

Join the Stack Overflow Community

Stack Overflow is a community of 7.2 million programmers, just like you, helping each other. Join them; it only takes a minute:

Sign up

Rails 4: List of available datatypes


Where can I find a list of data types that can be used in Ruby on Rails 4? Such as

- text
- string
- integer
- float
- date


I keep learning about new ones and I'd love to have a list I could easily refer to.

ruby-on-rails   ruby   ruby-on-rails-4   rails-activerecord

edited Feb 24 '16 at 8:48

 Flip  
1,558   2   11   31

asked Jul 29 '13 at 7:19

 Nicolas Raoul  
27.4k   38   129   239

- possible duplicate of [Where is the documentation page for ActiveRecord data types?](#) – Mark Thomas Mar 29 '14 at 1:51
- 1 @MarkThomas: Not a duplicate. My question here is specifically for Rails 4. – Nicolas Raoul Mar 30 '14 at 1:51
- The accepted answer in that question is essentially the same as your accepted answer. Both point to the Rails 4 docs. Also, it's a moot point because ActiveRecord datatypes didn't change from Rails 3 to Rails 4. – Mark Thomas Mar 30 '14 at 19:00
- 8 I for one did not know AR datatypes didn't change between rails 3 and 4 so I'm thankful this question/answer is here. – Dty Apr 12 '14 at 22:49

5 Answers

Here are all the Rails 4 (ActiveRecord migration) datatypes:

- :binary
- :boolean
- :date
- :datetime
- :decimal
- :float
- :integer
- :bigint
- :primary\_key
- :references
- :string
- :text
- :time

- :timestamp

Source:

[http://api.rubyonrails.org/classes/ActiveRecord/ConnectionAdapters/SchemaStatements.html#method-i-add\\_column](http://api.rubyonrails.org/classes/ActiveRecord/ConnectionAdapters/SchemaStatements.html#method-i-add_column)

These are the same as with [Rails 3](#).

If you use PostgreSQL, you can also take advantage of these:

- :hstore
- :json
- :jsonb
- :array
- :cidr\_address
- :ip\_address
- :mac\_address

They are stored as strings if you run your app with a not-PostgreSQL database.

**Edit, 2016-Sep-19:**

There's [a lot more](#) postgres specific datatypes in Rails 4 and [even more](#) in Rails 5.

edited 2 days ago



Community ♦

1 1

answered Jul 29 '13 at 7:19



Nicolas Raoul

27.4k 38 129 239

1 I believe these are the datatypes that are supported across all databases. However as Peter de Ridder mentions there are some datatypes like hstore which are still supported. – [Althaf Hameez](#) Jul 29 '13 at 8:10

Question: postgres documentation doesn't seem to have text data type. Yet, rails can still handle it? What goes on in the background? – [ahnbizcad](#) Aug 20 '14 at 9:46

PostgreSQL does have a text datatype. Under the hood all varchar/text fields are variable length arrays. [postgresql.org/docs/9.3/interactive/datatype-character.html](http://postgresql.org/docs/9.3/interactive/datatype-character.html) – [makhan](#) Aug 22 '14 at 4:59

I'd like to add that if you're using a non-postgres database and your application schema\_format is not set to use :sql, then your schema.rb file wont be able to dump the table that uses types like :json. The schema will still be dumped for the tables that use default types but you'll see a comment for the table with special types like, "could not dump table...". [Look here to set the schema\\_format](#). – [bpercevic](#) Feb 18 '15 at 19:19

Also, those columns will have type nil in a non-postgres database. You can inspect the type in the console with `Model.columns_hash["column_name"].type`. These are just things that I've run into when using :json column type, I may be wrong and this may not happen to everyone but I thought I'd let future readers know in case they have troubles. Regardless, +1 for this answer because it really helped me. – [bpercevic](#) Feb 18 '15 at 19:26

You might also find it useful to know generally what these data types are used for:

- :string - is for small data types such as a title. ([Should you choose string or text?](#))
- :text - is for longer pieces of textual data, such as a paragraph of information
- :binary - is for storing data such as images, audio, or movies.
- :boolean - is for storing true or false values.
- :date - store only the date
- :datetime - store the date and time into a column.
- :time - is for time only
- :timestamp - for storing date and time into a column. ([What's the difference between datetime and timestamp?](#))
- :decimal - is for decimals ([example of how to use decimals](#)).
- :float - is for decimals. ([What's the difference between decimal and float?](#))
- :integer - is for whole numbers.
- :primary\_key - unique key that can uniquely identify each row in a table

There's also references used to create associations. But, [I'm not sure this is an actual data type](#).

