A Brain-Friendly Guide

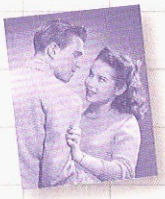
Head First Rails

A Learner's Companion to Ruby on Rails

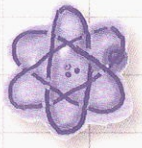
Master your data with Rails finders

Integrate your apps with Google Maps and never get lost again

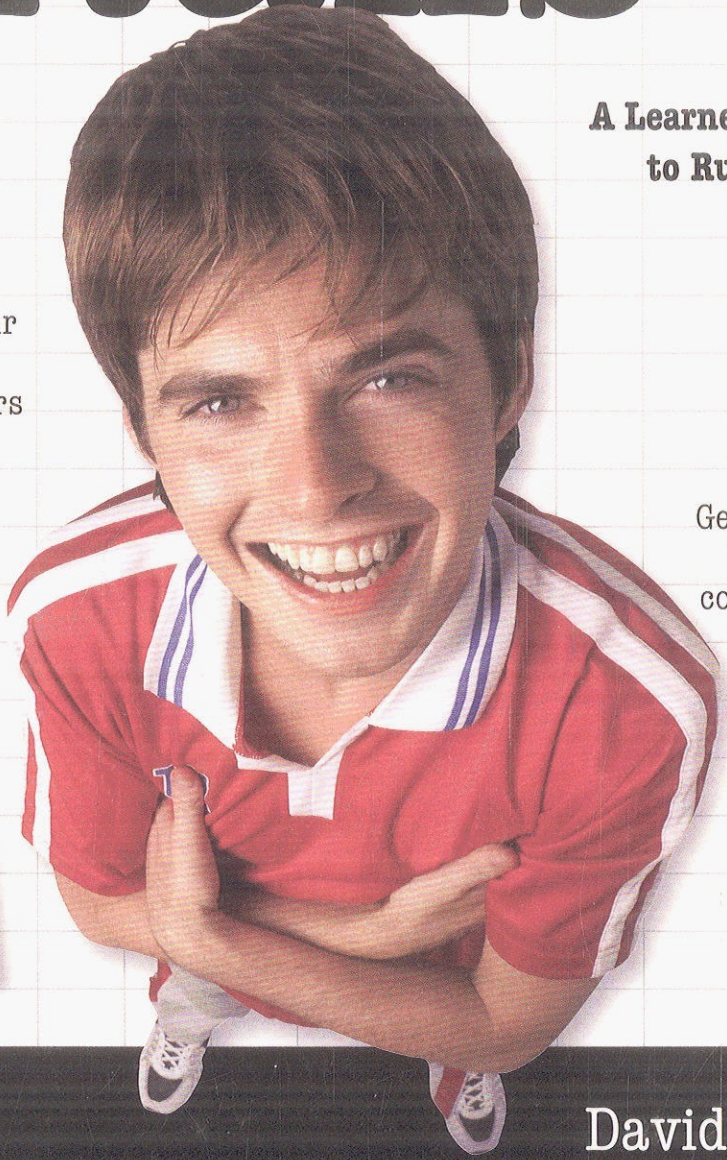
Learn how Suzy validated all her dates and avoided another awful evening



Get inside Embedded Ruby, and take control of your apps



Add Ajax and XML to your Rails universe



O'REILLY®

David Griffiths

Table of Contents (Summary)

Table of Contents (the real thing)

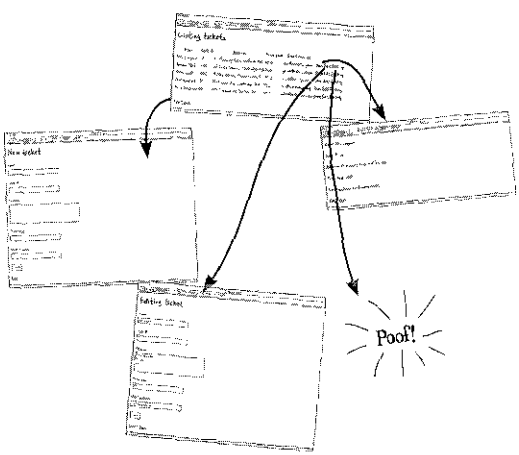
Intro

Your brain on Rails. Here *you* are trying to *learn* something, while here your *brain* is doing you a favor by making sure the learning doesn't *stick*. Your brain's thinking, "Better leave room for more important things, like which wild animals to avoid and whether naked snowboarding is a bad idea." So how *do* you trick your brain into thinking that your life depends on knowing Rails?

