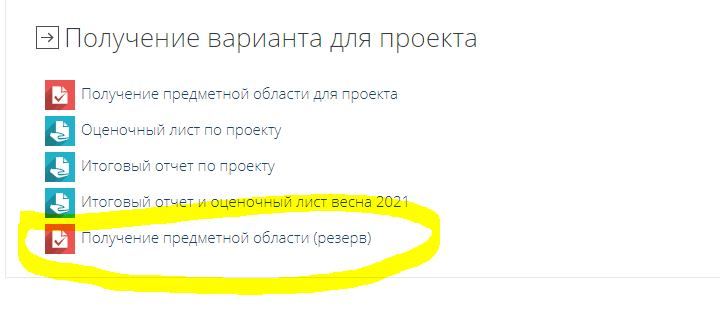
Задание по модулю “Современная разработка на Django”, критерии для проекта

Основа - базовый проект из курса <https://online.mospolytech.ru/course/view.php?id=12602>

Получить предметную область, если это еще не сделано.



В колонке “комментарий” размещены (будут размещены) ссылки на документацию.

Преподаватель - Голубева Ирина Владимировна (телеграм @ivgolubeva), Даньшина Марина Владимировна (телеграм @mdanshina ).

|  |  |  |
| --- | --- | --- |
| № | Критерий | Комментарий |
| 1. | Django REST framework | <https://www.django-rest-framework.org/>  Документация на русском: <https://ilyachch.gitbook.io/django-rest-framework-russian-documentation/>  Валидация полей (минимум 1 метод “своей” логики валидации)  Дополнительные методы для списка @action(methods=['GET'], **detail=False**) и объекта @action(methods=['POST'], **detail=True**) |
| 2. | Использовать Q в запросах | <https://djbook.ru/rel1.9/topics/db/queries.html#complex-lookups-with-q-objects> |
| 3. | Пагинация | <https://www.django-rest-framework.org/api-guide/pagination/> |
| 4. | Фильтрация | <https://www.django-rest-framework.org/api-guide/filtering/>  Реализовать различные варианты, например:   1. возвращать результаты, относящиеся к текущему аутентифицированному пользователю 2. фильтр по именованным аргументам в url 3. фильтр по get параметрам в url 4. django\_filters.rest\_framework.DjangoFilterBackend 5. django\_filters 6. SearchFilter |
| 5. | Django-simple-history | Подключить сохранение истории изменений объектов  <https://django-simple-history.readthedocs.io/en/latest/> |
| 6. | Экспорт данных из админки в эксель | <https://django-import-export.readthedocs.io/en/latest/api_admin.html> |
| 7 | Написать management команды | <https://docs.djangoproject.com/en/4.0/howto/custom-management-commands/> |

**Как будет оцениваться работа**

**Критерии на оценку удовлетворительно:**

1. Реализовано API с помощью Django REST framework для минимум 2х моделей вашей предметной области + **для моделей Comment и Post**
2. Есть минимум 1 запрос c Q, например:

from django.db.models import Q

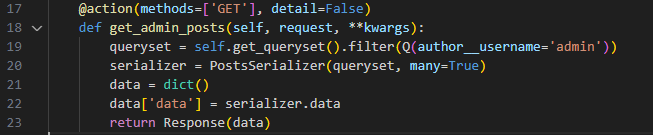
queryset = User.objects.filter(

Q(first\_name\_\_startswith='R')

| Q(last\_name\_\_startswith='D')

) **+**

**Файл Views.py 19 строка здесь применение action GET для**



Запросы должны быть осмысленными и подходящими под вашу предметную область

1. Есть пагинация + **папка Blog файл pagination.py**
2. Есть 1 вариант фильтрации + **папка Blog файл views.py строка**



1. Есть сохранение истории изменений объектов + **для моделей Comments и Posts**
2. Экспорт данных из админки в эксель - реализовано только подключение экспорта данных в адмике, без дополнительных настроек + **папка Blog** **файл admin.py PostAdmin**
3. Написана 1 management команда + **папка Blog -> managements-> get\_user\_email -** **команда get\_user\_email**
4. Реализован на выбор либо @action(methods=['GET'], **detail=False**), либо @action(methods=['POST'], **detail=True**) + **смотри пункт 1**
5. В admin.py есть list\_display, fields, list\_filter, есть Гиперссылки на страницы других объектов, short\_description, fieldsets, inlines. **+ это уже было в том семаке**

**Критерии на оценку хорошо - все задания из критериев на 3+:**

1. Минимум 2 запроса c Q, включающих одновременно OR и AND и NOT ( |, &, ~)

Например:

User.objects.filter(

Q(first\_name\_\_startswith='R') & ~Q(last\_name\_\_startswith='Z')

1. Есть 3 варианта фильтрации:
2. Экспорт данных из админки в эксель

get\_export\_queryset, dehydrate\_{field\_name}, get\_{field\_name}

1. Должна быть обязательно реализована валидация полей
2. Реализованы и @action(methods=['GET'], **detail=False**) и @action(methods=['POST'], **detail=True**)

**Критерии на оценку отлично**

**Выполнены все требования из критериев на 4** и даны в полном объеме ответы на вопросы по Django, выделенные жирным шрифтом.. При ответе следует приводить примеры, можно пользоваться заранее подготовленными материалами:

Django:

**1. Шаблон проектирования Model-View-Controller, его реализация в Джанго (MTV (Model, Template, and View)), особенности и отличия - кратко, можно показывать на примере своего проекта**

2. Что такое CSRF? Какие механизмы есть в джанго для защиты (https://docs.djangoproject.com/en/4.1/ref/csrf/)

3. Клиент посылает запрос на сервер, как вернуть ему какие-то данные в ответ? (JsonResponse, Response) Какие бывают статусы ответов? (2xx, 3xx, 4xx, 5xx + знать по 2-3 примера из каждой группы)

**4. Что такое Middleware, для чего, как реализуется**

5. Запросы на сервер, отличия работы с GET и POST запросами?

**6. Что такое select\_related и prefetch\_related**

7. Что такое Meta в классах Django и для чего нужен

8. Административная панель Django - где настраивается, регистрация моделей - показать на примере своего проекта

9. Административная панель Django - list\_display, list\_filter, fieldsets - показать на примере своего проекта

10.  **Административная панель Django - ModelAdmin.inlines - показать на примере своего проекта**

11. **Административная панель Django - raw\_id\_fields**

12. Функции login, logout, get\_user\_model ( из django.contrib.auth )

REST:

1. **Что такое REST**

2. Отличия viewsets.ViewSet / GenericViewSet /viewsets.ModelViewSet / APIView

3. Дополнительные действия для маршрутизации - @detail\_route, @list\_route,

4. Дополнительные действия для маршрутизации - @action

5. REST framework - поддержка автоматического роутинга (Метод register() и его аргументы)

6. **Сериализаторы в REST framework (что такое, зачем нужно, какие могут быть поля и как они описываются)**

7. Сериализаторы - class Meta, extra\_kwargs

8. Сериализаторы - class Meta, validators

9. reverse и reverse\_lazy - что это за функции и уметь привести пример на основе своего проекта (можно в консоли)

10. Функция get\_serializer\_context

Пример реализации (будет обновляться) <https://github.com/igolubeva/mospol_ip_2022>