

Тестовое задание для Python-разработчика

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, реализующий следующее API:

GET "/" - 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 "/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,
},
```

- 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.

Решение необходимо предоставить в виде git репозитория.

Язык реализации Python3.5 или старше.

Фреймворк для реализации aiohttp.

Плюсом будет наличие тестов.