金数据

基于 Rails 的 SaaS 应用实践与总结

冯智超 @金数据

金数据

冯智超 @金数据 @西安

@flankerfc @fengzhichao

喜欢 #编程 #龙珠

20 years+ 编程经验

Basic @1997

Ruby @2006 #RPG_Maker

Rails @2011

金数据
人人可用的在线表单工具

帮助用户收集和管理日常工作中的数据，提升工作效率。任何行业和岗位的人员，无需特殊技能，都可以方便的创建出符合业务需求的表单。

朋友 Anna 开花店，想在微信里卖花。

```
class FlowerOrder

  field :created_at, type: DateTime

  field :mobile, type: String
  field :flower_type, type: Enum
  field :flower_quantity, type: Integer

end
```

鲜花订单

id	created_at	mobile	flower_type	quantity
KEY	DateTime	String	Enum	Number
101	10-13	138…	rose	99

金数据


```
{
  id: 11,
  created_at: '2018-10-13',
  mobile: '13888888888',
  flower_type: 'rose',
  flower_quantity: 99
}
```

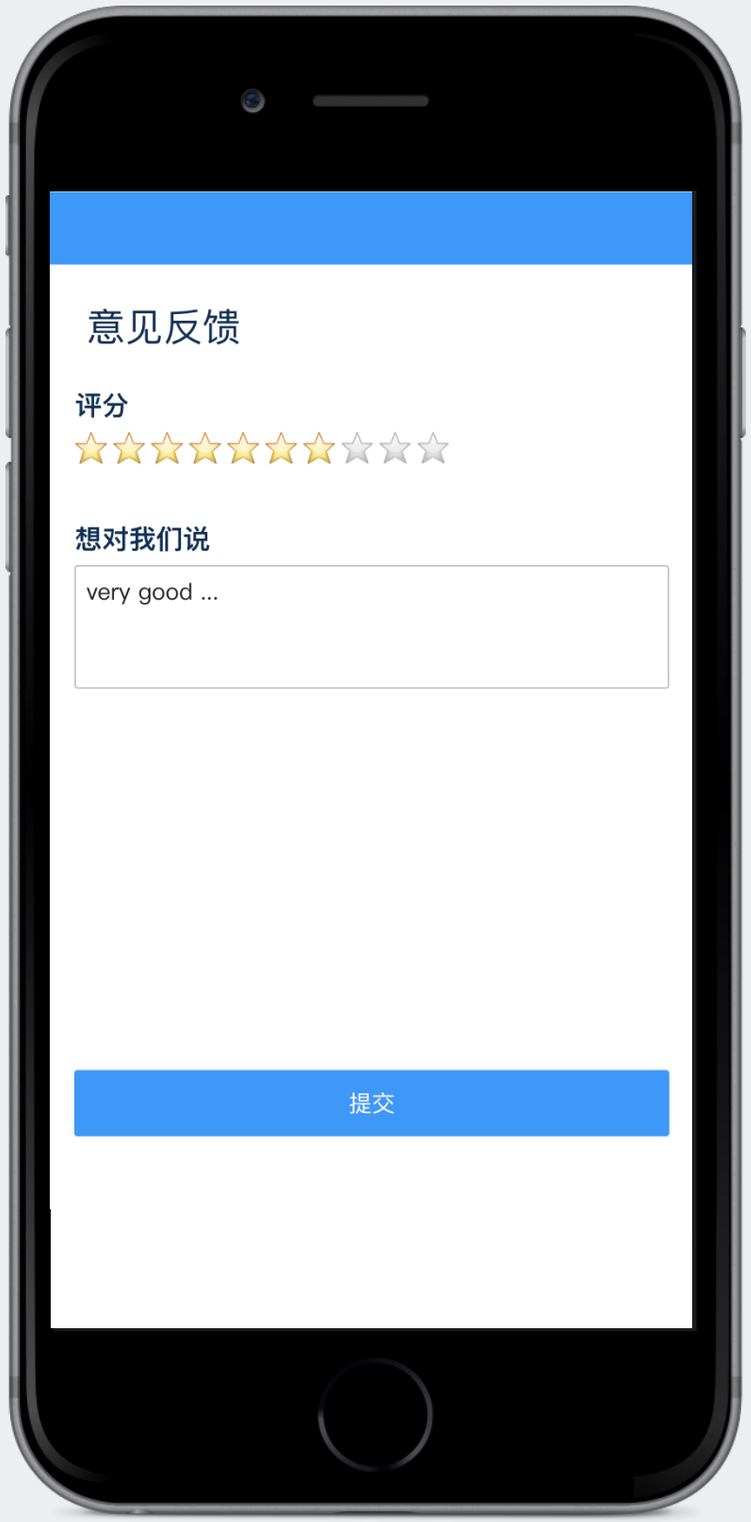
Anna：很不错，但我还想收集客户反馈

```
class Feedback

  field :created_at, type: DateTime
  field :rating, type: Integer
  field :feedback, type: String

end
```

