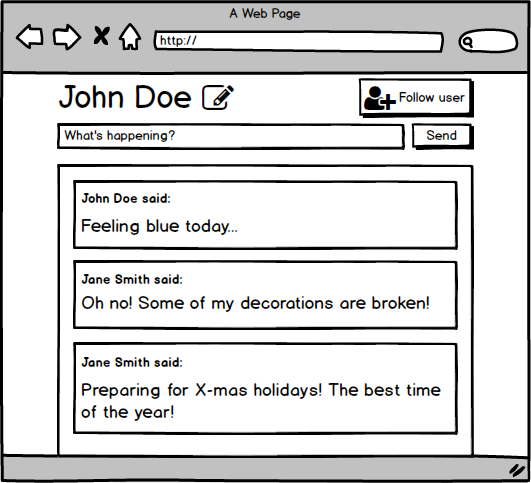
# Контрольная работа 6

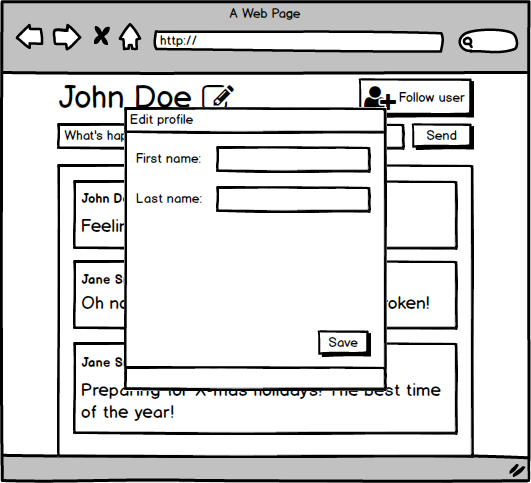
Необходимо создать приложение - аналог ленты в социальной сети, которое будет работать со специально созданным API. Описание API находится далее по тексту.

Общий вид такого приложения:



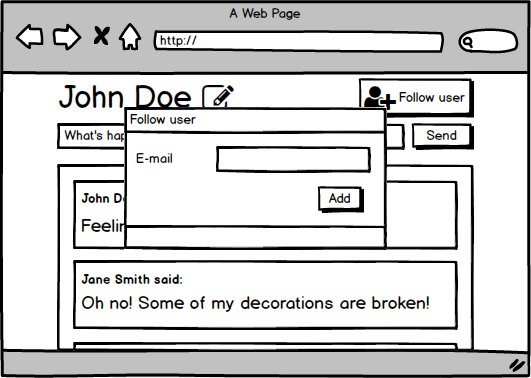
На данной странице должны отображаться ваши посты и посты других пользователей, на которых вы подписаны, в порядке убывания даты (то есть, самые недавние посты должны располагаться сверху)

Кнопка "Редактировать" открывает окно редактирования профиля, в него вы вводите имя и фамилию.



При нажатии на "Save" информация на странице должна обновиться.

Нажатие на кнопку "Follow user" открывает окно ввода e-mail, на которого вы хотите подписаться.



После того, как вы подписались на пользователя, его сообщения должны также отображаться, одновременно с вашими.

Вы можете меняться адресами e-mail с вашими коллегами, и подписываться друг на друга, или создать несколько пользователей.

## Описание API

Базовый URL: <http://146.185.154.90:8000/blog>/{e-mail}

Для того, чтобы совершать запросы на API вам потребуется свой адрес электронной почты. Он будет служить вашим уникальным идентификатором. Для этого вы добавляете к базовому URL свой e-mail, например, таким образом:

<http://146.185.154.90:8000/blog>/john.doe@gmail.com/

Остальные адреса, указанные ниже будут **добавляться к этому URL.**

### Профиль

Чтобы начать работу, вам нужно отправить GET-запрос на адрес:

* GET /profile

При этом в базе данных создастся ваш профиль, со стандартным именем John Doe.

Только после этого запроса можно будет делать запросы на все остальные API. Если сделать запрос на этот адрес в следующий раз, будет приходить информация о вашем профиле. Если вы измените имя и фамилию (см. дальше), будет приходить уже ваша новая (обновленная) информация.

Чтобы изменить свои данные:

* POST /profile
* Отправляете данные:
* firstName: текст, не должен быть пустым
* lastName: текст, не должен быть пустым

### Лента

Чтобы запросить "ленту" (последние 20 постов):

* GET /posts

Чтобы запросить "ленту" с определенной даты:

* GET /posts?datetime={datetime}
* где {datetime} - дата в формате ISO-строки (которую вы получаете от API)

Чтобы отправить новый пост:

* POST /posts
* Отправляете данные:
* message: текст, не должен быть пустым

### Подписка

Чтобы подписаться на нового пользователя:

* POST /subscribe
* Отправляете данные:
* email: текст, должен присутствовать в базе

Также вы можете отправлять такой POST-запрос без данных, чтобы очистить все подписки (например, для тестирования):

* POST /subscribe/delete

Чтобы посмотреть ваших подписчиков (например, чтобы проверить, добавился ли пользователь):

* GET /subscribe

Вам будет возвращен список пользователей, на которые вы подписаны, в виде массива.

## Указания

* Приблизительный алгоритм работы приложения:
* Основная программа:
* Запросить ваш профиль. Сохранить полученные данные в переменную.
* Отобразить ваше имя на странице
* Запросить последние посты.
* Добавить последние посты в список
* Создать интервал, который будет запрашивать новые посты.
* При клике на кнопку "Редактировать профиль" отобразить окно (попап) с вводом данных (имени и фамилии).
* При клике на кнопку "Save" отправить данные на сервер
* При успешном ответе от сервера обновить данные в переменной и отображаемые данные
* Закрыть окно
* При вводе текста в поле создания нового сообщения и клике на "Send"
* Отправить данные нового сообщения на сервер
* При клике на кнопку "Добавить подписчика" отобразить окно (попап) с вводом e-mail
* При клике на кнопку "Add" отправить данные на сервер
* При успешном ответе от сервера закрыть окно
* Необходимо пользоваться Promise-ами везде, где можно. Необходимо пользоваться конструкцией async/await везде, где применимо.
* Не делайте слишком быстрый интервал! Вы можете все одновременными запросами перегрузить сервер. Установите его на как минимум 2 секунды.
* Можно пользоваться любыми библиотеками, плагинами, и т.п. какие вы найдете (например, для popup-окон).
* Можете оформить ваше приложение с помощью Bootstrap (по желанию).
* Для тестирования запросов можно пользоваться Postman или аналогичными инструментами.