getting started

1 Really Rapid Rails

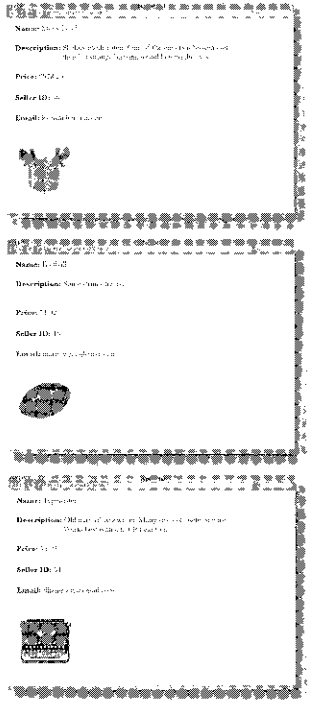
Want to get your web app development off to a flying start? Then you need to know **Rails**. Rails is the **coolest** and **quickest development framework** in town, allowing you to develop **fully functional web apps** quicker than you ever thought possible. Getting started is simple; all you need to do is **install Rails**, and start turning the pages. Before you know it, you'll be **miles ahead of the competition**.

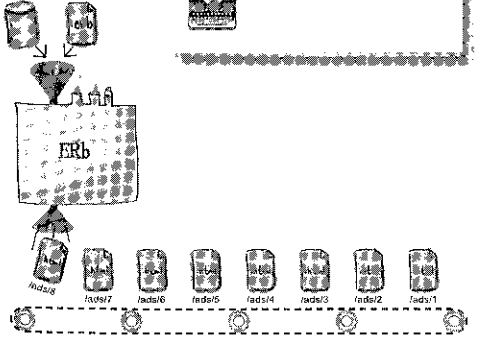


beyond scaffolding

2 Rails apps, made to order

So what's really going on with Rails? You've seen how **scaffolding** generates heaps of code and helps you write web applications wicked fast, but what if you want something a little different? In this chapter you'll see how to really ***seize control*** of your Rails development and take a look underneath the hood of the framework. You'll learn how Rails decides which **code** to run, how **data** is read from the database, and how **web pages** are generated. By the end, you'll be able to publish data the way ***you*** want.

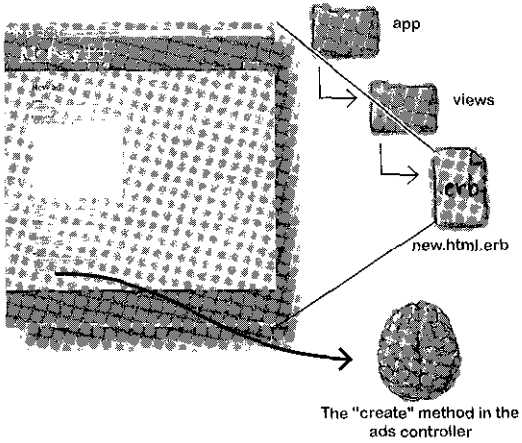


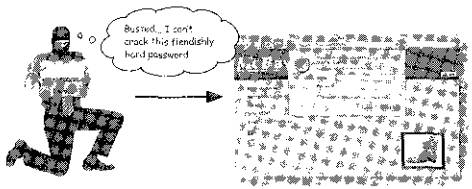


3 inserting, updating, and deleting

Everything changes

Change is a fact of life—especially for data. So far you've seen how to whip up a quick Rails application with scaffolding, and how to write your own code to publish data from a database. But what if you want users to be able to edit data *your* way? What *if scaffolding doesn't do what you want*? In this chapter, you'll learn how to **insert**, **update**, and **delete** data in exactly the way you want. And while you're doing that, you'll be taken deeper into how Rails *really* works and *maybe* even learn a little about security along the way.