意见反馈

id	created_at	rating	feedback
KEY	DateTime	Number	String
102	10-13	8	Very good …

金数据




```
{
  id: 11,
  created_at: '2018-10-13',
  rating: 8,
  feedback: 'very good'
}
```

Anna：采购记录、库存管理、插花培训报名...

通用的表单/数据系统（金数据）

鲜花订单

id	created_at	mobile	flower_type	quantity
KEY	DateTime	String	Enum	Number
101	10-13	138…	rose	99

意见反馈

id	created_at	rating	feedback
KEY	DateTime	Number	String
102	10-13	8	Very good …

鲜花订单

id	created_at	mobile	flower_type	quantity
KEY	DateTime	String	Enum	Number
101	10-13	138…	rose	99

意见反馈

id	created_at	rating	feedback
KEY	DateTime	Number	String
102	10-13	8	Very good …

鲜花订单

id	created_at	field_1	field_2	field_3
KEY	DateTime	String	Enum	Number
101	10-13	138…	rose	99

意见反馈

id	created_at	field_1	field_2
KEY	DateTime	Number	String
102	10-13	8	Very good …

鲜花订单

id	created_at	field_1	field_2	field_3
KEY	DateTime	String	Enum	Number
101	10-13	138…	rose	99

意见反馈

id	created_at	field_1	field_2
KEY	DateTime	Number	String
102	10-13	8	Very good …

通用数据表

id	created_at	field_1	field_2	field_3	field_4	field_5
KEY	DateTime	String	Enum	Number	Number	String
101	10-13	138…	rose	99		
102	10-13				8	Very good

方案一：扩展字段 Custom Fields

通用数据表

id	created_at	fields
KEY	DateTime	JSON
101	10-13	{“mobile”: “1388888888”, “flower_type”: “rose”, “quantity”: 99}
102	10-13	{“rating”: 8, “feedback”: “very good”}

方案二：嵌入结构化数据
embedded structured data

通用数据表

id	created_at
KEY	DateTime
101	10-13
102	10-13

属性表

id	entry_id	key	value
KEY	KEY	String	
1	101	mobile	138888
2	101	flower_type	rose
3	101	quantity	99
4	102	rating	8
5	102	feedback	Very good

方案三：关联属性表 Attribute Table

通用数据表

id	created_at
KEY	DateTime
101	10-13
102	10-13

属性表

id	entry_id	key	value
KEY	KEY	String	
1	101	mobile	138888
2	101	flower_type	rose
3	101	quantity	99
4	102	rating	8
5	102	feedback	Very good

方案三：关联属性表 Attribute Table

通用数据表

id	created_at
KEY	DateTime
101	10-13
102	10-13

属性表

id	entry_id	key	value
KEY	KEY	String	
1	101	mobile	138888
2	101	flower_type	rose
3	101	quantity	99
4	102	rating	8
5	102	feedback	Very good

方案三：关联属性表 Attribute Table

field_x

- String
- Number
- Date
- Hash
- Array
- ID
- ...

