

Реализовать рабочий календарь

Необходимо реализовать реактивный компонент на Angular, который будет отображать рабочее время по сотрудникам.

Пример макета календаря приведен ниже:

Рабочий календарь + ДОБАВИТЬ ЗАПИСЬ

Тип записи: не выбрано Сотрудник: не выбрано

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30								
Алена Анцибор										отгул (10.1)																												
Оксана Башкирова		отпуск (02.11.2020 - 15.11.2020)																																				
Дарья Верченко																отпуск (15.11.2020 - 29.11.2020)																						
Юлия Злоян																							отгул (24.1)															
Олег Касаткин										отпуск (09.11.2020 - 15.11.2020)																												
Наталья Подгайская		отпуск (19)																																				
Екатерина Ткаченко					отгул (05.11.2020 - 06)																						отгул (27.1)											

1. Требования к функционалу календаря:

1.1 По умолчанию при открытии html страницы график должен отобразить данные за текущий месяц и год по всем типам записей и всем сотрудникам;

1.2 Календарь должен иметь фильтр по типу записи и по пользователю в виде селектов:

Тип записи: не выбрано Сотрудник: не выбрано

Селекты “тип записи” и “сотрудник” должны иметь тип множественного выбора значений (multiselect).

1.3 Календарь должен иметь возможность просмотра данных за любой месяц.

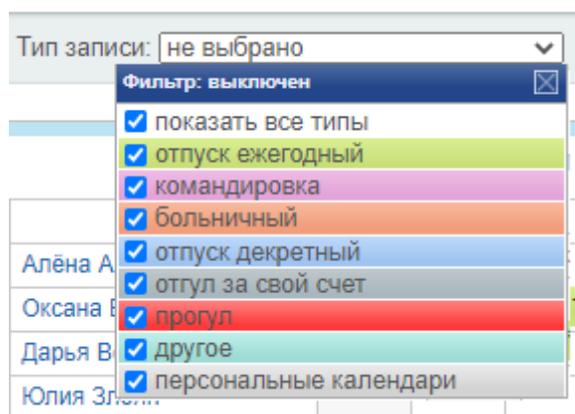
Изменение месяца происходит по клику на стрелочки влево и вправо:



При изменении месяца, необходимо подгружать новые данные с помощью аякс только за выбранный месяц и год без перезагрузки всей страницы целиком. В процессе загрузки данных показывать прелодер в центре таблицы:



1.4 Цвет рабочей линии необходимо отображать в зависимости от типа записи, например “рабочее время” - отобразить все таймлайн линии синим цветом, “отпуск” - отобразить красным, “отгул” - отобразить зеленым и т.д.;



1.5 В таблице фон ячеек с выходными днями (суббота и воскресенье) необходимо отобразить серым цветом #f7f7f7.

1.6 В таблице фон ячеек с текущим днем необходимо отобразить цветом #E7EEF1.

background: #f7f7f7;

20	21	22	23	24	25	26	27

2. Требования к верстке календаря:

2.1 Верстка должна быть адаптивной, таблица должна растягиваться на всю ширину контейнера в зависимости от разрешения экрана. Таблица календаря должна сжиматься и растягиваться при изменении размера браузера.

2.2 Календарь должен одинаково работать в современных браузерах Chrome, Firefox, Safari, Yandex Браузер и т.п.

2.3 Отображение таймлайн-линии в ячейках таблицы должно быть пропорционально указанному времени. Например, если была указана дата 1 ноября 2020 г и время с 18:00 по 08:00 то линия должна отображаться так:

	01	02	03	04
Алёна Анцибор				
Оксана Башкирова				
Дарья Верченко				
Юлия Злоян				
Олег Касаткин				
Наталья Подгайская				
Екатерина Ткаченко				

Если было указана дата 3 ноября 2020 г и время с 08:00 по 18:00 то линия должна отображаться так:

	01	02	03	04
Алёна Анцибор				
Оксана Башкирова				
Дарья Верченко				
Юлия Злоян				
Олег Касаткин				
Наталья Подгайская				
Екатерина Ткаченко				

3. Описание функционала:

3.1 Для получения данных календаря необходимо выполнить **GET**-запрос на адрес <https://dev-api.gisgkx.ru/calendar>

Для получения данных в JSON формате, необходимо в заголовке запроса указать Content-Type: application/json.

По умолчанию выводятся данные только за текущий месяц и год.

Для фильтрации данных (это необходимо для реализации п. 1.2 и 1.3) необходимо в url передать дополнительные параметры, например:

[https://dev-api.gisgkx.ru/calendar?date=2020-11&user_id\[\]=1&user_id\[\]=2&type_id\[\]=1&type_id\[\]=2](https://dev-api.gisgkx.ru/calendar?date=2020-11&user_id[]=1&user_id[]=2&type_id[]=1&type_id[]=2)

, где

- date - дата в формате год-месяц (2020-12),
- user_id - уникальный идентификатор пользователя (тип - массив идентификаторов),
- type_id - уникальный идентификатор типа записи (тип - массив идентификаторов).

