

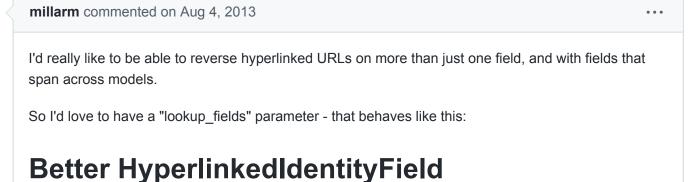
# Support multiple parameters for Hyperlinked fields? #1024

New issue



millarm opened this issue on Aug 4, 2013 · 7 comments





from rest\_framework.relations import HyperlinkedIdentityField
from rest\_framework.reverse import reverse

class ParameterisedHyperlinkedIdentityField(HyperlinkedIdentityField):
 """
 Represents the instance, or a property on the instance, using hyperlinking.

## Assignees

No one assigned

#### Labels

None yet

#### **Projects**

None yet

#### Milestone

No milestone

Linked pull requests

```
lookup_fields is a tuple of tuples of the form:
    ('model_field', 'url_parameter')
11 11 11
lookup_fields = (('pk', 'pk'),)
def __init__(self, *args, **kwargs):
    self.lookup_fields = kwargs.pop('lookup_fields', self.lookup_fields)
    super(ParameterisedHyperlinkedIdentityField, self).__init__(*args, **kwargs)
def get_url(self, obj, view_name, request, format):
    Given an object, return the URL that hyperlinks to the object.
   May raise a `NoReverseMatch` if the `view_name` and `lookup_field`
    attributes are not configured to correctly match the URL conf.
    kwargs = \{\}
    for model_field, url_param in self.lookup_fields:
        attr = obj
       for field in model_field.split('.'):
            attr = getattr(attr, field)
        kwargs[url_param] = attr
    try:
        return reverse(view_name, kwargs=kwargs, request=request, format=format)
    except NoReverseMatch:
        pass
    raise NoReverseMatch()
```

Successfully merging a pull request may close this issue.

None yet

3 participants







<u>4</u> 3



millarm commented on Aug 4, 2013

Author •••

this lets me do things like:

```
vote_url = ParameterisedHyperlinkedIdentityField(
    view_name="vote", lookup_fields=(('poll.id', 'poll_id'), ('id', 'choice_id')))
```

where the urlscheme has <poll\_id> and <choice\_id> and the Choice object has a Foreign Key to a poll object





### tomchristie commented on Aug 15, 2013

Member

• • •

Related: https://groups.google.com/forum/#!searchin/django-rest-framework/relevant\$20key/django-rest-framework/rNGOVUAIv9Q/J2io9Nfvc74J

Also related: http://django-rest-framework.org/api-guide/generic-views.html#creating-custom-mixins

I would love to see someone write a good third party package that properly deals with this, but I don't want to have to maintain it in core (there's already plenty of maintaince overhead on the project). I think this needs to be covered both by a custom hyperlinked relational field, that accepts lookup\_fields instead of lookup\_field, a view mixin that does the same, and ideally also a router or router mixin that'd deal with automatically routing viewsets using lookup\_fields - there's plenty of work there for a good useful 3rd party package, and would absolutely want to link to that from the main docs if you or someone else wants to take it on.

Like I say though it's not something I'm willing to take on in core right now. I do think it'd be super useful to have, but there's no reason it can't be handled as a seperate package.



#### millarm commented on Aug 15, 2013

Author

• • •

Tom.

Is there a wiki with details of third party packages?

We've already created and tested the first part if this, and may be motivated to do the latter parts & package and push back out for the community.

Matt

On Thursday, August 15, 2013, Tom Christie wrote:

#### Related:

https://groups.google.com/forum/#!searchin/django-rest-framework/relevant\$20key/django-rest-framework/rNGOVUAIv9Q/J2io9Nfvc74J

Also related:

http://django-rest-framework.org/api-guide/generic-views.html#creating-custom-mixins

I would love to see someone write a good third party package that properly deals with this, but I don't want to have to maintain it in core (there's already plenty of maintaince overhead on the project). I think this needs to be covered both by a custom hyperlinked relational field, that accepts lookup\_fields instead of lookup\_field, a view mixin that does the same, and ideally also a router or router mixin that'd deal with automatically routing viewsets using lookup\_fields - there's plenty of work there for a good useful 3rd party package, and would absolutely want to link to that from the main docs if you or someone else wants to take it on.

Like I say though it's not something I'm willing to take on in core right now. I do think it'd be super useful to have, but there's no reason it can't be handled as a seperate package.

Reply to this email directly or view it on GitHubhttps://github.com//issues/1024#issuecomment-22725178



#### tomchristie commented on Aug 15, 2013

Member ) •••

@millarm - We can either link to it from appropriate parts of the docs (eg see the '3rd party packages' sections at the end of various parts of the docs, such as authentication). There's also this #1019 which I guess needs a bit more fleshing out but should hopefully get pulled in.



## tomchristie commented on Aug 15, 2013



(In particular I guess we'd link to it from the serializer relations docs and possibly also generic views docs, dep on what your package covers exactly)



## bkovacev commented on Feb 15, 2017

@millarm @tomchristie I was actually looking for something like this and it works as provided by millarm. One question - do you plan to support this in the future?



## tomchristie commented on Feb 16, 2017

Member

. .

**@bkovacev** Not planned no - folks tie themselves up in knots with Hyperlinked fields as it is - supporting multiple lookups in the core REST framework would only make that worse. Much more keen to have it as a third party package that we could link too.



Sign up for free

to join this conversation on GitHub. Already have an account? Sign in to comment

Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About



© 2022 GitHub, Inc.