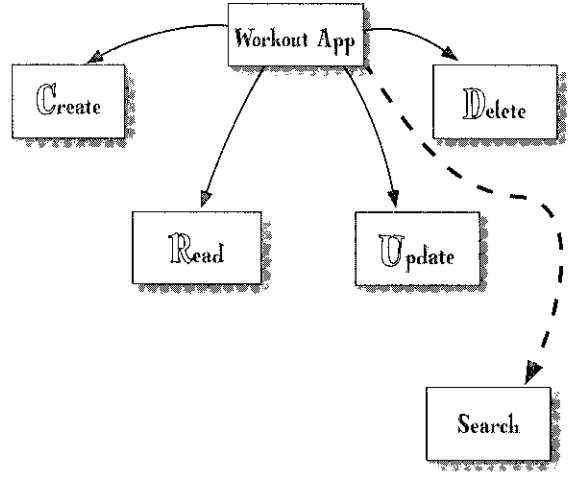


4 database finders

Truth or consequences?

Every decision you make has consequences.

In Rails, knowing how to make **good decisions** can save you both time and effort. In this chapter, we'll look at how **user requirements** affect the choices you make, right from the very **beginning** of your app. Should you use scaffolding and modify the generated code? Should you create things from scratch? Either way, when it comes time to customize your app further, you need to learn about **finders**: ***getting at your data*** in a way that makes sense to you and serves your **users' needs**.

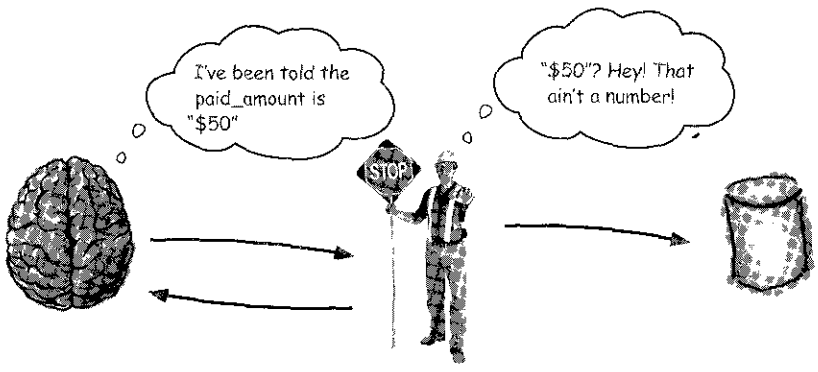


validating your data

5 Preventing mistakes

Everyone makes mistakes... but many of them are preventable!

Even with the very best of intentions, your users will still enter bad data into your web app, **leaving you to deal with the consequences**. But just imagine if there was some way of **preventing mistakes** from happening in the first place. That's where **validators** come in. Keep reading, and we'll show you how to add **clever Rails validation** to your web app so that you can **take control** of what data is allowed in—and what needs to be kept out.

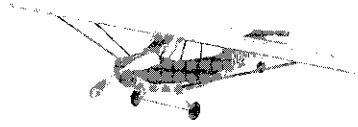


making connections

6 Bringing it all together

Some things are stronger together than apart.

So far you've had a taste of some of the **key Rails ingredients.** You've created entire web applications and taken what Rails generates and **customized** it for your needs. But out in the real world, **life can be more complex**. Read on.. it's time to build some **multi-functional web pages**. Not only that, it's time to deal with **difficult data relationships**, and take control of your data by writing your own **custom validators**.

