

Rails开发关键技术

创投圈内部培训资料

1. 常用shell命令

1. 创建scaffold、controller和model

1) 创建scaffold:

```
rails generate scaffold model_name column1:column1_type column2:column2_type ...
```

效果：创建一个名称为model_name的model，以及相关的controller、view、assets、DB migration等，是创建一个标准CRUD模块最快捷的方式。

2) 创建model:

```
rails generate model model_name column1:column1_type column2:column2_type ...
```

效果：创建一个和数据表对应的model，常用于需要存储数据而又不需要页面显示的模型。例如：各种关联表。

3) 创建controller:

```
rails generate controller controller_name
```

效果：创建一个controller和它对应的views页面。

2. 创建数据迁移任务migration

```
rails generate migration migration_name_to_table_name column1:type column2:type
```

...

效果：创建一个用于数据迁移的migration任务。惯例：后面可以带几个column和类型的参数，可以对应于migration_name_to_table_name中的table_name，自动生成添加column的代码：add_column :table_name, :column1, :type

2. HTTP请求的关键配置

1. Routes 路由配置

1) 尽量遵守Restful原则的路径

例如：resources :users

这句代码会自动为users的get、post、put、delete等请求添加路由配置，例如：

```
match "/users" => "users#index"
match "/users/:id" => "users#show", via: :get
match "/users" => "users#index", via: :post
match "/users/:id" => "users#index", via: :put
match "/users/:id" => "users#index", via: :delete
```

在resources中可以嵌套规则：

```
resources :users do
  member do
    get :show_info
  end
end
```

相当于添加了一个可访问的路径：/users/:id/show_info

```
resources :users do
  collection do
    get :descriptions
  end
end
```

相当于添加了一个可访问的路径：/users/descriptions

post请求修改get为post即可，其他http请求类型同理。

2) 自定义路径

例如：

```
match "/users/:id/special_route" => "users#special_route", via: :get
```

```
match "/note" => "home#note", via: :get, as: :note
```

常用语非resourceful的特殊路径配置中。

2. Controller action的respond_to配置

在一个controller的action方法末尾，一般带有一个respond_to方法，用于指定响应何种类型的请求，以及请求render什么模板、什么数据等。

例如，在 users_controller#show中：

```
respond_to do |format|  
  format.html  
  format.json { render json: @user.activities }  
end
```

根据Rails的惯例，一个action执行结束后，会自动render一个同名的view page，因此users_controller#show将render到users/show.html.erb下。

如果访问的请求类型是json，则会触发：

```
format.json { render json: @user.activities }
```

然后请求将向客户端返回一段json格式的字符串（一般用于API设计中）

3. Rails 数据库接口

1. Model的类方法

- Model.all
- Model.update_all(params_hash)
- Model.destroy_all
- Model.find(id)
- Model.new(params_hash)

2. Model的实例方法

- @obj.update_attribute(:column, value)
- @obj.update_attributes(params_hash)
- @obj.destroy

- `@obj.save`

3.