

# MONGO ON DEMAND

## NOSQL IN POSTGRESQL

# max

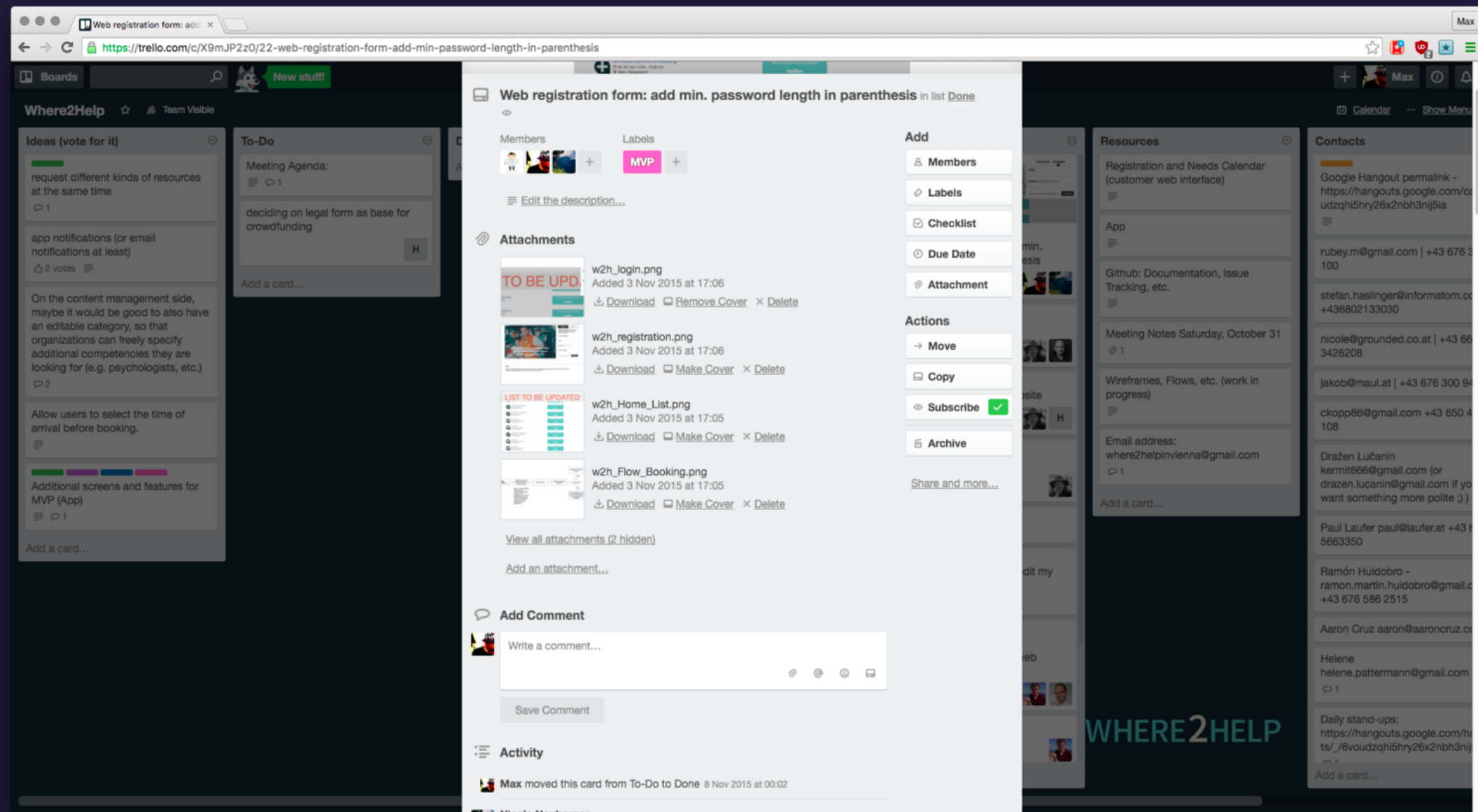
 |  klappradla

# NoSQL

"...data which is modeled in means other than the tabular relations.."

