

[КАК СТАТЬ АВТОРОМ](#)[Финальный питч-дек Битвы](#) [Рейтинг IT-работодателей как...](#)

vltsu

15 июн 2011 в 19:46

# RVM — подробно



4 мин



95K

Ruby\*



RVM — Ruby Version Manager

Программа для управления версиями Ruby.

Как быть если один проект использует Ruby 1.8.7, а другой 1.9.2?

А что если при этом у вас 2 проекта под версией 1.9.2, но с

разными наборами гемов? Хотелось бы вам иметь отдельные наборы

gem's для каждого проекта?

Основные задачи:

1. Физическое разделение версий ruby и наборов гемсетов
2. Возможность иметь несколько версий ruby и переключаться между ними
3. Возможность для каждой версии ruby иметь несколько gemsets – наборов gem's и переключаться между ними

Под катом будут описаны:

1. Установка RVM
2. Работа с разными версиями Ruby
3. Работа с gemsets
4. Задание окружения под отдельный проект с помощью .rvmrc
5. Команды RVM которые могут оказаться полезными
6. Шпаргалка по основным командам RVM

## 1. Установка RVM

1. Для установки вам понадобятся curl и git (apt-get install git curl)
- 2.

```
#Скачивание скрипта установки с сервера и запуск (от имени вашего пользователя)
% bash < <(curl -s https://rvm.beginrescueend.com/install/rvm)
```

3. После установки нужно добавить информацию об установленном приложении `rvm` в `bash`

```
#Дописывает в конфигурационный файл bash строку, которая проверяет
наличие директории с rvm и если
#всё ок, запускает программу
% echo '[[ -s "$HOME/.rvm/scripts/rvm" ]] && . "$HOME/.rvm/scripts/rvm" # Load RVM
function'>> ~/.bash_profile
```

В файле `$HOME/.bashrc` заменяем строку `[ -z "$PS1" ] && return` на `if [[ -n "$PS1" ]]; then`. Т.к. мы используем `if`, то в конце файла поставьте `fi`, это позволяет запускать программы в неинтерактивном режиме, а `rvm` как раз является примером такой программы

4. Проверяем как прошла установка

```
% rvm -v
rvm 1.6.20 by Wayne E. Seguin (wayneeseguin@gmail.com)
[https://rvm.beginrescueend.com/]
# если вы видите версию rvm, значит всё ОК
```

## 2. Работа с разными версиями Ruby

Чтобы просмотреть все доступные для установки версии Ruby

```
% rvm list known
# MRI Rubies
[ruby-]1.8.6[-p420]
[ruby-]1.8.6-head
[ruby-]1.8.7[-p334]
[ruby-]1.8.7-head
[ruby-]1.9.1-p378
[ruby-]1.9.1[-p431]
[ruby-]1.9.1-head
[ruby-]1.9.2[-p180]
[ruby-]1.9.2-head
ruby-head
# GoRuby
goruby
# JRuby
jruby-1.2.0
jruby-1.3.1
jruby-1.4.0
jruby-1.6.1
```

```
jruby[-1.6.2]
jruby-head
```

Предположим вы работаете с двумя версиями 1.8.7 для ROR2 и 1.9.2 для ROR3

```
# Устанавливаем 2 версии Ruby
% rvm install 1.8.7
% rvm install 1.9.2
```

Чтобы увидеть все установленные версии Ruby

```
% rvm list
rvm rubies
ruby-1.8.7-p334 [ i386 ]
ruby-1.9.2-p180 [ i386 ]
```

Переключиться на ruby 1.8.7

```
% rvm use ruby-1.8.7
Using /home/user/.rvm/gems/ruby-1.8.7-p334
```

Использовать версию ruby 1.9.2 по умолчанию

```
% rvm use ruby 1.9.2 --default
Using /home/user/.rvm/gems/ruby-1.9.2-p180
% rvm list
rvm rubies
=> ruby-1.9.2-p180 [ i386 ] # "=>" показывают версию Ruby
установленную по умолчанию
    ruby-1.8.7-p334 [ i386 ]
```

### 3. Работа с gemsets

Например вы используете Ruby On Rails версии 2 и 3 с Ruby 1.8.7 и для каждого из них у вас свой набор gem's.

