#choice method.		<b>Closed</b>
Is duplicate of Ruby master - Feature #8499: Importing Hash#slice, Hash#slice...		<b>Closed</b>

### History

#### #1 - 05/05/2019 03:37 AM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #8499: Importing Hash#slice, Hash#slice!, Hash#except, and Hash#except! from ActiveSupport added

#### #2 - 05/06/2019 06:25 PM - timoschilling (Timo Schilling)

- File deleted (hash-except.patch)

#### #3 - 05/06/2019 06:26 PM - timoschilling (Timo Schilling)

- File hash-except.patch added

#### #4 - 10/17/2019 06:34 AM - matz (Yukihiro Matsumoto)

- Related to Feature #13563: Implement Hash#choice method. added

#### #5 - 10/17/2019 06:42 AM - matz (Yukihiro Matsumoto)

Hash#slice was introduced via discussion in [#13563](#). We didn't see the need for Hash#except yet. Any (real world use-case)? I don't think the name except is the best name for the behavior.

Matz.

#### #6 - 10/17/2019 02:32 PM - zverok (Victor Shepelev)

[matz \(Yukihiro Matsumoto\)](#) About use-cases, there are multiple, like:

- Logging/debug printing some hash with sensitive or too large data excluded:

```
puts "REQUEST: #{request_data.except(:body, :apikey)}
```

- Data tidying:

```
CSV.read('data.csv', headers: true).map(&:to_h).map { |row| OpenStruct.new(row.except("Manager comment")) }
```

- Data conversion:

```
paragraphs.map { |para| ParaStats.new(para.to_h.except(:text)) }
```

...and so on (without even mentioning working with models in Rails, where it is used extensively to, for example, copy model without its id and timestamps).

Redmine's sources, for example, make some use of it:

```
$ grep -F ".except(" {app,lib} -r | wc -l
27
```

I recently [stumbled upon it, too](#).

About the name: I understand "ActiveSupport uses this name" is kinda lazy one, but it does :) And, TBH, for me (even before I started working with Rails, I used Ruby for first my 12 years without ActiveSupport) it always felt natural: request.except(:path, :body) requires no further explanations. Facets [uses the name](#), too.

The alternative method name I can think of is, maybe, descriptive without\_keys(\*keys) (or except\_keys, or just without(\*keys)), but I am not sure it is any better.

## Files

---

hash-except.patch	4.05 KB	05/06/2019	timoschilling (Timo Schilling)
-------------------	---------	------------	--------------------------------