

Merged PRs

An Untapped Resource for
Practice & Exploration

@thomascountz

Merged PRs

Let's Talk about Debugging and then
Practice Debugging and also Build
Confidence by Diving into the Rails
Codebase which Hopefully Helps Us
Become a Better Rails Debuggers and
Developers

@thomascourtz

01

Theory

Defining Core
Competencies with
Insights from the Industry

02

Practice

Strengthening Core
Competencies with
Insights from the
Community

@thomascountz

This is the slide where I introduce myself...

01 - Theory

Defining Core Competencies with Insights
from the Industry

What is Debugging?

Debugging is the process of identifying and correcting unintended behavior in software systems.

Survey

**What percentage
of your time do you
think you spend on
debugging?**

**The average software
engineer spends between
35% - 50%
of their time on debugging
tasks.**

The cost of finding and fixing deficiencies is the largest single expense element in the software development lifecycle.. almost fifty cents out of every dollar will go to finding and fixing bugs.¹

1. Krasner, Herb. "The Cost of Poor Software Quality in the US: A 2022 Report," 2022.

*Everyone knows
that debugging is
twice as hard as
writing a program
in the first
place.*

*So if you're as
clever as you can
be when you write
it, how will you
ever debug it?¹*

1. Kernighan, Brian W., and P. J. Plauger. *The Elements of Programming Style*. 2d ed. New York: McGraw-Hill, 1978.

Debugging Competencies

- 01 Identifying problems
- 02 Reproducing issues
- 03 Applying a systematic approach
- 04 Comprehending code
- 05 Proficiently using debugging tools
- 06 Communicating effectively
- 07 Learning from experience

01

Identifying problems

Recognizing and isolating issues in code, based on error messages, unexpected behavior, or user reports

02

Reproducing issues

Consistently and accurately reproducing bugs in order to understand their root cause

03

Applying a **systematic approach**

Employing a structured, methodical process when debugging, such as the scientific method or other problem-solving techniques

04

Comprehending code

Reading and understanding code quickly, even if it's unfamiliar or written by someone else, and having a strong mental model of the code's architecture and design

05

Proficiently using debugging tools

Effectively using tools and techniques such as `puts`, logging, step debuggers, and profilers to help identify and resolve issues

06

Communicating effectively

Succinctly describing the problem, the steps taken to reproduce it, and the proposed solution to others

07

Learning from experience

Continuously improving debugging skills by learning from past experiences, keeping up to date with best practices, and updating mental models as new understanding is gained

Debugging Competencies

| Competency | Description |
|--|--|
| 01 ——— Identifying problems | Recognizing and isolating issues in code, based on error messages, unexpected behavior, or user reports |
| 02 ——— Reproducing issues | Consistently and accurately reproducing bugs in order to understand their root cause |
| 03 ——— Applying a systematic approach | Employing a structured, methodical process when debugging, such as the scientific method or other problem-solving techniques |

| Competency | Description |
|--|--|
| 04 ——— Comprehending code | Reading and understanding code quickly, even if it's unfamiliar or written by someone else, and having a strong mental model of the code's architecture and design |
| 05 ——— Proficiently using debugging tools | Effectively using tools and techniques such as <code>puts</code> , logging, step debuggers, and profilers to help identify and resolve issues |
| 06 ——— Communicating effectively | Succinctly describing the problem, the steps taken to reproduce it, and the proposed solution to others |
| 07 ——— Learning from experience | Continuously improving debugging skills by learning from past experiences, keeping up to date with best practices, and updating mental models as new understanding is gained |

**How do developers
become competent
at debugging?**

When we asked developers where their knowledge of debugging comes from, many said that "big chunks are self-taught" and "I picked up various bits and pieces on the Internet."¹

1. Beller, Moritz, Niels Spruit, and Andy Zaidman. "How Developers Debug." Preprint. PeerJ Preprints, January 23, 2017. <https://doi.org/10.7287/peerj.preprints.2743v1>.

~~Formal Education~~
Intentional Practice

In Brief

- Debugging takes time
- Debugging costs money
- Debugging is development
- Debugging isn't taught

In Brief

- Debugging takes time
- Debugging costs money
- Debugging is development
- Debugging isn't taught

In Brief

- Debugging takes time
- Debugging costs money
- Debugging is development
- Debugging isn't taught

In Brief

- ✓ Debugging takes time
- ✓ Debugging costs money
- ✓ Debugging is development
- ✓ Debugging isn't taught

I'm excited because the Ruby and Rails communities have the opportunity to lead the industry with our debugging expertise, just as we've done with our testing mastery.¹

1. Me (2023)

02 - Practice

Strengthening Core Competencies with
Insights from the Community

1. Setup your lab

Fork, clone, and/or use Github
Codespaces

See: Rails Guides

3. Debug the code

Don't be afraid to use a new
technique

2. Choose a candidate PR

Bug fix, clear description, few
changes. No peeking!

