## Требования

* Python 3.5 или выше;
* Django 2.2 или выше;
* PostgreSQL или MySQL на выбор.

## DRF Задача #1: Отзывы об автомобилях

Развернуть Django проект, развернуть базу данных, настроить Django Rest Framework.

### Создать модели данных:

1. **Страна**, с полями: «Имя»
2. **Производитель**, с полями: «Имя», «Страна»
3. **Автомобиль**, с полями: «Имя», «Производитель», «Год начала выпуска», «Год окончания выпуска»
4. **Комментарий**, с полями: «Email автора», «Автомобиль», «Дата создания», «Комментарий»

### Создать Endpoint’ы (/api/…) для:

1. Добавления, изменения, удаления и просмотра записей в моделях: «**Страна**», «**Производитель**», «**Автомобиль**», «**Комментарий**».

(GET, POST, PUT, DELETE)

1. Экспорта данных в формате xlsx + csv в зависимости от передаваемого в запросе GET параметра.

### Настроить сериализаторы:

1. При запросе **страны** на стороне сериализатора добавить **производителей** в выдачу, которые ссылаются на нее.
2. При запросе **производителя** добавлять **страну**, **автомобили** и **количество** **комментариев** к ним к выдаче.
3. При запросе **автомобиля** добавить **производителя** и **комментарии** с **их** **количеством** в выдачу.
4. При добавлении комментария проводить валидацию входных данных.

**Предоставить доступ к операциям добавления, редактирования и удаления страны, производителя и автомобиля только через передачу определенного токена. Доступ к добавлению и просмотру комментария оставить публичный, редактирование или удаление через токен.**

### Дополнительно, но не обязательно:

1. К проекту приложить файл с дампом запросов из ПО для тестирования (прим.: Postman)
2. При наличии знаний JS:
   1. Создать и базово оформить веб-страницу для работы с API в корневом url. Отдавать ее используя встроенные средства Django.
   2. Используя Fetch/Axios написать базовый функционал для работы с API на странице.
   3. Динамически обновлять данные на странице при внесении изменений.
3. При наличии знаний Docker:
   1. Контейнеризировать Базу данных и приложение Django, адаптировать проект для развертывания через Docker-Compose.