source: urlpatterns.py

Format suffixes

> Section 6.2.1 does not say that content negotiation should be used all the time.
>
> — Roy Fielding, <u>REST discuss mailing list</u>

A common pattern for Web APIs is to use filename extensions on URLs to provide an endpoint for a given media type. For example, 'http://example.com/api/users.json' to serve a JSON representation.

Adding format-suffix patterns to each individual entry in the URLconf for your API is error-prone and non-DRY, so REST framework provides a shortcut to adding these patterns to your URLConf.

format_suffix_patterns

Signature: format_suffix_patterns(urlpatterns, suffix_required=False, allowed=None)

Returns a URL pattern list which includes format suffix patterns appended to each of the URL patterns provided.

Arguments:

- **urlpatterns**: Required. A URL pattern list.
- **suffix_required**: Optional. A boolean indicating if suffixes in the URLs should be optional or mandatory. Defaults to `False`, meaning that suffixes are optional by default.
- **allowed**: Optional. A list or tuple of valid format suffixes. If not provided, a wildcard format suffix pattern will be used.

Example:

```
from rest_framework.urlpatterns import format_suffix_patterns
from blog import views

urlpatterns = [
    url(r'^/$', views.apt_root),
    url(r'^comments/$', views.comment_list),
    url(r'^comments/(?P<pk>[0-9]+)/$', views.comment_detail)
]

urlpatterns = format_suffix_patterns(urlpatterns, allowed=['json', 'html'])
```

When using `format_suffix_patterns`, you must make sure to add the `'format'` keyword argument to the corresponding views. For example:

```
@api_view(('GET', 'POST'))
def comment_list(request, format=None):
    # do stuff...
```

Or with class based views:

```
class CommentList(APIView):
    def get(self, request, format=None):
        # do stuff...

    def post(self, request, format=None):
        # do stuff...
```

The name of the kwarg used may be modified by using the `FORMAT_SUFFIX_KWARG` setting.

Also note that `format_suffix_patterns` does not support descending into `include` URL patterns.

Using with `i18n_patterns`

If using the `i18n_patterns` function provided by Django, as well as `format_suffix_patterns` you should make sure that the `i18n_patterns` function is applied as the final, or outermost function. For example:

```
url patterns = [
    …
]

urlpatterns = i18n_patterns(
    format_suffix_patterns(urlpatterns, allowed=['json', 'html'])
)
```

Accept headers vs. format suffixes

There seems to be a view among some of the Web community that filename extensions are not a RESTful pattern, and that `HTTP Accept` headers should always be used instead.

It is actually a misconception. For example, take the following quote from Roy Fielding discussing the relative merits of query parameter media-type indicators vs. file extension media-type indicators:

"That's why I always prefer extensions. Neither choice has anything to do with REST." — Roy Fielding, REST discuss mailing list

The quote does not mention Accept headers, but it does make it clear that format suffixes should be considered an acceptable pattern.