New Rails 4 datatypes available in PostgreSQL:

- :hstore - storing key/value pairs within a single value ([learn more about this new data type](#))
- :array - an arrangement of numbers or strings in a particular row ([learn more about it and see examples](#))
- :cidr\_address - used for IPv4 or IPv6 host addresses
- :inet\_address - used for IPv4 or IPv6 host addresses, same as cidr\_address but it also accepts values with nonzero bits to the right of the netmask

- Learn more about the address datatypes [here](#) and [here](#).

Also, here's the official guide on migrations: <http://edgeguides.rubyonrails.org/migrations.html>

answered Mar 29 '14 at 1:19



1 1

2.745

2.745 1 12 22

- 9 Applause. +1 for thoroughness, and anticipation of usage. That's UX mentality right there. — [ahnbizcad](#) Aug 20 '14 at 9:47
- 3 Absolutely terrific answer - many thanks for this. The links to the articles on differences literally took the questions right out of my mouth. — [nlh](#) Sep 11 '14 at 0:07
- 
- 1 For Postgres there is additionally a `uuid` type which can be used as *normal field* like `t.uuid :name...` or as *primary key* like `create_table :users, id: :uuid do... or.e.g. t.primary_key :id, :uuid, :default => 'uuid_generate_v1()'`  — [TNT](#) Jul 28 '15 at 11:33
- 
- 1 Additional PostgreSQL types supported by Rails listed in the [API docs](#) for [ActiveRecord::ConnectionAdapters::PostgreSQL::ColumnMethods](#) . Highlights include `money` , `json` , `xml` , `daterange` — [Eliot Sykes](#) Aug 21 '15 at 15:09

	db2	mysql	openbase	oracle
:binary	blob(32768)	blob	object	blob
:boolean	decimal(1)	tinyint(1)	boolean	number(1)
:date	date	date	date	date
:datetime	timestamp	datetime	datetime	date
:decimal	decimal	decimal	decimal	decimal
:float	float	float	float	number
:integer	int	int(11)	integer	number(38)
:string	varchar(255)	varchar(255)	char(4096)	varchar2(255)
:text	clob(32768)	text	text	clob
:time	time	time	time	date
:timestamp	timestamp	datetime	timestamp	date

	postgresql	sqlite	sqlserver	sybase
:binary	bytea	blob	image	image
:boolean	boolean	boolean	bit	bit
:date	date	date	date	datetime
:datetime	timestamp	datetime	datetime	datetime
:decimal	decimal	decimal	decimal	decimal
:float	float	float	float(8)	float(8)
:integer	integer	integer	int	int
:string	(note 1)	varchar(255)	varchar(255)	varchar(255)
:text	text	text	text	text
:time	time	datetime	time	time
:timestamp	timestamp	datetime	datetime	timestamp

edited Feb 15 '16 at 17:44

answered Sep 6 '14 at 17:02

gotqn



11.9k2683159

- 1

I would be interested to see the json type in this table, if you feel up to it – [Freedom\\_Ben](#) Jan 22 '15 at 19:08
- 2


You can always check the `NATIVE_DATABASE_TYPES` for the adapter you need - [github.com/rails/rails/blob/master/activerecord/lib/...](#) – [gotqn](#) Jan 22 '15 at 21:11
- Is there a spot on the guides or a single source of truth for these things over time? – [Hugo](#) Jun 26 '15 at 18:12
- 1

@Hugo [pragprog.com/book/rails4/agile-web-development-with-rails-4](#) – [gotqn](#) Feb 15 '16 at 17:44

You can access this list everytime you want (even if you don't have Internet access) through:

```
rails generate model -h
```

edited Mar 7 '16 at 13:43

 [tomascharad](#)  
1,6401118

answered May 8 '15 at 20:56

Rails4 has some added datatypes for Postgres.  
For example, railscast #400 names two of them:

Rails 4 has support for native datatypes in Postgres and we'll show two of these here, although a lot more are supported: array and hstore. We can store arrays in a string-type column and specify the type for hstore.

Besides, you can also use cidr, inet and macaddr. For more information:  
<https://blog.engineyard.com/2013/new-in-rails-4>

answered Jul 29 '13 at 7:26

 [Peter de Ridder](#)  
1,9731015