4. Evaluate your solution

Look at the code that was
merged; how did you do?

1. Set up your lab

1. Set up your lab

```
# Clone your rails/rails fork
$ git clone https://github.com/thomascountz/rails.git && cd rails
Cloning into 'rails'...
```

```
# Branch off from the merge commit's parent using `^`
$ git checkout -b practice-pr-46683 59fe981^ # PR#46683
Switched to a new branch 'practice-pr-46683'
```

```
# Run bundle install
$ bundle install
...
Using rails 7.1.0.alpha from source at `.`
...
```

```
# Create a new Rails app using the checked-out code
$ bundle exec rails new ../test_app --dev
...
Using activerecord 7.1.0.alpha from source at `/Users/thomas.countz/Code/rails`
...
```

All 1 Your codespaces Go to docs New codespace

Templates

By repository

Thomascountz/rails 1

Explore quick start templates [See all](#)

Blank
By github

Start with a blank canvas or import any packages you need.

Use this template

React
By github

A popular JavaScript library for building user interfaces based on UI components.

Use this template

Jupyter Notebook
By github

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data.

Use this template

Owned by Thomascountz

Thomascountz/rails ...

urban bassoon
practice

2-core • Active

1. Set up your lab

GitHub Codespaces

- Runs in the browser or locally with VSCode/Docker
- Run tests
- No additional dependencies

2. Find a candidate PR

2. Find a candidate PR

| Criteria | Description |
|----------------|--|
| Bug Fix | Not all PRs are bug fixes; many of them are new features and refactors |
| Linked issue ☺ | A linked issue is a good sign that the PR is related to a bug and let's really imagine picking up a newly reported bug |
| Labels | The <code>rails/rails</code> repo uses labels for specific Rails components like <code>actionable</code> , <code>activejob</code> , <code>activerecord</code> , etc. |

| Criteria | Description |
|--------------------|---|
| Few changes | PRs under +/- 100 lines are usually better for practice as they're often lower in complexity |
| Little discussion | PRs with a lot of discussion usually involve additional context that may be hard to glean retroactively |
| Merged; not closed | <code>is:closed</code> will returned both merged and closed PRs; use <code>is:merged</code> instead. |

2. Find a candidate PR

#47824

rails / rails Public

Watch 2.4k Fork 21.1k Star 52.7k

<> Code Issues 379 Pull requests 443 Actions Projects 2 Security 3 Insights

Fix autoincrement on primary key for mysql #47824

Merged eileencodes merged 1 commit into main from fix-auto-increment-option-on-primary-key 3 weeks ago

Conversation 1 Commits 1 Checks 15 Files changed 2 +12 -1

eileencodes commented 3 weeks ago

MySQL adapters take an `auto_increment` option but when using a primary key type that option is silently ignored and doesn't change behavior. This fixes the issue by only applying `options[:auto_increment] = true` if it wasn't set to false already.

I didn't make changes to the other adapters because they don't accept `auto_increment` as an option. If we want this for postgres we'll need to implement the option first.

Reviewers: rafaelfranca ✓

Assignees: No one assigned

Labels: activerecord

Fix autoincrement on primary key Verified 01c4605

2. Find a candidate PR

#46683

rails / rails Public

Watch 2.4k Fork 21.1k Star 52.7k

<> Code Issues 379 Pull requests 443 Actions Projects 2 Security 3 Insights

Fix `add_foreign_key` with `if_not_exists` referencing the same table but via different columns #46683

Merged eileencodes merged 1 commit into rails:main from fatkodima:add_foreign_key-if_not_exists on Dec 9, 2022

Conversation 0 Commits 1 Checks 3 Files changed 2 +12 -1

fatkodima commented on Dec 9, 2022 Contributor

Fixes #46680.

Fix `add_foreign_key` with `if_not_exists` referencing the same table... cf3c04f

rails-bot bot added the activerecord label on Dec 9, 2022

Reviewers: No reviews

Assignees: No one assigned

Labels: activerecord

2. Find a candidate PR

#46192

rails / rails Public

Watch 2.4k Fork 21.1k Star 52.7k

<> Code Issues 379 Pull requests 444 Actions Projects 2 Security 3 Insights

Add support for unique constraints (PostgreSQL-only). #46192

Merged yahonda merged 1 commit into rails:main from alpaca-tc:support_unique_constraints on Mar 12

Conversation 20 Commits 1 Checks 4 Files changed 15 +534 -3

alpaca-tc commented on Oct 4, 2022 · edited Contributor

Motivation / Background

Add support for unique constraints (PostgreSQL-only).

```
add_unique_key :sections, [:position], deferrable: :deferred, name: "unique_section_position"
remove_unique_key :sections, name: "unique_section_position"
```

See PostgreSQL's [Unique Constraints](#) documentation for more on unique constraints.

