**Занятие №13**

**Задание №1**

Есть гостевая книга. Похожа на статью (Post).

Нет заголовка, есть только текст, и можно подписаться только именем (любым).

* Должна содержать модель GuestBookItem
* Поле автор
* Текст сообщения может быть любой текст длиннее 5 символов
* Дата и время, когда опубликована
* Должна содержать булевое поле, если захотим удалить конкретную запись
* Должна отдавать все записи по адресу “/items” и методу GET в формате JSON
* Должна быть возможность создавать записи по адресу “/items” и методу POST
* Должна работать валидация при помощи формы модели

**Задание №2(посложнее)**

* Отдавать конкретную запись по адресу “/items/<item\_id>” и методу GET в формате JSON
* На POST запрос из задания 1 надо отдавать код ответа 201 и адрес созданного объекта ( “/items/<item\_id>”) в заголовке Location и JSON в теле ответа, с созданным объектом (как в пункте 1)
* Добавить поле updated\_at в модель GuestBookItem, при создании нового объекта проставлять его равным дате и времени создания
* Редактировать конкретную запись по адресу “/items/<item\_id>” и методу PATCH, если ресурс с таким id не существует - код ответа 404, если ресурс отредактирован - JSON в теле ответа, с созданным объектом (как в пункте 1) и обновленным значение поля updated\_at. Передаются только поля, значение которых надо обновить.
* Заменять конкретную запись по адресу “/items/<item\_id>” и методу PUT, по логике, описанной в пункте 4. Поля передаются все.
* Удалять все записи, при запросе по адресу “/items” и методу PUT c пустым телом запроса, возвращается пустой список объектов.
* Удалять (проставлять поле видимости в False) по адресу “/items/<item\_id>” и методу DELETE, если записи с таким id не существует - код ответа 204

**Задание №3(сложное)**

* Отдавать адресу “/items?page=<page\_num>&per\_page=<items\_count\_in\_page>” по items\_count\_in\_page на страницу.
* Ограничить per\_page так, чтобы его нельзя было сделать слишком большим
* Дополнительно должен отдаваться заголовок “X-Total-Count" с полным количеством элементов, и заголовок Link. Информация в нем должна соответствовать <https://developer.github.com/v3/#link-header>
* GET “/items” должен поддерживать сортировку вида - /items&sort=<имя\_поля\_по\_поторому\_сортируем>, если перед именем поля стоит “-”, то сортировку делаем в обратном порядке
* GET “/items” должен поддерживать фильтрацию вида - /items&<имя\_поля\_по\_которому\_фильтруем>=<значение> и выводить только записи, удовлетворяющие фильтру
* Расширить фильтрацию, теперь перед значением можно добавлять символы “>”, “<”, “>=”, “<=”, означающие, что фильтруем записи, значение полей которые больше, меньше и т.д. чем переданное для фильтра значение
* GET “/items” должен поддерживать список полей, которые отдаются вида - /items&fields=<список полей через запятую>