Создадим два разных набора gemset'ов:

```
% rvm use 1.8.7@rails2 --create
Using /home/user/.rvm/gems/ruby-1.8.7-p334 with gemset rails2
% rvm use 1.8.7@rails3 --create
Using /home/user/.rvm/gems/ruby-1.8.7-p334 with gemset rails3
% rvm gemset list
gemsets for ruby-1.8.7-p334 (found in /home/slip/.rvm/gems/ruby-
1.8.7-p334)
    global # gemset по умолчанию
```

```
rails2
rails3
% rvm use 1.8.7@rails3 --default # Использовать gemset rails3 по умолчанию
% rvm gemset list
gemsets for ruby 1.8.7-p334 (found in /home/user/.rvm/gems/ruby-1.8.7-p334)
  global
  rails2
=> rails3 # rails3 стоит по умолчанию
```

Gemset'ы можно удалять, очищать, экспортировать и импортировать гемы из одного в gemset'а в другой.

RVM предоставляет следующие команды для работы с gemsets:

**create** – создание нового gemset  
**export** – экспорт списка гемов в файл default.gems  
**import** – установка в текущий gemset списка гемов из файла default.gems  
**delete** – удалить gemset  
**empty** – очистить gemset

#### 4. Задание окружения под отдельный проект с помощью .rvmrc

Как быть если у вас несколько проектов, каждый из которых используют разную версию gemset? Можно конечно переключиться между gemset'ами вручную с помощью `rvm use {rubyversion}@{gemsetname}`, но и тут RVM приходит нам на помощь, делая эту часть работы за нас.

Создаем файл .rvmrc в корневой директории проекта. Например проект используют ruby версии 1.8.7 с gemset projectname.

```
#Содержимое файла .rvmrc
rvm use 1.8.7@projectname
```

Теперь когда вы заходите в директорию, `cd /home/user/www/projectname` – RVM исполняет команду из файла .rvmrc и вы видите на экране подобное сообщение

```
Using /home/user/.rvm/gems/ruby-1.8.7-p334 with gemset projectname
```

Таким образом вам больше не приходится думать какой gemset использует конкретный проект и устанавливать его вручную.

#### 5. Команды RVM которые могут оказаться полезными

1. Completion – позволяет использовать tab при работе с rvm

Чтобы включить, добавьте строку `[[ -r $rvm_path/scripts/completion ]] &&`.

`$rvm_path/scripts/completion` в файл `.bashrc` или `.bash_profile`, после строки с подключением `rvm`. Подробнее можно почитать здесь

[rvm.beginrescueend.com/workflow/completion](https://rvm.beginrescueend.com/workflow/completion)

2. **rvmreset** – перезагрузка RVM

3. **rvm uninstall** – удалить одну или несколько версию Ruby, оставив исходники

4. **rvm implode** – полностью удалить RVM (удаляет ВСЁ)

## 6. Шпаргалка по основным командам RVM

**rvm list known** – получить список всех версий ruby доступных для установки

**rvm install 1.9.1** – установить ruby версии 1.9.1

**rvm remove 1.9.2** – удалить ruby версии 1.9.2

**rvm use 1.9.2** – переключиться на ruby версии 1.9.2

**rvm use 1.9.2@rails3 --default** – установить версию ruby 1.9.2 с gemset rails3 по умолчанию

**rvm use system** – использовать системную версию ruby

**rvm list** – список установленных версий ruby

**rvm gemset list** – список gemset'ов в выбранной версии ruby

**rvm use 1.9.2@rails3 --create** создать gemset rails3 для ruby версии 1.9.2

**rvm gemset export** – экспортировать гемсет в файл `default.gems`

**rvm gemset import default.gems** – установить gem's из списка в файле `defaults.gem` в текущий gemset

Теги: `rvm`, `ruby`, `ruby on rails`, `ruby version manager`

Хабы: [Ruby](#)