存储 Schema

Storage Schema

Schemaless

- 任意字段（field）
- 任意值（value）

```
{
  id: 11,
  created_at: '2018-10-13',

  mobile: '13888888888',
  flower_type: 'rose',
  flower_quantity: 99
}
```

鲜花订单

```
{
  id: 11,
  created_at: '2018-10-13',

  rating: 8,
  feedback: 'very good'
}
```

意见反馈

金数据

```
{
  id: 11,
  created_at: '2018-10-13',

  field_1: '13888888888',
  field_2: 'rose',
  field_3: 99
}
```

鲜花订单

```
{
  id: 11,
  created_at: '2018-10-13',

  field_1: 8,
  field_2: 'very good'
}
```

意见反馈

```
entry_flower_order.field_1 = '13888888888'
entry_flower_order.field_2 = 'rose'
entry_flower_order.field_3 = 99

entry_feedback.field_1 = 8
entry_feedback.field_2 = 'very good'
```

Ruby
动态类型语言

文档数据库
动态类型数据模型

Schemaless 并不是没有 Schema

- 具体业务具有数据结构
- 代码需要反映业务

金数据

```
{
  id: 11,
  created_at: '2018-10-13',

  field_1: '13888888888',
  field_2: 'rose',
  field_3: 99
}
```

鲜花订单

```
{
  id: 11,
  created_at: '2018-10-13',

  field_1: 8,
  field_2: 'very good'
}
```

意见反馈

金数据

元数据表：表单的定义

- 基本信息（创建人、日期、状态...）
- **业务字段（数组、单表继承）**
    - Key、标题
    - 类型
        - 文本、单选、下拉框、数字...
        - 商品、计算、关联、表格...
    - 校验规则、默认值
    - 业务属性

```
class Form

  field :created_at

  embeds_many :form_fields

  has_many :entries

end
```

金数据

```
{
  id: 11,
  created_at: '2018-10-13',
  field_1: '13888888888',
  field_2: 'rose',
  field_3: 99
}
```

鲜花订单

```
{
  id: 13,
  created_at: '2018-10-13',
  form_fields: [
    {
      _type: 'MobileField',
      key: 'field_1',
      label: '电话',
      validations: ['presence'],
      hidden: false,
      hint: '请输入中国手机号码',
      ...
    },
    ...
    ...
  ]
}
```

金数据

```
form = Form.create form_fields: [
  MobileField.new(key: 'field_1', label: '电话', ...)
]

entry = form.entries.create field_1: 8

entry.valid?
=> false

entry.errors
=> {field_1: 'data type error ...'}
```

隐式的 Schema（Implicit Schema）

```
form_flower_order.entries.create field_1: '13888888888'
form_flower_order.entries.create field_1: '13666666666'

form_feedback.entries.create field_1: 13888888888
form_feedback.entries.create field_1: 13666666666

Entry.where(field_1: '13888888888')
Entry.where(field_1: 13888888888)
```

查询的 Schema（Predicate Schema）

Anna：采购记录、库存管理、插花培训报名...通用的表单/数据系统

Anna: 生意做大了，升级为连锁花店

```ruby
class User
end

class Form
  belongs_to :user
end

class Subscription
  belongs_to :user
end
```

```
Form.count
{
  "count" => "forms",
  "query" => {}
}

user.forms.create
{
  "insert" => "forms",
  "document" => {"user_id" => 28}
}

Form.where(status: 'published')
{
  "find" => "forms",
  "filter" => {"status" => "published"}
}
```

