Action View Helpers

1.1 AssetTagHelper

Provides methods for generating HTML that links views to assets (images, JS files, stylesheets, feeds etc).

Direct rails to link to assets from a dedicated assets server (config.asset host).

```
config.asset_host = "assets.example.com"
image_tag("rails.png")
```

Means the same thing as <img src="http://assets.example.com/images/rails.png" />

auto_discovery_link_tag

Returns a link tag that browsers and feed readers can use to auto detect rss json etc

Image_path

Computes path to an image asset (app/assets/images) directory.
Can also add a fingerprint for added uniqueness if needed. (Set config.assets)

```
image_path(“edit.png”)
```

Image_url

Computes a URL to an image asset in the app/assets/images dir.
Merges with current or asset host.

```
image_url(“edit.png”)
```

Image_tag

Returns an HTML image tag for the source (full path or a file that exists in app/assets/images)

```
image_tag(“icon.png”)
```

Javascript_include_tag

ReturnsHTML script tag.
Can pass the filename of the JS file within the app/assets/javascripts

```
javascript_include_tag "common"
```

Javascript_path

Computes path to JavaScript asset (app/asset/javascripts) directory. If .js is not added in parameter, it is appended.

javascript_path = “common”

Javascript_url

Computes the URL to a JavaScript asset (calls javascript_path internally) + merges asset host

javascript_url "common"
# => http://www.example.com/assets/common.js

Picture_tag

Returns HTML picture tag

picture_tag("icon.webp", "icon.png")

Preload_link_tag

Returns a link tag to preload the source.

preload_link_tag “application.css”

Stylesheet_link_tag

Returns a stylesheet link tag for the sources (given in arguments).

stylesheet_link_tag “application”

stylesheet_path

Path to stylesheet asset (app/assets/stylesheets)

Stylesheet_path “application”

stylesheet_url

URL to stylesheet asset (app/assets/stylesheets)

Stylesheet_url “application”

AtomFeedHelper

Atom_feed : For building an Atom feed easily.

BenchmarkHelper

Benchmark: Measures the executive time of a block in a template and record the result to the log.

```
<% benchmark "Process data files" do %>
  <%= expensive_files_operation %>
<% end %>
```

CacheHelper

Cache: Method for caching fragments of view (menus, lists, HTML fragments etc).

```
<% cache do %>
  <%= render "application/footer" %>
<% end %>
```

CaptureHelper

capture method: Extract part of a template into a variable (can be used anywhere in layout).

```
<% @greeting = capture do %>
  <p>Welcome! The date and time is <%= Time.now %></p>
<% end %>
```

content_for: Stores a block of markup in an identifier (if it needs to be used later). Can be passed as an argument to yield.

For example: rendering javascript code within a file

DateHelper

distance_of_time_in_words

The distance in time between two Time or Date objects or integers as (seconds).

Include_seconds = true -> for more detailed approximation

Time_ago_in_words

To_time is Time.now in this case.

DebugHelper

pre: This tag is returned by debug helper and contains the object dump. Can use it to inspect objects.

FormHelper

This helps generate HTML for forms + with methods for each input e.g. text, password etc. After the form is submitted, the form input is sent as a params object to the controller.

JavaScriptHelper

escape_javascript
Single and double quotes for JavaScript segments.

javascript_tag
JavaScript tag wrapping the provided code.

NumberHelper

Methods for converting numbers to formatted strings.

number_to_currency
Formats a number into a currency string.

number_to_human
Prettifying / formatting a number for readability.

number_to_human_size
More understandable representation in size e..g 1.21 KB

number_to_percentage
Formats a number as a percentage string. E.g. 100%

number_to_phone
Formats a number into a phone num (US is default).

number_with_delimiter

Grouped thousands: 12345678 -> 12,345,678

number_with_precision
Formats a number with the specified level of precision (3 default).

SanitizeHelper
Method for removing text of undesirable HTML elements.

sanitize: Encodes all HTML tags and strips all non specifically allowed elements.

If :attributes and :tags options are passed, only mentioned attr/tags are allowed

sanitize_css: Sanitizes a block of CSS code.

strip_links(html): Strips all link tags from texts only leaves the link text.

strip_tags(html): Strips all HTML tags as well as comments.

UrlHelper
Provides methods to make links and get URLs etc

url_for
Return the URL for the set of options.
E.g url_for @profile gives /profiles/1

link_to
Links to a URL derived from url_for. (Creates a RESTful resource link)

link_to "Profile", @profile -> <a href="/profiles/1">Profile</a>

button_to
Generates a form and has a submit button.

CsrfHelper

Returns meta tags “csrf-param” and “csrf-token” with the name of cross-site req forgery protection parameter and token.

<%= csrf_meta_tags %>