◆ +43

📖 271



💬 44



33

0

Карма

Рейтинг

Цуканов Владимир @vltsu

Пользователь

Подписаться



Реклама



## Комментарии 44

**iTube**

15 июн 2011 в 20:31

Спасибо, за хороший cheatsheet



0

Ответить

**simonoff**

15 июн 2011 в 20:31

по поводу джемсетов – желательно использовать bundler.  
это безопаснее и удобнее. мы сетапим новый сервак, ставим gvm, нужную версию руби и все. а не ставим все нужные нам джемсеты.



+2

Ответить

**azakharov**

16 июн 2011 в 16:42

Согласен. К раз про это писал недавно [habrahabr.ru/blogs/ruby/120272/](http://habrahabr.ru/blogs/ruby/120272/)



0

Ответить

**ognevsky**

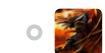
17 июн 2011 в 01:27

желательно, простите, для кого? и чем безопаснее, и чем удобнее?



0

Ответить

**simonoff**

17 июн 2011 в 16:13

для тех кто пишет приложения с использованием Ruby On Rails.  
безопаснее тем что у каждого приложения все ему нужные гемы находятся в его папке и подставить модифицированный системный не получится.  
удобнее тем что админу всего этого не надо для каждого проекта делать свой жемсет и не ставить все нужные жемы в этот жемсет.



0

Ответить

**vltsu**

17 июн 2011 в 16:41

При использовании gvm bundler ставить гемы в текущий gemset



0

Ответить

**simonoff**

17 июн 2011 в 16:42

а зачем?



 0 Ответить **vltsu**  
17 июн 2011 в 16:54

При запуске `bundle install` под гемсетом `1.9.2@projectname`, геммы устанавливаются в директорию `/home/user/.rvm/gems/ruby-1.9.2-p180@projectname/gems`

Так что никакой директории с геммами под отдельный проект нет. Есть только отдельный `gemset`

 0 Ответить **simonoff**  
17 июн 2011 в 16:54

вы капистрано пользуетесь или нет?

 0 Ответить **vltsu**  
17 июн 2011 в 16:56

нет, я про случай без капистрано

 0 Ответить **simonoff**  
17 июн 2011 в 17:19

а я вам про капистрано. да и вообще `bundler` у себя можно настроить так что будет ставить жемы в папку с проектом, что очень удобно. и тем более вы перестанете использовать `rake`, `rails`, etc без `bundle exec`.

 +1 Ответить **vltsu**  
17 июн 2011 в 17:23

спасибо, не знал. Поэкспериментирую с этим

 0 Ответить **simonoff**  
17 июн 2011 в 17:25

`bundle install --path vendor/bundle`

Более подробно здесь – [gembundler.com/bundle\\_install.html](http://gembundler.com/bundle_install.html)

 +1 Ответить **ognevsky**  
17 июн 2011 в 19:49

Ну так в гемсетах и нет «модифицированных системных» никаких, просто гемы лежат не в текущей папке, а в системе (в отдельной папке своего гемсета). Про модификацию гема руками я даже не говорю, этим наверняка уже никто не страдает.

по поводу удобства: у меня в каждом проекте лежит `.gvmrc`, который автоматически создает гемсет. В глобальном гемсете у меня лежит только бандлер и одной простой командой все эти гемы ставятся в новый гемсет: изоляция идеальная.

Нет, я совсем не говорю, что бандлером пользоваться нельзя или не нужно, просто не нужно так категорично заявлять, что гемсеты использовать ну прямо никак нельзя. Разницы, по большому счету, нет совершенно никакой, все делают так, как конкретно им удобнее.

