

Rails Workshop

Routing & Ressourcen

Router

Mapping von URLs auf Actions in Controllern

```
URL: GET /patients/17
```

```
Route: get '/patients/:id', to: 'patients#show'
```

Router

- Generieren von Pfaden und URLs

```
Route:      get '/patients/:id', to: 'patients#show', as: ,patient'  
Controller: @patient = Patient.find(17)  
View:      link_to 'Patient Record', patient_path(@patient)  
HTML:      <a href="/patients/17">Patient Record</a>
```

Ressourcen

- HTTP Methoden
 - GET
 - POST
 - PATCH/PUT
 - DELETE

Ressourcen

HTTP Verb	Path	Controller#Action	Used for
GET	/photos	photos#index	zeige eine Liste aller Fotos
GET	/photos/new	photos#new	Formular um ein neues Foto anzulegen
POST	/photos	photos#create	Erstellen eines Fotos
GET	/photos/:id	photos#show	Zeige ein bestimmtes Foto
GET	/photos/:id/edit	photos#edit	Formular um ein bestimmtes Foto bearbeiten
PATCH/PUT	/photos/:id	photos#update	Ein bestimmtes Foto ändern
DELETE	/photos/:id	photos#destroy	Ein bestimmtes Foto löschen

Ressourcen

resources :photos

Ressourcen

- `photos_path` liefert `/photos`
- `new_photo_path` liefert `/photos/new`
- `edit_photo_path(:id)` liefert `/photos/:id/edit`
- `photo_path(:id)` liefert `/photos/:id`

Wann sollte ich
Ressourcen einsetzen
und wann nicht?

Ressourcen

- Grundsätzlich immer
 - denn Ressourcen sind sehr flexibel
- außer
 - es existiert kein Model auf dem ich arbeite
 - ich arbeite nicht mit einem Model (z.B. nur Anzeige von Daten)
 - der Controller aggregiert Daten von vielen Models.

Ich habe aber gar keine „Bearbeiten“ Funktion für mein Model.

Eingeschränkte Ressourcen

```
resources :photos, only: [:index, :show]  
resources :photos, except: :destroy
```

Ich habe eine Ressource, die für jeden Nutzer unterschiedlich ist, aber ohne ID aufgerufen wird.

z.B. einen Warenkorb

Singular Ressourcen

resource :cart

HTTP Verb	Path	Controller#Action	Used for
GET	/cart/new	carts#new	Formular um einen neuen Warenkorb anzulegen
POST	/cart	carts#create	Erstellt einen neuen Warenkorb
GET	/cart	carts#show	Zeigt den Warenkorb des Nutzers
GET	/cart/edit	carts#edit	Formular zum ändern des Warenkorbs
PATCH/PUT	/cart	carts#update	Ändert den Warenkorb
DELETE	/cart	carts#destroy	Löscht den Warenkorb des Nutzers

Ich möchte meine Routen (und damit auch die Controller) besser strukturieren.

Controller in Underordner, Ressourcen gruppieren.

Namespaces

```
namespace :admin do
  resources :posts, :comments
end
```

HTTP Verb	Path	Controller#Action	Named Helper
GET	/admin/posts	admin/posts#index	admin_posts_path
GET	/admin/posts/new	admin/posts#new	new_admin_post_path
POST	/admin/posts	admin/posts#create	admin_posts_path
GET	/admin/posts/:id	admin/posts#show	admin_post_path(:id)
GET	/admin/posts/:id/edit	admin/posts#edit	edit_admin_post_path(:id)
PATCH/PUT	/admin/posts/:id	admin/posts#update	admin_post_path(:id)
DELETE	/admin/posts/:id	admin/posts#destroy	admin_post_path(:id)

verschachtelte Ressourcen

```
class Magazine < ActiveRecord::Base
  has_many :ads
end
```

```
class Ad < ActiveRecord::Base
  belongs_to :magazine
end
```

```
resources :magazines do
  resources :ads
end
```

