

## Тестовое задание для соискателя в отдел Web разработки компании Frumatic

Необходимо написать простой клиент для сервиса Storia со следующими возможностями:

### *Обязательные:*

1. Отображения списка моментов ленты пользователя (Достаточно отображать имя автора, название истории, заголовок момента, если он есть, и одну выборочную картинку, если картинки есть).
2. Выборочная картинка из момента должна соответствовать следующему критерию: это должна быть либо первая картинка с конца не имеющая цифр в поле file.title, либо последняя картинка, если таких картинок нет.
3. Отображение количества лайков момента.
4. Нужна возможность лайкнуть момент.
5. Если нет соединения с интернетом, то необходимо отображать моменты из кеша, даже если страница была перезагружена.
6. Реализовать отдельное view со своим роутом для детального отображения момента. Возможность лайкать момент должна быть сохранена.
7. Организуйте фильтрацию “Моменты с картинками”/”Моменты без картинок” по кнопке

### *Дополнительные возможности:*

8. Если лайкнуть момент при отсутствующем соединении с интернетом, то информация о лайке должна быть передана на сервер после того как соединение появится, даже если страница была перезагружена.
9. Реализуйте загрузку последующих страниц ленты пользователя по кнопке Load More.
10. Реализуйте server render.

### *Условия выполнения:*

1. Данное задание должно быть написано на react.js
2. Для user interface используйте библиотеку Bootstrap (желательно), либо любую иную.
3. Реализация кеша на Ваше усмотрение.
4. Наличие/реализация сервера на Ваше усмотрение.
5. Комментируйте ключевые моменты в приложении.
6. Если у Вас осталось время и желание, реализуйте дополнительные 8-10 пункты

### *Дополнительная информация:*

Для выполнения задания необходимо использовать пользователя со следующими данными:

email: test\_task@example.com  
password: qwe123

## Описание API Storia

### Авторизация

POST [https://storia.me/api/acl/auth/Selfish/test\\_task@example.com](https://storia.me/api/acl/auth/Selfish/test_task@example.com)

тело запроса:

```
{  
  "password": "qwe123",  
  "remember": false,  
  "token": ""  
}
```

### Получения списка моментов:

GET <https://storia.me/api/feed/content>

Описание ответа:

items - массив моментов ([описание структуры момента ниже](#))

maxId - id самого молодого момента в возвращённом массиве

minId - id самого старого момента в возвращённом массиве

moreAvailable - возможно ли запросить ещё моменты

unread - количество непрочитанных моментов

По умолчанию запрос возвращает 20 элементов. Подгрузку следующих элементов можно организовать используя параметры GET запроса:

limit - указывается максимальное количество возвращаемых моментов

until - id момента, старше которого нам нужно запросить список

### Получение момента:

GET <https://storia.me/api/core/stories/087ef7a4a9470000/moments/0911e5f2b9870000>

где [087ef7a4a9470000](#) - id истории

где [0911e5f2b9870000](#) - id момента

Описание ответа(только нужные поля):

- index - индекс момента в истории
- moment - информация о моменте, соответствует [возвращаемой в ленте пользователя](#).

### **Поставить/удалить лайк:**

POST <https://storia.me/api/core/stories/{storyId}/moments/{momentId}/like>

Удалить лайк момента:

DELETE <https://storia.me/api/core/stories/{storyId}/moments/{momentId}/like>

{storyId} – id истории

{momentId} – id момента

### **Структура момента:**

objectPreview - json объект, краткая информация о момента. В данной задаче нас интересуют поля:

- id – id момента
- title - заголовок момента
- storyId – id истории
- storyTitle – название истории
- attachments - массив объектов, содержащих информацию о картинках. Может быть пуст. Нас интересует поле file, которое является объектом json, а в нём поле path, в котором содержится информация о пути до картинки.
- owner - json объект, информация об авторе момента. В нём есть поле name, которое нас интересует
- stats – json объект, информация о разных числовых характеристиках момента, нас интересует поле likes - информация о количестве лайков момента