
Babosa



Babosa is a library for creating human-friendly identifiers, aka “slugs”. It can also be useful for normalizing and sanitizing data.

It is an extraction and improvement of the string code from FriendlyId. I have released this as a separate library to help developers who want to create libraries similar to FriendlyId.

Features / Usage

Transliterate UTF-8 characters to ASCII

```
1 "Gölcük, Turkey".to_slug.transliterate.to_s #=> "Gołcuk, Turkey"
```

Locale sensitive transliteration, with support for many languages

```
1 "Jürgen Müller".to_slug.transliterate.to_s      #=> "Jurgen Muller"
2 "Jürgen Müller".to_slug.transliterate(:german).to_s  #=> "Juergen Mueller"
```

Currently supported languages include:

- Bulgarian
- Danish
- German
- Greek
- Hindi
- Macedonian
- Norwegian
- Romanian
- Russian
- Serbian
- Spanish
- Swedish
- Turkish
- Ukrainian
- Vietnamese

Additionally there are generic transliterators for transliterating from the Cyrillic alphabet and Latin alphabet with diacritics. The Latin transliterator can be used, for example, with Czech. There is also a transliterator named “Hindi” which may be sufficient for other Indic languages using Devanagari, but I do not know enough to say whether the transliterations would make sense.

I’ll gladly accept contributions from fluent speakers to support more languages.

Strip non-ASCII characters

```
1 "Gölcük, Turkey".to_slug.to_ascii.to_s #=> "Glck, Turkey"
```

Truncate by characters

```
1 "üüü".to_slug.truncate(2).to_s #=> "üü"
```

Truncate by bytes

This can be useful to ensure the generated slug will fit in a database column whose length is limited by bytes rather than UTF-8 characters.

```
1 "üüü".to_slug.truncate_bytes(2).to_s #=> "ü"
```

Remove punctuation chars

```
1 "this is, um, **really** cool, huh?".to_slug.word_chars.to_s #=> "this  
is um really cool huh"
```

All-in-one

```
1 "Gölcük, Turkey".to_slug.normalize.to_s #=> "gölcuk-turkey"
```

Other stuff

Using Babosa With FriendlyId 4+

```
1 require "babosa"  
2  
3 class Person < ActiveRecord::Base  
4   friendly_id :name, use: :slugged
```

```
5
6   def normalize_friendly_id(input)
7     input.to_s.to_slug.normalize(transliterations: :russian).to_s
8   end
9 end
```

UTF-8 support Babosa normalizes all input strings to NFC.

Ruby Method Names Babosa can generate strings for Ruby method names. (Yes, Ruby 1.9+ can use UTF-8 chars in method names, but you may not want to):

```
1 "this is a method".to_slug.to_ruby_method! #=> this_is_a_method
2 "über cool stuff!".to_slug.to_ruby_method! #=> uber_cool_stuff!
3
4 # You can also disallow trailing punctuation chars
5 "über cool stuff!".to_slug.to_ruby_method(allow_bangs: false) #=>
  uber_cool_stuff
```

Easy to Extend You can add custom transliterators for your language with very little code. For example here's the transliterator for German:

```
1 module Babosa
2   module Transliterator
3     class German < Latin
4       APPROXIMATIONS = {
5         "ä" => "ae",
6         "ö" => "oe",
7         "ü" => "ue",
8         "Ä" => "Ae",
9         "Ö" => "Oe",
10        "Ü" => "Ue"
11      }
12    end
13  end
14 end
```

And a spec (you can use this as a template):

```
1 require "spec_helper"
2
3 describe Babosa::Transliterator::German do
4   let(:t) { described_class.instance }
5   it_behaves_like "a latin transliterator"
6
7   it "should transliterate Eszett" do
8     t.transliterate("ß").should eql("ss")
9   end
10 end
```

```
9   end
10
11   it "should transliterate vowels with umlauts" do
12     t.transliterate("üöä").should eql("ueoeae")
13   end
14 end
```

Rails 3.x and higher

Some of Babosa’s functionality was added to Active Support 3.0.0.

Babosa now differs from ActiveSupport primarily in that it supports non-Latin strings by default, and has per-locale ASCII transliterations already baked-in. If you are considering using Babosa with Rails, you may want to first take a look at ActiveSupport’s `transliterate` and `parameterize` to see if they suit your needs.

Please see the API docs and source code for more info.

Getting it

Babosa can be installed via Rubygems:

```
1 gem install babosa
```

You can get the source code from its Github repository.

Reporting bugs

Please use Babosa’s Github issue tracker.

Misc

“Babosa” means “slug” in Spanish.

Maintainers

- Philip Arndt
- Norman Clarke

Contributors

Many thanks to the following people for their help:

- Dmitry A. Ilyashevich - Deprecation fixes
- anhkind - Vietnamese support
- Martins Zakis - Bug fixes
- Vassilis Rodokanakis - Greek support
- Peco Danajlovski - Macedonian support
- Philip Arndt - Bug fixes
- Jonas Forsberg - Swedish support
- Jaroslav Kalistsuk - Greek support
- Steven Heidel - Bug fixes
- Edgars Beigarts - Support for multiple transliterators
- Tiberiu C. Turbureanu - Romanian support
- Kim Joar Bekkelund - Norwegian support
- Alexey Shkolnikov - Russian support
- Martin Petrov - Bulgarian support
- Molte Emil Strange Andersen - Danish support
- Milan Dobrota - Serbian support
- Norman Clarke - Original author

Copyright

Copyright (c) 2010-2021 Norman Clarke and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT

OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.