0 Ответить

 **simonoff**  
17 июн 2011 в 20:18

то есть вы бандлер у себя вообще не используете? или как?

0 Ответить

 **ognevsky**  
17 июн 2011 в 20:22

почему же не использую? очень использую!

есть 3 варианта:

- ваш подход (гемы для проекта хранятся внутри проекта, отчего сам проект тяжелеет)
- мой подход (в `Gemfile` описаны необходимые гемы, при настройке проекта я выполняю `bundle install` и гемы скачиваются в папку чистого и свежего гемсета)
- устаревший вариант (без `gvm`: есть некий глобальный гемсет, куда сваливаются все гемы для всех проектов).

я не вижу разницы между первым и вторым вариантом, за исключением того, что я не храню гемы в репозитории с проектом.

+1 Ответить

 **simonoff**  
17 июн 2011 в 20:26

ну по поводу третьего варианта я скажу то что бандлер туда как раз очень хорошо и вписывается.

`gvm` просто средство управления версиями руби в моём варианте.

а по поводу утяжеления проекта, скажу так что при деплое очень даже ускоряется скорость `bundle install`, за счет того что нужные жемы у вас уже в папке. Для меня это важно, потому как у меня ноды поднимаются автоматом если нагрузка выросла, и как бы чем быстрее она поднимется, тем лучше.





0

Ответить

**ognevsky**

17 июн 2011 в 20:30



Но зато деплой/пуш занимает больше времени, это тоже не очень круто. Нужно иметь для серверов локальную/распределенную хранилку гемов.



0

Ответить

**simonoff**

17 июн 2011 в 20:36



это почему он будет занимать больше времени?



0

Ответить

**ognevsky**

17 июн 2011 в 20:37



Ну гемы тяжелые, их много. Но это да, мелочь.



0

Ответить

**simonoff**

17 июн 2011 в 20:40



хм... помоему больше времени при bundle install занимают запросы на индекс с зеркал rubygems.org.



0

Ответить

**ognevsky**

17 июн 2011 в 20:55



Ну тогда делать локальный сторадж гемов, откуда все ваши проекты будут забирать геммы: не будет этой задержки. akzhan про это больше рассказать сможет



0

Ответить

**simonoff**

17 июн 2011 в 21:01



лишние пару метров с жемами в репе роли не играют.



0

Ответить

**Eklykti**

9 авг 2011 в 08:22



> за исключением того, что я не храню геммы в репозитории с проектом.

```
echo 'vendor/bundle' >> .gitignore
```

 0 Ответить

 **Flcn**  
15 июн 2011 в 20:41

Вот всё примерно тоже самое, только на видео с RailsClub

[univertv.ru/video/informatika/programmirovanie/konferenciya\\_railsclubmoscow/rvm\\_i\\_nashi\\_servery/](http://univertv.ru/video/informatika/programmirovanie/konferenciya_railsclubmoscow/rvm_i_nashi_servery/)

 +2 Ответить



 **SmartBye**  
15 июн 2011 в 20:42

В `.rvmrc` использую команду с ключом `--create`. Даже первым делом создаю этот файл `echo «rvm 1.9.2@project_name – create >> project_name/.rvmrc` И после `cd` в `project_name` гемсет сам создается. А уж потом делаю все остальное.

 +1 Ответить



 **reloky**  
15 июн 2011 в 21:26

Спасибо за перевод страниц сайтика `rvm` на русский.

Хотелось бы освещения так же вопросов, которые не обозначены на сайте, например –

У меня два rails проекта – один на руби 1.8, второй на руби 1.9. Как запустить оба сайта с помощью `rvm`?

ну и тому подобные...

