

RESTful Controllers

Kas ir REST

Definēja Roy Fielding



REpresentational State Transfer

Web aplikāciju arhitektūras stils

Balstīts uz resursiem un darbībām ar tiem

REST elementi

- Resursi – “lietvārdi”
<http://blog/posts> <http://blog/post/1>
- Standarta darbības – “darbības vārdi”
GET, POST, PUT, DELETE
- HTTP pieprasījumi – “teikumi”
GET <http://blog/post/1>
- Reprzentācija – attēlošanas formāts
HTML, XML, CSV, ...

Kāda vaina šim?

GET <http://blog/post/destroy/1>

- darbība iekļauta URI
- darbība pretrunā ar HTTP pieprasījumu

Un šim?

GET http://www.saeima.lv/saeima9/viena_9.d_fra?KODS=F801

- principā OK
- vienīgi URI “lietvārds” ir nesaprotams simbolu kopums

CRUD darbības vārdi

HTTP	CRUD	SQL	Controller action
POST	Create	INSERT	create
GET	Read	SELECT	show
PUT	Update	UPDATE	update
DELETE	Delete	DELETE	destroy

Rails resursu URLi

```
map.connect ':controller/:action/:id' map.resources :posts
```

Tradicionāli	RESTful pieeja
POST /posts/create	POST /posts
GET /posts/show/1	GET /posts/1
POST /posts/update/1	PUT /posts/1
POST /posts/destroy/1	DELETE /posts/1

Rails resursu reprezentācija

Viens resurss / kontrolieris,
dažādas reprezentācijas / formāti

Pieprasījums	Rezultāts
GET /posts	HTML
GET /posts.xml	XML
GET /posts.csv	CSV

Nestandarta darbības

- “Sameklēt visus ierakstus XML formātā, kuri virsraksti sākas ar ‘Pirmais’”

GET [http://blog/posts.xml;search?
title=Pirmais](http://blog/posts.xml;search?title=Pirmais)

Skatījumu palīgmetodes

Action	HTTP	Path metode
index	GET /posts	posts_path
show	GET /posts/1	post_path(1)
new	GET /posts/new	new_post_path
edit	GET /posts/1;edit	edit_post_path
create	POST /posts	posts_path
update	PUT /posts/1	post_path(1)
delete	DELETE /posts/1	post_path(1)

Pakļautie resursi

```
map.resources :posts do |posts|  
  posts.resources :comments  
end
```

Path metode	URI
comments_path(1)	/posts/1/comments
comment_path(1, 2)	/posts/1/comments/2
new_comment_path(1)	/posts/1/comments/new
edit_comment_path(1)	/posts/1/comments/2;edit

ActiveResource

- ActiveRecord analogs, lai strādātu ar REST resursiem
- Objektorientēta pieeja REST web servisiem
- Iekļauts Rails 2.0

Standarta metodes

```
class Post < ActiveResource::Base  
  self.site = "http://blog"  
end
```

```
posts = Post.find(:all)  
post = Post.find(1)
```

```
post.save
```

```
post.destroy
```

Ieraksta izveidošanas piemērs

```
post = Post.new(:title => "Pirmais ieraksts",  
                :body => "Ļoti interesants")  
post.save
```

```
# POST http://blog/posts.xml  
<post>  
  <title>Pirmais ieraksts</title>  
  <body>Ļoti interesants</body>  
</post>
```

```
# HTTP/1.1 200 OK  
# Location: /posts/1.xml  
# Body: <empty>
```

REST ieguvumi

- vienkārši un standartizēti URLi
- dažādas reprezentācijas
- mazāks koda apjoms
- CRUD orientēti kontrolieri
- standartizēta aplikācijas arhitektūra
- mērogojamība