Задача: спроектировать и разработать API для системы опросов пользователей.

Функционал для администратора системы:

- авторизация в системе (регистрация не нужна)

- добавление/изменение/удаление опросов. Атрибуты опроса: название, дата старта, дата окончания, описание. После создания поле "дата старта" у опроса менять нельзя

- добавление/изменение/удаление вопросов в опросе. Атрибуты вопросов: текст вопроса, тип вопроса (ответ текстом, ответ с выбором одного варианта, ответ с выбором нескольких вариантов)

Функционал для пользователей системы:

- получение списка активных опросов

- прохождение опроса: опросы можно проходить анонимно, в качестве идентификатора пользователя в API передаётся числовой ID, по которому сохраняются ответы пользователя на вопросы; один пользователь может участвовать в любом количестве опросов

- получение пройденных пользователем опросов с детализацией по ответам (что выбрано) по ID уникальному пользователя

Использовать следующие технологии: Django 2.2.10, Django REST framework.

Результат выполнения задачи:

- исходный код приложения в github (только на github, публичный репозиторий)

- инструкция по разворачиванию приложения (в docker или локально)

- документация по API

Выполнение:

Программа написана на Python 3.9, должна работать со всеми врелизами Python 3.x

Допольнительные пакеты:

Django 2.2.10 pip install Django==2.2.10

Django REST framework pip install djangorestframework

Достоинства и недостатки в реализации

Достоинства:

При одинаковом функционале для разных таблиц создаються практически одинаковые классы с одинаковыми функциями. Отличие состоит лишь в используемых дополнительных классах, описываемые модели и их сериализацию. Проблема решена Передачей классов как параметров в родительсктй класс при помощи функции super().

Файл api\_views.py строки 12-67

Недостатки:

1) При пакетном создании анкет данные отдельно записывались в соответствующие таблицы без принципа транзакции. RestFramework скорее всего обладает необходимым функционалом, требуеться более углубленное его изучение.

Запуск программы:

path../proj1/python manage.py runserver

Структура БД



Таблицы:

types - справочник типов вопросов (текст, переключатель, список)

groups - группа вопросов (анкета)

tasks - вопросы

tlist - id типа вопроса (1,2,3)

quests - опросы

work - база ответов, на которые ответили пользователи

-------------------------------------------------------------------------

REST

<http://127.0.0.1:8000/api/>

Просмотр таблиц :

Тип запроса: GET

REST / <имя таблицы>/

Пример: [http HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/"://127.0.0.1:8000/ HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/"api HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/" HYPERLINK "http://127.0.0.1:8000/api/"/](http://127.0.0.1:8000/api/)types/

-------------------------------------------------------------------------

Добавление записей в таблицы:

Тип запроса: POST

REST / <имя таблицы>/

Data:

{

Filed\_1: value\_1,

…

Filed\_n: value\_n

}

Пример: [http HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/"://127.0.0.1:8000/ HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/"api HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/"/ HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/"groups HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/"/](http://127.0.0.1:8000/api/groups/)

Data =

{

“name”:”Анкета 1”

}

-------------------------------------------------------------------------

Корректировка записей:

Тип запроса: POST

REST / <имя таблицы>/<id>

Data:

{

Filed\_1: value\_1,

…

Filed\_n: value\_n

}

Пример: [http HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/"://127.0.0.1:8000/ HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/"api HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/"/ HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/"groups HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/" HYPERLINK "http://127.0.0.1:8000/api/groups/"/](http://127.0.0.1:8000/api/groups/)1

Data =

{

“name”:”Анкета 2”

}

-------------------------------------------------------------------------

Создание анкеты:

Тип запроса: POST

REST /cmd/

Data: - подробная json структура

Пример: <http://127.0.0.1:8000/api/cmd>

Data:

{

"cmd":"create\_form",

"data":

{

"name":"form1",

"tasks":

[

{

"task":"ФИО",

"type":"1",

"list":[]

},

{

"task":"Национальность",

"type":"1",

"list":[]

},

{

"task":"Пол",

"type":"2",

"list":

[

"Мужчина",

"Женщина"

]

},

{

"task":"Увлечения",

"type":"3",

"list":

[

"Спорт",

"Кино",

"Музыка",

"Рисование"

]

}

]

}

}

-------------------------------------------------------------------------

Старт анкетирования для пользователя :

Тип запроса: POST

REST /cmd/

Data: - подробная json структура

Пример: <http://127.0.0.1:8000/api/cmd>

Data:

{

"cmd":"create\_quest",

"data":

{

"name":"quest1",

"group":"39",

"description":"Тест для Иванова И.И."

}

}

или

{

"cmd":"create\_quest",

"data":

{

"name":"quest1",

"group":"39",

"user\_auth":"user\_6765127177"

"description":"Тест для Иванова И.И."

}

}

Если поле "user\_auth" отсутствует, генерируеться новый id пользователя

--------------------------------------------------------

Проссмотр списк вопросов в анкете

Тип запроса: GET

REST /quest\_read/<id>

Пример: [http://127.0.0.1:8000/api/ HYPERLINK "http://127.0.0.1:8000/api/quest\_read/22" HYPERLINK "http://127.0.0.1:8000/api/quest\_read/22" HYPERLINK "http://127.0.0.1:8000/api/quest\_read/22"quest\_read/22](http://127.0.0.1:8000/api/quest_read/22)

--------------------------------------------------------

Проссмотр ответов в анкете

Тип запроса: GET

REST /quest\_view/<id>

Пример: [http://127.0.0.1:8000/api/ HYPERLINK "http://127.0.0.1:8000/api/quest\_view/22" HYPERLINK "http://127.0.0.1:8000/api/quest\_view/22" HYPERLINK "http://127.0.0.1:8000/api/quest\_view/22"quest\_view/22](http://127.0.0.1:8000/api/quest_view/22)

--------------------------------------------------------

Проссмотреть все анкеты определенного пользователя

Тип запроса: GET

REST /quest\_view/?user\_id=<userId>

Пример: [http://127.0.0.1:8000/api/ HYPERLINK "http://127.0.0.1:8000/api/quest\_view/?user\_id=user\_1634880891977" HYPERLINK "http://127.0.0.1:8000/api/quest\_view/?user\_id=user\_1634880891977" HYPERLINK "http://127.0.0.1:8000/api/quest\_view/?user\_id=user\_1634880891977"quest\_view/?user\_id=user\_1634880891977](http://127.0.0.1:8000/api/quest_view/?user_id=user_1634880891977)

===============================================================