*wikipedia*

**WHY?**





```
{  "id": "1000000000000000",
  "email": "vienna@rb.com",
  "first_name": "Vienna",
  "gender": "other",
  "last_name": "Rb",
  "link": "https://www.facebook.com/vienna.rb",
  "locale": "de_AT",
  "name": "Vienna Rb",
  "timezone": "2",
  "updated_time": "2014-03-31T08:26:19+0000",
  "username": "viennarb",
  "verified": "false"
}
```

```
{
  "username": "vienna-rb",
  "bio": "",
  "website": "",
  "profile_picture": "http://instagram.com/123.jpg",
  "full_name": "",
  "counts": {"media": 1, "followed_by": 0, "follows": 0},
  "id": "1000000001"
}
```

```
{
  "preferences":{
    "newsletter":false,
    "web_notifications":[
      "Notifications::CommentOnUsersPost",
      "Notifications::AlsoCommentedPost",
      "Notifications::NewWallComment"
    ],
    "immediate_mail_notifications":[
      "Notifications::AlsoCommentedPost",
      "Notifications::LocationApproved",
      "Notifications::MeetingUpdated",
      "Notifications::NewWallComment"
    ],
    "daily_mail_notifications":[
      "Notifications::NewUserPost",
      "Notifications::NewLocationPost",
      "Notifications::NewMeeting"
    ]
  }
}
```



**MONGO**

**in**

**POSTGRES**

```
# Serialize a preferences attribute.
```

```
class User < ActiveRecord::Base  
  serialize :preferences  
end
```

```
# Serialize preferences using JSON as coder.
```

```
class User < ActiveRecord::Base  
  serialize :preferences, JSON  
end
```

```
# Serialize preferences as Hash using YAML coder.
```

```
class User < ActiveRecord::Base  
  serialize :preferences, Hash  
end
```

Active Record and PostgreSQL X

edgeguides.rubyonrails.org/active\_record\_postgresql.html

Max

Back to rubyonrails.org

RAILSGUIDES

Home Guides Index ↕ Contribute Credits

edge

# Active Record and PostgreSQL

This guide covers PostgreSQL specific usage of Active Record.

After reading this guide, you will know:

- ✔ How to use PostgreSQL's datatypes.
- ✔ How to use UUID primary keys.
- ✔ How to implement full text search with PostgreSQL.
- ✔ How to back your Active Record models with database views.

## Chapters

- Datatypes**
  - Bytea
  - Array
  - Hstore
  - JSON
  - Range Types
  - Composite Types
  - Enumerated Types
  - UUID
  - Bit String Types
  - Network Address Types
  - Geometric Types
- UUID Primary Keys**
- Full Text Search**
- Database Views**

In order to use the PostgreSQL adapter you need to have at least version 8.2 installed. Older versions are not supported.

To get started with PostgreSQL have a look at the [configuring Rails guide](#). It describes how to properly setup Active Record for PostgreSQL.

## 1 Datatypes

PostgreSQL offers a number of specific datatypes. Following is a list of types, that are supported by the PostgreSQL adapter.

### 1.1 Bytea

- type definition
- functions and operators

```
# db/migrate/20140207133952_create_documents.rb
create_table :documents do |t|
  t.binary 'payload'
end

# app/models/document.rb
class Document < ApplicationRecord
end
```

# JSON

DOUBPLICATED KEYS,  
PRESERVE ORDERING

# JSONB

INDEXES, FUNCTIONS

# CREATE

```
CREATE TABLE users (  
    id serial not null,  
    settings json not null default '{}',  
    preferences jsonb not null default '{}'  
);
```

# INSERT

```
INSERT INTO users (settings) VALUES (1,  
    '{  
        "newsletter":true,  
        "interests":[  
            "biking",  
            "programming"  
        ]  
    }'  
);
```

# QUERY

```
SELECT count(*)  
FROM users  
WHERE preferences ->> 'newsletter' = 'true';
```

```
SELECT count(*)  
FROM users  
WHERE preferences -> 'interests' ? 'biking';
```

```
SELECT count(*)  
FROM users  
WHERE preferences @> '{"newsletter":true}';
```

# INDEX

```
CREATE INDEX idxpreferences_interests_exp  
ON users ((preferences->'interests'));
```

```
CREATE INDEX idxpreferences_news_exp  
ON users ((preferences->>'newsletter'));
```

```
CREATE INDEX idxpreferences_gin  
ON users USING GIN (preferences);
```



DEMO

RAILS?

# CREATE

```
class CreateUsers < ActiveRecord::Migration
  def change

    create_table :users do |t|
      t.string :name, null: false
      t.jsonb :preferences, null: false, default: '{} '
    end

    add_index :users, :preferences, using: :gin
  end
end
```

# USE

```
class User < ActiveRecord::Base

  scope :programmers, -> {
    where('preferences @> ?',
          {interests: ['programming']}.to_json
        )
  }

end
```

```
user.preferences['newsletter'] = true
user.save
```

#=> updates whole preferences document

# ACCESS

```
class User < ActiveRecord::Base
  store_accessor :preferences, :newsletter, :interests
end
```

```
user = User.new(newsletter: false, interests: ['runninng'])
```

```
user.preferences
#=> {"newsletter" ...
```

```
user.newsletter
#=> false
```

```
user.preferences[:newsletter]
#=> false
```

DEMO