By default, unique constraints in PostgreSQL are checked after each statement. This works for most use cases, but becomes a major limitation when replacing records with unique column by using multiple statements.

Reviewers: yahonda ✓

Assignees: No one assigned

Labels: activerecord PostgreSQL

Projects: None yet

3. Debug the code

STDOUT

`puts`

Debugging

Use `puts` to output variables and other information to `STDOUT` during runtime.

STDOUT

Configuring the Rails Logger

In debugging, it's often easiest to
configure the logger to print to `STDOUT`

```
config.logger = Logger.new(STDOUT)
logger.info "--- HERE! ---"
```


Technique

Scientific Debugging

Observe a failure, invent a hypothesis, make predictions, test the hypothesis by experiments and observations, and repeat until the defect is found and removed.

Technique

Desk Checking

Read the code line-by-line!
It's amazing how easy it is to
forget to read.

Assertion

Tests

Each component, `activerecord`, `actionmailer`, `actionview`, etc, has their own `bin/test` executable that runs the test for that particular component.

Assertion

Bug report and benchmark templates

Located in `rails/guides/bug_report_templates/`, these templates are self-contained Ruby scripts that set up the minimal environment required to test and benchmark specific components.

Debugger

**ruby/debug (rdbg),
Pry, Byebug**

These are Ruby's debuggers. They let you set breakpoints, step through the code line-by-line, evaluate code, check variables, find where a given method is defined, produce a stack trace without an error, and more

Debugger

rdbg VSCode extension

The extension for Visual Studio Code integrates with rdbg to provide a powerful debugger that works within the editor. This makes it easy to set breakpoints, step through code, and inspect variables directly within the editor.

See: @searls Debugging Ruby on Rails with Visual Studio Code

Debugger

Web Console

The Web Console is a Rails-specific tool that provides an interactive console within the context of a specific page in your Rails application.

Code navigation

```
bundle open
```

The `bundle open` command opens the source code of a gem in your default editor.

Code navigation

```
git bisect
```

We would reach for `git bisect` when we want to identify the exact commit that introduced a bug in our Rails application's commit history.

Code navigation

```
#source_location,  
#caller
```

The `Method#source_location` method will return the file path and line number of the method on which it was called, `Kernel#caller` returns the current execution stack.

Parser

`parser`

& Ripper

We can manipulate the Ruby AST
(abstract syntax tree) to modify or
refactor our code
programmatically.*

Audience Participation

"#{tool}"

**What tool or technique have
you found helpful?**

4. Evaluate your solution

4. Evaluate your Solution

| Criteria | Description |
|--|--|
| 01 ——— Identifying problems | Recognizing and isolating issues in code, based on error messages, unexpected behavior, or user reports |
| 02 ——— Reproducing issues | Consistently and accurately reproducing bugs in order to understand their root cause |
| 03 ——— Applying a systematic approach | Employing a structured, methodical process when debugging, such as the scientific method or other problem-solving techniques |

| Competency | Description |
|--|--|
| 04 ——— Comprehending code | Reading and understanding code quickly, even if it's unfamiliar or written by someone else, and having a strong mental model of the code's architecture and design |
| 05 ——— Proficiently using debugging tools | Effectively using tools and techniques such as <code>puts</code> , logging, step debuggers, and profilers to help identify and resolve issues |
| 06 ——— Communicating effectively | Succinctly describing the problem, the steps taken to reproduce it, and the proposed solution to others |
| 07 ——— Learning from experience | Continuously improving debugging skills by learning from past experiences, keeping up to date with best practices, and updating mental models as new understanding is gained |

Elephants

What is Debugging?

Debugging is the process of identifying and correcting unintended behavior in software systems.

**The average software
engineer spends between
35% - 50%
of their time on debugging
tasks.**

Debugging Competencies

- 01 Identifying problems
- 02 Reproducing issues
- 03 Applying a systematic approach
- 04 Comprehending code
- 05 Proficiently using debugging tools
- 06 Communicating effectively
- 07 Learning from experience

1. Setup your lab

Fork, clone, and/or use Github
Codespaces

See: Rails Guides

3. Debug the code

Don't be afraid to use a new
technique

2. Choose a candidate PR

Bug fix, clear description, few
changes. No peeking!

4. Evaluate your solution

Look at the patch; how did you
do?

01

Theory

Defining Core
Competencies with
Insights from the Industry

02

Practice

Strengthening Core
Competencies with
Insights from the
Community

In Closing

Thanks!

**Questions?
Comments?
Concerns?**

**Thoughts?
Surprises?
Delights?**