Tr8n Cheat Sheet

Gender Based Tokens

Definition: you can see/modify this definition in the **config/tr8n/config.json** file

```
"gender_token": {
  "suffixes": ["user", "profile", "actor", "target", "partner", "parent", "child", "sibling"],
  "method":   "gender",
  "values":   {"unknown": "u", "neutral": "n", "female": "f", "male": "m"}
}
```

Simple token:

```
<%= tr("Dear {user}", nil, :user => current_user) %>
```

Token with embedded method calls:

```
<%= tr("Dear {user.first_name} {user.last_name}", nil, :user => current_user) %>
```

Token with decorative substitution:

```
<%= tr("Dear {user}", nil, :user => [current_user, display_profile(current_user)]) %>
```

Token with decorative substitution using symbol method call:

```
<%= tr("Dear {user}", nil, :user => [current_user, :first_name])  %>
```

Token with decorative substitution using symbol method call with parameters:

```
<%= tr("Dear {user}", nil, :user => [current_user, :some_method, "value"])  %>
```

Token with decorative substitution using lambda method call:

```
<%= tr("Dear {user}", nil, :user => [current_user, lambda{|val| html_for(val)}])  %>
```

Token with decorative substitution using lambda method call with parameters:

```
<%= tr("Dear {user}", nil, :user => [current_user, lambda{|val, test| html_for(val, test)}], "test"])  %>
```

Gender based sentence with a hidden token:

```
<%= tr("{user} changed {_his_her} name", nil, :user => [current_user, :name], :_his_her => current_user.his_her)  %>
```

Number Based Tokens

Definition: you can see/modify this definition in the **config/tr8n/config.json** file

```
"numeric_token": {
  "suffixes": ["count", "num", "age", "hours", "minutes", "years", "seconds"],
  "method":   "to_i"
}
```

Number based sentence with a hidden token:

```
<%= tr("You have {count} {_messages}", nil, :count => i, :_messages => "message".pluralize_for(i))%>
```

Tr8n Cheat Sheet

Decoration Tokens

Definition: you can add/modify decorations in the **config/tr8n/default_lambdas.json** file

```
{
  "bold":       "<strong>{$0}</strong>",
  "italic":     "<i>{$0}</i>",
  "link":       "<a href='{$1}' style='{$2}'>{$0}</a>"
}
```

Note: $0 parameter is always the translated value of the lambda

Bold decoration with default docarator:

```
<%= tr("[bold: Hello World]”)%>
```

Link with style using default decorator:

```
<%= tr("[link: Hello World]”, nil, :link => [“http://www.google.com”, “text-decoration:none”])%>
```

Bold decoration with custom string substitution:

```
<%= tr("[bold: Hello World]”, nil, :bold => "<strong>{$0}</strong>")%>
```

Link with gender dependent tokens and lambda decoration:

```
<%= tr("{user} updated [link: {_his_her} profile]", nil,
      :user       => [current_user, display_profile(current_user)],
      :_his_her   => current_user.his_her,
      :link       => lambda{|value| display_profile(value)}
)  %>
```

Link with number dependent tokens and lambda decoration:

```
<%= tr("You have [link: {count} {_messages}]", nil,
      :count      => i,
      :_messages => "message".pluralize_for(i),
      :link       => lambda{|value| link_to(value, “/inbox”)}
)  %>
```

Sentence with gender dependent and number dependent tokens and a link decoration:

```
<%= tr("{user} added {user1}, {user2}, {user3} and [link: {count} other family {_members}] to {_his_her} tree", nil,
      :user                => [actor, display_profile(actor)],
      :user1               => display_profile(user1),
      :user2               => display_profile(user2),
      :user3               => display_profile(user3),
      :count               => i,
      :_members            => "member".pluralize_for(i),
      :_his_her            => actor.his_her,
      :link                => lambda{|value| link_to(value, “/tree”)}
) %>
```

Note: this sentence is an example of a complicated structure that would be translated into 9 variations in Russian language:

```
     3 gender variation on {user} for “added” and “{_his_her}”
     3 numeric variations on {count} for “{_members}”

     total: 3 * 3 = 9 combintations
```