 0 Ответить



 **vltsu**  
15 июн 2011 в 21:41

Устанавливаете две версии руби

```
rvm install ruby-1.8.7
```

```
rvm install ruby-1.9.2
```

Работаете с одним проектом из под `rvm use 1.8.7`, с другим `rvm use 1.9.2`

Чтобы не переключаться вручную можно создать файл `.rvmrc` в корневой директории каждого проекта.

 0 Ответить



 **reloky**  
15 июн 2011 в 21:47

работать то можно, вот только скажем пассажир так не запустишь.



0

Ответить

**vltsu**

15 июн 2011 в 22:00



у меня есть в планах статья про разворачивание гыбу проектов на серверах, скорее всего это будет там



0

Ответить

**reloky**

15 июн 2011 в 22:43



было бы чудесно.



0

Ответить

**simonoff**

15 июн 2011 в 22:45



а пассажир и не нужен, используйте unicorn.



0

Ответить

**МраKus**

16 июн 2011 в 10:31



пассажир как-то ведь попроще, тем более с 1.8.7 EE версия пошустрее будет



0

Ответить

**simonoff**

16 июн 2011 в 16:00



ну так спрашивали по поводу разных версий руби и пассажира. Вариант ниже с запуском отдельных пассажиров с RVM не добавляет каких-то упрощений по сравнению с unicorn.



0

Ответить

**vltsu**

15 июн 2011 в 21:43



напишите ещё, что бы вы хотели узнать, сделаю седьмой пункт «RVM в жизни»



0

Ответить

**МраKus**

16 июн 2011 в 10:32



как в таком случае настраивать проекты если разные версии Ruby 1.8 и 1.9 под Nginx + Passenger, чтобы последний запускал как надо?



0

Ответить

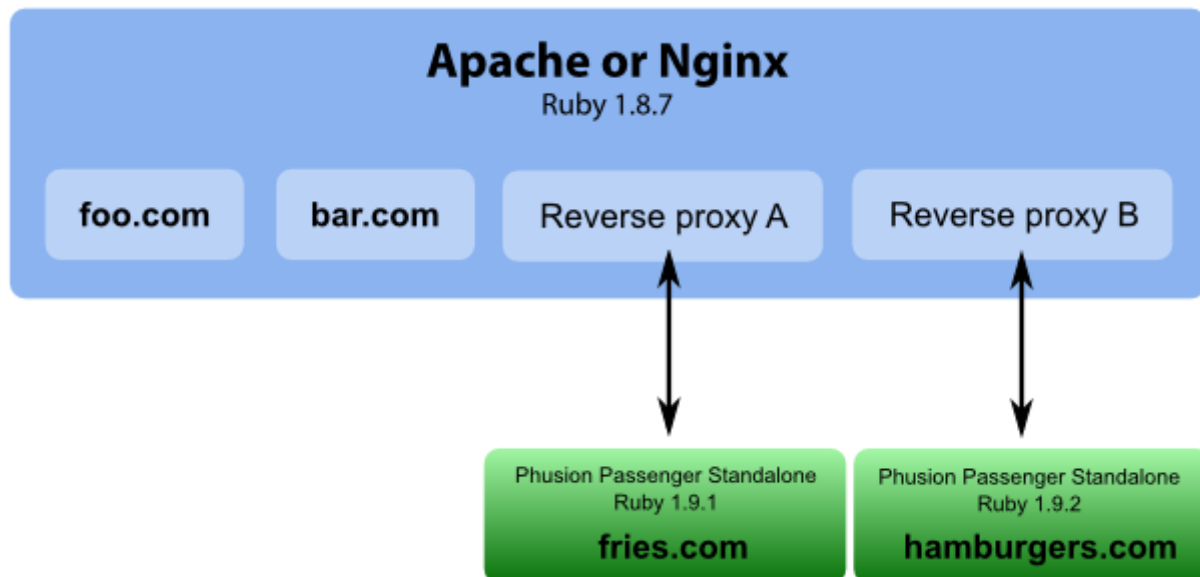


**levkin**

16 июн 2011 в 14:25

[blog.phusion.nl/2010/09/21/phusion-passenger-running-multiple-ruby-versions/](http://blog.phusion.nl/2010/09/21/phusion-passenger-running-multiple-ruby-versions/)

Вкратце, они предлагают на системном руби ставить одни проекты (1.8.7), а потом с помощью RVM в стандалон-режиме запускать другие на той версии рубей, какая нужна для каждого проекта. Ну и на эти пассенджеры, отличные от системного заводить юзеров через РеверсПрокси.



