





⚡ Mastering Ruby on Rails Performance: 4 Pro Tips for IT Wizards! ⚙️



diffiter.com

Introduction :

 Welcome to the **Ruby on Rails** fast track!  In this guide, we're spilling the secrets to make your app a speed superstar, catching the eyes of IT pros worldwide. Brace yourself for an exciting journey as we reveal four coding tricks that'll not only turbocharge your app but also lock it down securely!  

1) Caching Magic: Turbocharge Your App's Speed 🚀

Caching is your secret weapon! Implement the Rails.cache to store query results and boost page load times.

jsgrip.com

```
# Example caching in Rails controller action
@data = Rails.cache.fetch('unique_cache_key') do
  # Your expensive query or computation
  SomeModel.expensive_operation
end
```

2) Database Optimization: Mastering the Art of Efficient Queries 🎨

Craft optimized queries and use indexes judiciously. Active Record's includes and joins can be your best friends.


jsgrip.com



```
# Efficient querying with includes
@posts = Post.includes(:comments).where('comments.approved' => true)
```

3) Load Balancing Act: Scaling Your App Horizontally

Scale horizontally with load balancing. Tools like **Nginx** or **HAProxy** can help distribute traffic across multiple servers.



```
# Example Nginx configuration for load balancing
upstream backend {
    server backend1.example.com;
    server backend2.example.com;
}
```

4) SEO-Friendly Rails: Elevate Your Visibility

Optimize your routes, meta tags, and use clean **URLs** for better SEO. Leverage gems like *friendly_id* for cleaner slugs.

```
jsrgrip.com

# Using friendly_id for clean slugs
class Article < ApplicationRecord
  extend FriendlyId
  friendly_id :title, use: :slugged
end
```