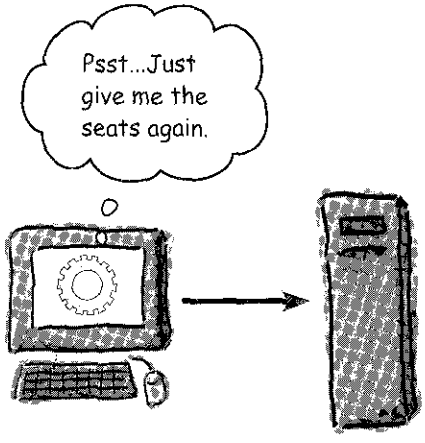


ajax

7 Avoiding the traffic

People want the best experiences out of life... and their apps.

No matter how good you are at Rails, there are times when traditional web apps just don't cut it. Sometimes users want something that's more **dynamic** and that responds to their every whim. Ajax allows you to build **fast, responsive web apps**, designed to give your users the **best experience the web has to offer**, and Rails comes complete with its own set of Ajax libraries just waiting for you to use. It's time to ***quickly and easily add* Ajax goodness** to your web app and please even *more users than before*.



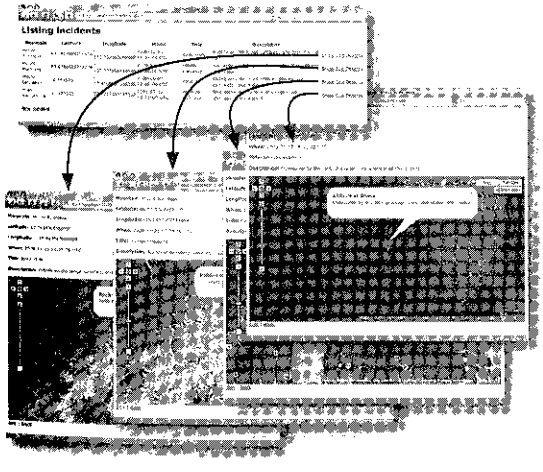
XML and multiple representations

8

It all looks different now...

You can't please everyone all of the time. Or can you?

So far we've looked at how you can use Rails to quickly and easily develop web apps that **perfectly fit one set of requirements**. But what do you do when **other requirements come along**? What should you do if some people want **basic web pages**, others want a **Google mashup**, and yet more want your app available as an **RSS feed**? In this chapter you'll create **multiple representations** of the same basic data, giving you the **maximum flexibility** with **minimum effort**.

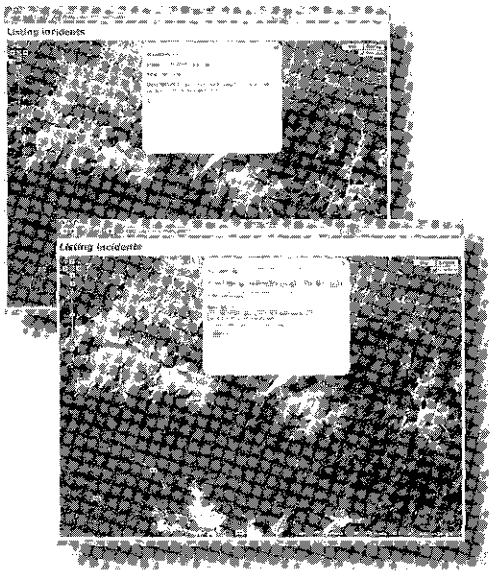


REST and Ajax

9 Taking things further

It's time to consolidate your mash-up skills.

So far you've seen how you can add **Google Maps** to your web apps to clearly show spatial data. But what if you want to **extend the functionality that's already there**? Keep reading, and we'll show you how you can add more **advanced Ajax goodness** to your **mash-ups**. And what's more, you'll learn a bit more about **REST** along the way.



real-world applications

10 Rails in the real world

You've learned a lot about Ruby on Rails.

But to apply your knowledge to **the real world**, there are a number of things you need to think about. How do you connect your application to **another database**? *How do you* ***test*** *Rails apps? How do you make the most out the Rails and the* **Ruby language**? And where do you find out the latest on **what's happening** with Rails? Keep reading, and we'll put you on the inside track that will take your development skills to the next level.