KAZAKH KBTU BRITISH
TECHNICAL
UNIVERSITY

WEB DEVELOPMENT

Lesson 14

related_name attribute in models

```
class Product(models.Model):
  category = models.ForeignKey(Category,
                               on_delete=models.CASCADE,
                               related_name=‘products')
...

category.products.all()
```

instead of **product_set**

Get products of selected category

```
def get_queryset(self):
  try:
    category = Category.objects.get(id=self.kwargs[‘pk'])
  except Category.DoesNotExist:
    raise Http404
  return category.product_set.all()
```

Create product with category

```
category_id = serializers.IntegerField(write_only=True)
```

Get category with its products

1. StringRelatedField
2. PrimaryKeyRelatedField
3. Nested objects

Create category with it’s products

```
def create(self, validated_data):
    products = validated_data.pop(‘product_set')
    category = Category.objects.create(**validated_data)
    bulk = [Product(category=category, **product) for product in products]
    Product.objects.bulk_create(bulk)
    return category
```

```
def create(self, validated_data):
    products = validated_data.pop(‘product_set')
    category = Category.objects.create(**validated_data)
    for product in products:
        Product.objects.create(category=category, **product)
    return category
```

Filtering against the current user

Filtering against query parameters

Django Filters

Django Filters - Installation

```
1) pip install django-filter
```

```
2)
INSTALLED_APPS = [
  ...
  'django_filters',
]
```

Setting filter backends

```
REST_FRAMEWORK = {
  ‘DEFAULT_FILTER_BACKENDS’: (
    ‘django_filters.rest_framework.DjangoFilterBackend',
  )
}
```

Equality-based filtering

```
filterset_fields = (‘name’,)
```

FilterSet

```
filter_backends = (DjangoFilterBackend,)
filter_fields = (‘name’,)
filter_class = (ProductFilter)
```

`http://url?`**`name`**`=text`

SearchFilter

```
filter_backends = (filters.SearchFilter,)
search_fields = (‘name',)
```

http://url?**search**=text

OrderingFilter

```
filter_backends = (filters.OrderingFilter)
ordering_fields = (‘name',)
ordering = ('-name',)
```

http://url?**ordering**=name

Pagination

Pagination

1. `PageNumberPagination`
2. `LimitOffsetPagination`

Pagination

```
REST_FRAMEWORK = {
   ‘DEFAULT_PAGINATION_CLASS': 'rest_framework.pagination.PageNumberPagination',
   'PAGE_SIZE': 10
}
```

Pagination

```
pagination_class = LimitOffsetPagination
```

Questions?