

Full Stack Notes

[Rails Overview](#) / URLs are Routes

URLs are Routes

Routes are how our Rails application turns the URLs it receives into calls to a Controller action.

Table of Contents

- 1 [Configuring Routes](#)
- 2 [Resource Based Routes](#)
- 3 [The Seven RESTful Routes](#)
- 4 [Selecting Specific RESTful Routes](#)
- 5 [Manual Routing](#)
- 6 [Setting the Root Route](#)
- 7 [The link_to Helper](#)

Configuring Routes

Routes are configured in the `config/routes.rb` file.

The complete routing documentation can be found in the [Rails Routing Guide](#).

Resource Based Routes

If, for example, we are building routes for a Products controller and we want [all seven RESTful CRUD routes](#) we need only add:

```
resources :products
```

The Seven RESTful Routes

The above `resources` command would generate the following routes:

HTTP Verb	Path	Controller#Action	Used for
GET	/products	products#index	display a list of all products
GET	/products/new	products#new	display HTML form for creating a product
POST	/products	products#create	create a new product

HTTP Verb	Path	Controller#Action	Used for
GET	/products/:id	products#show	display a specific product
GET	/products/:id/edit	products#edit	display HTML form for editing a product
PATCH/PUT	/products/:id	products#update	update a specific product
DELETE	/products/:id	products#destroy	delete a specific product

Selecting Specific RESTful Routes

If we only wanted the `index` and `show` routes:

```
resources :products, only: [:index, :show]
```

We can also exclude specific routes:

```
resources :products, except: [:edit, :update, :delete]
```

Manual Routing

We can also manually generate routes by following this pattern:

```
httpverb 'url/path' => 'controller#action', as: 'named_route'
```

So to generate the `index` route for a products controller:

```
get 'products' => 'products#index', as: 'products'
```

If we want to generate a member route (one that makes reference to a specific product by id) like `show`:

```
get 'products/:id' => 'products#show', as: 'product', id: /\d+/'
```

Member routes like this contain placeholders (in this case `:id`, which is constrained to a number with the regular express at the end). The actual id sent via the URL will be accessible in the controller action via the `params` hash. So the products `show` action might look like this:

```
def show # Within the products controller
  @product = Product.find(params[:id]) # Find the product with the primary key mentioned in the URL.
end
```

Setting the Root Route

In the Rails routing file (`config/routes.rb`) we can specify which controller action to execute when our users visit the root of our domain.

For example, to set the `products` controller's `index` action as the domain root:

```
root to: 'products#index'
```

When using our development server we can invoke this route by visiting <http://localhost:3000/>

The `link_to` Helper

Once you have routes in place you may want to link to these routes from a view. We use the `link_to` helper to build our links in Rails.

Linking to some of the routes defined above:

```
# Links to the "products" route:  
<%= link_to 'All Products', products_path %>  
  
# Links to the "product" route, given a Product variable named "product":  
<%= link_to product.name, product %>
```

```
# Links to the root route:
```

```
<%= link_to 'Home', root_path %>
```