Запрос вернет JSON следующего формата:

```
{
  "success": 1,
  "data": [
    {
      "id":1,
      "user_id":1,
      "type_id":2,
      "description":"смена 1",
      "start":"2019-12-01 08:00",
      "end":"2019-12-01 18:00"
    },
    {
      "id":2,
      "user_id":1,
      "type_id":2,
      "description":"смена 1",
      "start":"2019-12-04 08:00",
      "end":"2019-12-04 18:00"
    },
    {
      "id":3,
      "user_id":2,
      "type_id":2,
      "description":"смена 2",
      "start":"2019-12-30 08:00",
      "end":"2019-12-30 18:00"
    },
    {
      "id":4,
      "user_id":2,
      "type_id":2,
      "description":"смена 2",
      "start":"2019-12-02 08:00",
      "end":"2019-12-02 18:00"
    }
  ]
}
```

Где:

id	уникальный идентификатор записи таймлайна
user_id	уникальный идентификатор сотрудника
type_id	уникальный идентификатор типа записи
description	Комментарий
start	Начальная дата и время
end	Конечная дата и время

3.2 Для получения списка сотрудников необходимо выполнить GET-запрос на адрес

<https://dev-api.gisgkx.ru/calendar/users>

Запрос вернет JSON следующего формата:

```
[
  {
    "id":1,
    "name":"Сергей Алексеев",
  },
]
```

```
{
  "id":2,
  "name":"Сергей Алексеев",
},
{
  "id":3,
  "name":"Оксана Башкирова",
},
{
  "id":4,
  "name":"Оксана Башкирова",
}
]
```

3.3 Для получения списка типов записей необходимо выполнить GET-запрос

<https://dev-api.gisgkx.ru/calendar/types>

Запрос вернет JSON следующего формата:

```
[
  {
    "id":1,
    "name":"отпуск ежегодный",
    "color":"#cef233",
  },
  {
    "id":2,
    "name":"рабочий день",
    "color":"#ee4",
  },
  {
    "id":3,
    "name":"отпуск",
    "color":"#aaa",
  },
  {
    "id":4,
    "name":"прогул",
    "color":"#345223",
  }
]
```

Все списки должны динамически подгружаться при клике на соответствующий селект.

4. Описание действий в календаре:

4.1 При нажатии на кнопку “Добавить запись” должно появляться модальное окно со следующими элементами:

- Сотрудник, тип - селект, поле обязательное. Селект должен иметь контекстный поиск по элементам;
- Тип записи, тип - селект, поле обязательное;
- Комментарий, тип - инпут, поле необязательное;
- Период, тип - календарь и инпуты, поле обязательное;
- Ссылка “Добавить период”
- Кнопка “Добавить”
- Ссылка “Закрыть”

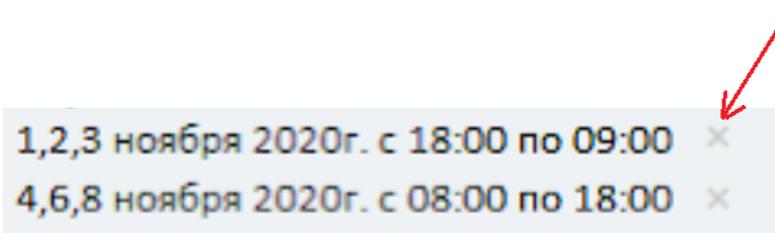
Сотрудник:

4.1.1 При нажатии на ссылку “Добавить период”, должен появиться календарь с множественным выбором нужных дней, а также с указанием рабочего времени:

4.1.2 После выбора нужных дней и времени в календаре и нажатия на ссылку “выбрать”, календарь нужно закрыть, а в модальном окне отобразить соответствующую строку, например:

1,2,3 ноября 2020г. с 18:00 по 08:00

Рядом со строчкой необходимо расположить иконку в виде крестика, при клике на которую, запись можно удалить.



4.1.3 После нажатия на зеленую кнопку “Добавить” необходимо отправить данные на сервер методом **POST** на адрес: <https://dev-api.gisgkx.ru/calendar>

С телом запроса в формате JSON:

```
[
  {
    "user_id": 1,
    "type_id": 1,
    "description": "смена 1",
    "start": "2020-12-01 08:00",
```

```
"end": "2020-12-01 18:00"
},
{
  "user_id": 1,
  "type_id": 1,
  "description": "смена 1",
  "start": "2020-12-04 08:00",
  "end": "2020-12-04 18:00"
},
{
  "user_id": 2,
  "type_id": 1,
  "description": "смена 2",
  "start": "2020-12-30 08:00",
  "end": "2020-12-30 18:00"
},
{
  "user_id": 2,
  "type_id": 1,
  "description": "смена 2",
  "start": "2020-12-02 08:00",
  "end": "2020-12-02 18:00"
}
]
```

При успешном добавлении данных, сервер вернет 200 статус, в остальных случаях необходимо вывести модальное окно с ошибкой:

Заголовок: ошибка.

Тело сообщения: неизвестная ошибка попробуйте повторить запрос.

4.1.4 После нажатия на кнопку “Закреть” модальное окно должно закрыться без сохранения введенных данных.

4.2 При клике на таймлайн-линию должно открыться модальное окно с информацией о записи:

Информация о записи

ФИО Сотрудника:
[Алена Анцибор](#)

Тип записи:
[Рабочий день](#)

Комментарий:
[Смена 1](#)

Период:
[с 01 ноября 18:00 по 02 ноября 08:00](#)

Продолжительность:
[14 часов 00 минут](#)

[Закреть](#) [Удалить запись](#)

Атрибут “Продолжительность” вычисляется как разность между конечной и начальной датой.

4.2.1 При клике на ссылку “Удалить запись” необходимо отправить запрос методом **DELETE** по ссылке: <https://dev-api.gisgkx.ru/calendar/234>, где 234 - уникальный идентификатор записи таймлайна.

При успешном удалении данных, сервер вернет 200 статус, в остальных случаях необходимо вывести модальное окно:

Заголовок: ошибка.

Тело сообщения: неизвестная ошибка попробуйте повторить запрос.

После успешного запроса необходимо закрыть модальное окно и скрыть удаленный таймлайн в таблице.

4.2.2 При клике на ссылку “Закреть” - модальное окно необходимо закрыть без изменения данных.