0

Ответить

**МраK999**

16 июн 2011 в 16:12

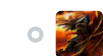
Вот как, интересная идея.

А если через Passenger запускать системный Ruby 1.8, а через Unicorn + RVM пускать 1.9 как вариант обойтись без обратного прокси?



0

Ответить

**simonoff**

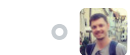
16 июн 2011 в 16:32

Эм... чето я немного не понимаю что вы хотите сделать? выставлять наружу Passenger или Unicorn? без apache или nginx перед ними? насчет пассенжера незнаю хорошая ли это идея, и вообще как это сделать, а вот насчет Unicorn'a скажу что это плохая идея. Он не рассчитан на медленных клиентов, возьмите Rainbows!



0

Ответить

**levkin**

16 июн 2011 в 16:34

Passenger в чистом виде не выставишь. В случае с апачем он в виде либы, в стандалоне юзает nginx как ядро.

 0 Ответить  **simonoff**  
16 июн 2011 в 16:37

Э... а можете тыкнуть в доку, где написано что Standalone версия использует nginx как ядро?

 0 Ответить  **levkin**  
16 июн 2011 в 16:46

[www.modrails.com/documentation/Users%20guide%20Standalone.html](http://www.modrails.com/documentation/Users%20guide%20Standalone.html)

«It is powered by an Nginx core.»

 0 Ответить  **simonoff**  
16 июн 2011 в 16:48

Хм... спасибо. Но мне такой вариант уже не нравится. мухи отдельно, котлеты отдельно.

 0 Ответить  **levkin**  
16 июн 2011 в 16:32

Тоже вариант. И не обязательно Unicorn. В этом случае хоть Mongrel, хоть Thin.

 0 Ответить

Вы можете оставлять комментарии только к свежим публикациям

## Публикации

ЛУЧШИЕ ЗА СУТКИ    ПОХОЖИЕ

 **mr-pickles**  
23 часа назад

16-, 8- и 4-битные форматы чисел с плавающей запятой

 Средний

15 мин



8.8K

 +85 107 47

ru\_vds

20 часов назад

## Искусство создания понятных графиков

 Средний

7 мин



6K

 Тutorial Перевод +58 100 5

SLY\_6

15 часов назад

## Известная, но очень странная кошачья повадка: кошки, приносящие игрушку — это эволюционная загадка

 5 мин

15K

 Перевод +40 29 77

alizer

вчера в 16:00

## Автономия разработчиков. Как устроены компании нового типа

 Простой

7 мин



5.7K

 Мнение +31 27 30

Vutshi

2 часа назад

## Доступная система видеонаблюдения с охраной на дачу за 60 минут

 24 мин

2.4K

 +17 34 13

badcasedaily1

17 часов назад

## Garbage Collection и JVM

 Простой  17 мин  3.3K

[Обзор](#)

 +17

 89

 5



**SolaireOfAstora**

7 часов назад

## «Импортозамещение» в видеонаблюдении. Часть третья: НИЦ “Технологии”

 Простой  9 мин  3.7K

[Обзор](#)

