

Описание.

- Разработать web-приложение на aiohttp.
- Приложение слушает порт 8080.
- Логи должны выводиться в stdout
- Приложение должно работать внутри docker
 - Необходимо составить докер файл на основе fedora:32
 - будет плюсом не использовать pip/requirements.txt, а использовать ванильные пакеты из fedora.
 - Необходимо запустить приложение внутри докер контейнера.
- Даны 2 json файла:
- **news.json**, в котором находится список новостей.

```
{
  "news": [
    {
      "id": 1,
      "title": "news_1",
      "date": "2019-01-01T20:56:35",
      "body": "The news",
      "deleted": false,
    },
  ],
  "news_count": 1,
}
```

- **comments.json**, в котором находятся комментарии к новостям, сопоставление по полю "news_id".

```
{
  "comments": [
    {
      "id": 1,
      "news_id": 1,
      "title": "comment_1",
      "date": "2019-01-02T21:58:25",
      "comment": "Comment",
    },
  ],
  "comments_count": 1,
}
```

Необходимо разработать бекенд, web-приложение следующее API:

- **GET /api** - возвращает список новостей следующего формата.

```
{
  "news": [
    {
      "id": 1,
      "title": "news_1",
      "date": "2019-01-01T20:56:35",
      "body": "The news",
      "deleted": false,
      "comments_count": 1,
    },
  ],
  "news_count": 1,
}
```

Все записи должны быть отсортированы по дате создания (поле "date" в формате ISO 8601).
возвращать необходимо не удаленные записи (поле "deleted").
так же не нужно возвращать записи, время которых еще не наступило.
необходимо возвращать количество комментариев к каждой новости (поле "comments_count").

- **GET /api/news/{id}** - возвращает новость по ее id.

```
{
  "id": 1,
  "title": "news_1",
  "date": "2019-01-01T20:56:35",
  "body": "The news",
  "deleted": false,
  "comments": [
    {
      "id": 1,
      "news_id": 1,
      "title": "comment_1",
      "date": "2019-01-02T21:58:25",
      "comment": "Comment",
    },
  ],
  "comments_count": 1,
}
```

- **POST /api/news** - создать новость по ее id.

```
{
  "id": 1,
  "title": "news_1",
  "body": "The news",
  "comments": [
    {
      "id": 1,
      "news_id": 1,
      "title": "comment_1",
      "comment": "Comment",
    },
  ],
}
```

- **DELETE /api/news/{id}** - удалить новость.

Все комментарии должны быть отсортированы по дате создания (поле "date" в формате ISO 8601).
необходимо вернуть количество комментариев к текущей новости (поле "comments_count").
в случае, если новости с таким id нет, необходимо вернуть код ошибки - 404.
в случае, если запись удалена (поле "deleted"), необходимо вернуть код ошибки - 404.
в случае, если время создания записи еще не наступило, необходимо вернуть код ошибки - 404.
Все записи сохранить в json.

Решение необходимо предоставить в виде git репозитория.
Язык реализации Python3.5 или старше.
Фреймворк для реализации aiohttp.
Плюсом будет наличие тестов.