企业版？加一张表、一个字段就OK

```ruby
class Company
end

class User
  belongs_to :company
end

class Form
  belongs_to :company
  belongs_to :user
end

class Subscription
  belongs_to :company
  belongs_to :user
end
```

```
Form.count
{
  "count" => "forms",
  "query" => {}
}

user.forms.create
{
  "insert" => "forms",
  "document" => {"user_id" => 28}
}

Form.where(status: 'published')
{
  "find" => "forms",
  "filter" => {"status" => "published"}
}
```

```
Form.where(company_id: 13).count
{
  "count" => "forms",
  "query" => {"company_id" => 13}
}

user.forms.create(company_id: 13)
{
  "insert" => "forms",
  "document" => {"company_id" => 13, "user_id" => 28}
}

Form.where(company_id: 13, status: 'published')
{
  "find" => "forms",
  "filter" => {"company_id" => 13, "status" => "published"}
}
```

```
Form.where(company_id: 13).count
{
  "count" => "forms",
  "query" => {"company_id" => 13}
}

user.forms.create(company_id: 13)
{
  "insert" => "forms",
  "document" => {"company_id" => 13, "user_id" => 28}
}

Form.where(company_id: 13, status: 'published')
{
  "find" => "forms",
  "filter" => {"company_id" => 13, "status" => "published"}
}
```

金数据的挑战

- 10万级别的代码库（们）
- Controllers/Services/Models/Background Jobs

期望

- 数据在企业间隔离
- 代码修改量最小
- 对程序员透明

SaaS: Multi-Tenancy 多租户系统

- 租户 Tenant == 企业 Company

- tenant_id == company_id

- current_tenant == current_company

```ruby
class TenantBaseController

  before_action :set_tenant

  def set_tenant
    Tenant.current_tenant = fetch_tenant_from_request
  end

  def fetch_tenant_from_request
    # from subdomain, https://rubysummit.jinshuju.com
    # from url, https://form.rubysummit.com.cn
    # from request params
  end

end
```

Controller

```ruby
class TenantBaseController

  before_action :set_tenant

  def set_tenant
    Tenant.current_tenant = fetch_tenant_from_request
  end

  def fetch_tenant_from_request
    # from subdomain, https://rubysummit.jinshuju.com
    # from url, https://form.rubysummit.com.cn
    # from request params
  end

end
```

Controller

```ruby
class Form
  include ModelExtensions

  belongs_to :user
end
```

```ruby
class Subscription
  include ModelExtensions

  belongs_to :user
end
```

Model

```ruby
module ModelExtensions

  included do
    belongs_to :company

    tenant_scope = lambda do
      where(company_id: Tenant.current_tenant)
    end

    default_scope tenant_scope
  end

end
```

Model

```ruby
module ModelExtensions

  included do
    belongs_to :company

    tenant_scope = lambda do
      where(company_id: Tenant.current_tenant)
    end

    default_scope tenant_scope
  end

end
```

Model

```ruby
module ModelExtensions

  included do
    belongs_to :company

    tenant_scope = lambda do
      where(company_id: Tenant.current_tenant)
    end

    default_scope tenant_scope
  end

end
```

Model

```ruby
module ModelExtensions

  included do
    belongs_to :company

    tenant_scope = lambda do
      where(company_id: Tenant.current_tenant)
    end

    default_scope tenant_scope
  end

end
```

Model

Query
Model Initialization

default_scope? WHAT!

```ruby
module ModelExtensions

  included do
    belongs_to :company

    tenant_scope = lambda do
      where(company_id: Tenant.current_tenant)
    end

    default_scope tenant_scope
  end

end
```

Model

Query
Model Initialization

```
Form.count
{
  "count" => "forms",
  "query" => {}
}

user.forms.create
{
  "insert" => "forms",
  "document" => {"user_id" => 28}
}

Form.where(status: 'published')
{
  "find" => "forms",
  "filter" => {"status" => "published"}
}
```

```
Form.count
{
  "count" => "forms",
  "query" => {"company_id" => 13}
}

user.forms.create
{
  "insert" => "forms",
  "document" => {"company_id" => 13, "user_id" => 28}
}

Form.where(status: 'published')
{
  "find" => "forms",
  "filter" => {"company_id" => 13, "status" => "published"}
}
```

目录

总结

SaaS业务广度：通用的表单/数据模型

SaaS业务深度：Multi-Tenancy 多租户模型

期待你的加入

@西安

@成都

Ruby/Rails