jsonapi Documentation

Release 0.9.9

Kirill Pavlov

July 09, 2016

Contents

Warning: This documentation is in development and not updated frequently. Check tests and raise issue or question if you have any.

CHAPTER 1

JSONAPI Resource

Resources are related to Django models. Basic customization is done in Resource.Meta class. Exampele of resource declaration is shown below:

```
from jsonapi.api import API
from jsonapi.resource import Resource
from django.conf import settings

api = API()

@api.register
class UserResource(Resource):
    class Meta:
        model = settings.AUTH_USER_MODEL
        authenticators = [Resource.AUTHENTICATORS.SESSION]
        fieldnames_exclude = 'password',


@api.register
class AuthorResource(Resource):
    class Meta:
        model = 'testapp.Author'
        allowed_methods = 'GET', 'POST', 'PUT', 'DELETE'


@api.register
class PostWithPictureResource(Resource):
    class Meta:
        model = 'testapp.PostWithPicture'
        fieldnames_include = 'title_uppercased',
        page_size = 3

@staticmethod
def dump_document_title(obj):
    return obj.title
```

Available Meta parameters:

Name	Type	Default	Usage
model	str	None. Need to specify	'<appname>.<modelname>'
allowed_methods	tuple	('GET')	tuple of capitalized HTTP methods
authenticators	list	[]	[Resource.AUTHENTICATORS.SESSION]
fieldnames_include	list	[]	list of field names
fieldnames_exclude	list	[]	list of field names
page_size	int	None	integer if need pagination
form	django.forms.Form Default	ModelForm	form to use

1.1 GET/POST/PUT/DELETE method kwargs

name	note
request	Django request
ids	ids in request url, optional

1.2 Exceptions

Note: These exceptions are application specific. Than means that they are raised inside resource get/post/put/delete methods handlers but not in api. They are handled in request handler and serialized into correct response document with "errors" attribute on the top level.

Error standard http://jsonapi.org/format/#errors says that every member of error object could be optional, but suggest to use some. JSONAPI defines common exceptions and allows user to raise own ones. It uses code, status, title and detail members for every exception. Sometimed members links, paths and data (not suggested by document) could be added.

Warning: Exceptions are still in development. Codes of existing exceptions would not be changed, but titles could.

Code	Status	Title	Class
32000	400	General JSONAPI Error	JSONAPIError
32001	403	Resource Forbidden Error	JSONAPIForbiddenError
32002	400	Document parse error	JSONAPIParseError
32003	400	Invalid request	JSONAPIInvalidRequestError
32004	400	Invalid request data missing	JSONAPIInvalidRequestDataMissingError
32100	400	Resource Validation Error	JSONAPIResourceValidationError
32101	400	Model Form Validation Error	JSONAPIFormValidationError
32102	400	Model Form Save Error	JSONAPIFormSaveError
32103	400	Database Integrity Error	JSONAPIIntegrityError

CHAPTER 2

Installation

Requires: Django (1.6, 1.7); python (2.7, 3.3).

```
pip install jsonapi
```

CHAPTER 3

Quickstart

Create resource for model, register it with api and use it within urls!

```
# resources.py
from jsonapi.api import API
from jsonapi.resource import Resource

api = API()

@api.register
class AuthorResource(Resource):
    class Meta:
        model = 'testapp.author'

# urls.py
from .resources import api

urlpatterns = patterns(
    '',
    url(r'^api', include(api.urls))
)
```

CHAPTER 4

Notes

REST anti patterns http://www.infoq.com/articles/rest-anti-patterns

CHAPTER 5

Features

What makes a decent API Framework? These features:

- + Pagination
- + Posting of data with validation
- + Publishing of metadata along with querysets
- + API discovery
- Proper HTTP response handling
- Caching
- + Serialization
- Throttling
- + Authentication
- Authorization/Permissions

Proper API frameworks also need:

- Really good test coverage of their code
- Decent performance
- Documentation
- An active community to advance and support the framework

CHAPTER 6

Docs

- Resource definition
- Resource and Models discovery
- Authentication
- Authorization

CHAPTER 7

Examples

curl -v -H “Content-Type: application/vnd.api+json” 127.0.0.1:8000/api/author

Test Application Models