HTTP Verb	Path	Controller#Action
GET	/magazines/:magazine_id/ads	ads#index
GET	/magazines/:magazine_id/ads/new	ads#new
POST	/magazines/:magazine_id/ads	ads#create
GET	/magazines/:magazine_id/ads/:id	ads#show
GET	/magazines/:magazine_id/ads/:id/edit	ads#edit
PATCH/PUT	/magazines/:magazine_id/ads/:id	ads#update
DELETE	/magazines/:magazine_id/ads/:id	ads#destroy

oberflächlich verschachtelte Ressourcen

```
resources :posts do  
  resources :comments, only: [:index, :new, :create]  
end  
resources :comments, only: [:show, :edit, :update, :destroy]
```

==

```
resources :posts do  
  resources :comments, shallow: true  
end
```

Generieren von Pfaden und URLs aus Ressourcen

```
resources :magazines do
  resources :ads
end
```

```
link_to 'Ad details', magazine_ad_path(@magazine, @ad)
url_for([@magazine, @ad])
link_to 'Ad details', [@magazine, @ad]
=> /magazines/1/ads/2
```

```
link_to 'Magazine details', @magazine
=> /magazines/1
```

```
link_to 'Edit Ad', [:edit, @magazine, @ad]
=> /magazines/1/ads/2/edit
```

Ich benötige zusätzliche Routen an meinen Ressourcen.

z.B. für die Suche oder eine Vorschau

Member Routen an Ressourcen

```
resources :photos do
  member do
    get 'preview'
  end
end
```

```
resources :photos do
  get 'preview', on: :member
end
```

```
=> /photos/1/preview
=> params[:id] = 1
```

Collection Routen an Ressourcen

```
resources :photos do
  collection do
    get 'search'
  end
end
```

```
resources :photos do
  get 'search', on: :collection
end
```

```
=> /photos/search
```

Weitere Anpassungen an Ressourcen

- Anderer Controller:
resources **:photos**, controller: *,images'*
- Bedingungen für Parameter:
resources **:photos**, constraints: {id: */[A-Z][A-Z][0-9]+/*}
- Helper Namen anpassen/prefixen:
resources **:photos**, as: *,images'*
- Action Namen anpassen/übersetzen:
resources **:photos**, path_names: { new: *'make'*, edit: *'change'* }

Nicht Ressourcen
Routen

- Standard:
get `'/patients/:id'` , to: `,patients#show'`
- Dynamische Segmente:
get `':controller/:action/:id/:user_id'`
=> `params[:id]`
=> `params[:user_id]`
- Statische Segmente:
get `',:controller/:action/:id/with_user/:user_id'`
- Query String: `/photos/show/1?user_id=2`
=> `params[:id] = 1`
=> `params[:user_id] = 2`

- Parameter Defaults:

get *'photos/:id'*, to: *'photos#show'*, defaults: { format: *'jpg'* }
=> params[:format] = ,jpg'

- Benannte Routen:

get *'exit'*, to: *'sessions#destroy'*, as: **:logout**
=> Helper: logout_path

Bedingungen

- HTTP Methoden Bedingungen:
match 'photos', to: 'photos#show', via: **[:get, :post]**
- Segment Bedingungen:
get 'photos/:id', to: 'photos#show', id: /**[A-Z]\d{5}**/
- Request Bedingungen:
get 'photos', constraints: {subdomain: *admin*}

Weiterleitung

- Standard:
get `'/stories'`, to: `redirect(, '/posts')`
- Mit Parameter:
get `'/stories/:name'`, to: `redirect('/posts/{name}')`

Root

- Standard Route, wenn Requests auf / gehen.
Ist in der Regel die Homepage oder Einstiegsseite.
root to: `,pages#main'`
- Im Namespace kann es weitere Roots geben:
namespace **:admin** do
 root to: "admin#index"
end

Unicode Routes

- Selbst das ist möglich:
get 'こんにちは', to: 'welcome#index'

Fragen?

Vielen Dank für eure
Aufmerksamkeit.