 +16

 6

 10



**Petr0v1**

19 часов назад

## Марс всё ближе: несмотря на проблемы, запуск Starship можно считать успешным

 4 мин  5.7K

 +16

 12

 16



**Doctor\_IT**

20 часов назад

## Сможет ли високосная минута решить проблему синхронизации часов?

 6 мин  3.7K

 +15

 18

 17



**SLY\_6**

21 час назад

## Дайджест наукоп-новостей за неделю, о которых мы ничего не писали

 8 мин  2.1K

[Дайджест](#)

## Тур по городам России в поисках backend-разработчиков

Спецпроект

Показать еще

### МИНУТОЧКУ ВНИМАНИЯ



Турбо

Почему ИИ не воспринимают как благо



Спецпроект

Тур по городам России в поисках backend-разработчиков

### ЗАКАЗЫ

Доработка проекта на Ruby on Rails

200000 руб./за проект • 2 отклика • 11 просмотров

Миграция с AdminJS на Minimal Dashboard

60000 руб./за проект • 2 отклика • 53 просмотра

Сбор рабочих контактов: Founders of B2B Software Startups

3000 руб./за проект • 2 отклика • 26 просмотров

Доработка IOT кода node.JS (express, mqtt)

3500 руб./за проект • 40 просмотров

Сбор рабочих контактов email/LinkedIn/Instagram

3000 руб./за проект • 7 откликов • 64 просмотра

Больше заказов на Хабр Фрилансе



ЧИТАЮТ СЕЙЧАС

Известная, но очень странная кошачья повадка: кошки, приносящие игрушку — это эволюционная загадка

15K 77

Развенчание мифа о собственной продуктивности программистов

14K 45

Li-Fi — в 100 раз быстрее, чем Wi-Fi

37K 79

Технологическая сингулярность или угасание цивилизации? Наиболее вероятные сценарии развития человечества

25K 121

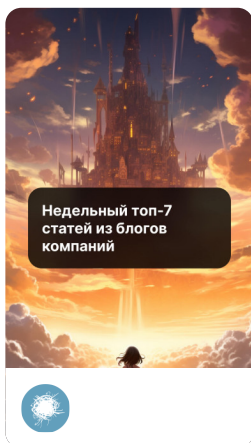
Доступная система видеонаблюдения с охраной на дачу за 60 минут

2.4K 14

Тур по городам России в поисках backend-разработчиков

Спецпроект

## ИСТОРИИ



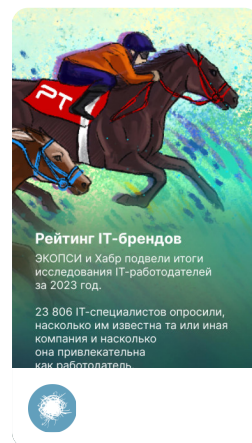
Топ годноты из блогов компаний



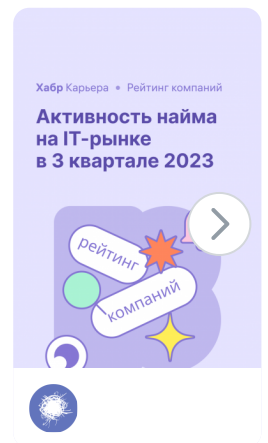
Идём на «Импульс Т1» по дороге из жёлтого кирпича



Как стать супергероем



Рейтинг IT-брендов работодателей 2023



Активность найма в 3 квартале 2023

## РАБОТА

---

Программист Ruby

12 вакансий

Ruby on Rails

19 вакансий

---

Все вакансии

Реклама



Ваш аккаунт

Профиль  
Трекер  
Диалоги  
Настройки  
ППА

Разделы

Статьи  
Новости  
Хабы  
Компании  
Авторы  
Песочница

Информация

Устройство сайта  
Для авторов  
Для компаний  
Документы  
Соглашение  
Конфиденциальность

Услуги

Корпоративный блог  
Медийная реклама  
Нативные проекты  
Образовательные  
программы  
Стартапам  
Спецпроекты



Настройка языка

Техническая поддержка

© 2006–2023, Habr