*Взаимодействие сервера и БД*

**GET** /video

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Объект JSON** | **Атрибут JSON** | **Название**  **таблицы в БД** | **Атрибут в**  **таблице БД** | **Комментарий** |
| video\_reports | vrep\_id | video\_reports | vrep\_id |  |
| vrep\_description | vrep\_description |  |
| vrep\_link | vrep\_link |  |
| vrep\_created\_at | vrep\_created\_at |  |
| pagination | current\_page | - | - | = значению параметра “page” в запросе;  диапазон значений: от 1 до 999 |
| per\_page | - | - | = значению параметра “per\_page” в запросе;  диапазон значений: от 1 до 100 |
| total | - | - | Общее кол-во записей в таблице video\_reports |
| last\_page | - | - | Номер по  Вычисляется как total/per\_page, округленное к большему |

1. Атрибуты объекта pagination необходимо возвращать для постраничной выдачи списков видео на фронте.
2. В запросе к БД необходимо:

* сделать сортировку по дате создания от новому к старому;
* ограничить выдачу количеством записей, равным значению параметра per\_page в запросе;
* пропускать количество записей, равное (page - 1)\*per\_page

Например, при параметрах запроса ?page=1&per\_page=10, запрос в БД будет выглядеть примерно так:

SELECT vrep\_id, vrep\_description, vrep\_link, vrep\_created\_at FROM video\_reports ORDER BY vrep\_created\_at DESC LIMIT 10 OFFSET 0

**GET** /families

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Объект JSON** | **Атрибут JSON** | **Название**  **таблицы в БД** | **Атрибут в**  **таблице БД** | **Комментарий** |
| families | f\_id | family | f\_id |  |
| f\_name | f\_name |  |
| fph\_path | family\_photos | fph\_path |  |
| f\_short\_desc | family | f\_short\_desc |  |
| f\_desc | f\_desc |  |
| f\_sum | f\_sum |  |
| f\_current\_sum | f\_current\_sum |  |
| f\_created\_at | f\_created\_at |  |
| pagination | current\_page | - | - | = значению параметра “page” в запросе;  диапазон значений: от 1 до 999 |
| per\_page | - | - | = значению параметра “per\_page” в запросе;  диапазон значений: от 1 до 100 |
| total | - | - | Общее кол-во записей в таблице family |
| last\_page | - | - | Номер по  Вычисляется как total/per\_page, округленное к большему |

1. Атрибуты объекта pagination необходимо возвращать для постраничной выдачи списков видео на главной странице.
2. В запросе к БД необходимо:

* сделать сортировку по дате создания от новому к старому (f\_created\_at);

если в запросе переданы параметры page и per\_page:

* ограничить выдачу количеством записей, равным значению параметра per\_page в запросе;
* пропускать количество записей, равное (page - 1)\*per\_page

(аналогично **GET** /api/video)

1. Значение для fph\_path нужно брать из сджойненных таблиц family и family\_photos (по f\_id) и условием на то, что fph\_main is true. Это нужно для загрузки только основной фотографии на карточку семьи на фронте

**GET** /main

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Объект JSON** | **Атрибут JSON** | **Название**  **таблицы в БД** | **Атрибут в**  **таблице БД** | **Комментарий** |
| donations | don\_id | donations | don\_id |  |
| don\_sum | don\_sum |  |
| don\_name | don\_name |  |
| don\_created\_at | don\_created\_at |  |
| blog | article\_id | blog | article\_id |  |
| article\_name | article\_name |  |
| article\_publish\_date | article\_publish\_date |  |
| photo\_reports | phrep\_id | photo\_reports | phrep\_id |  |
| phrep\_description | phrep\_description |  |
| phrep\_created\_at | phrep\_created\_at |  |
| photos | ph\_id | photos\_in\_report | ph\_id |  |
| phrep\_path | phrep\_path |

1. Из таблицы «donations» нужно всегда запрашивать топ-10 записей, отсортированных по атрибуту don\_created\_at по убыванию (от свежего к старому)
2. Из таблицы «blog» нужно всегда запрашивать топ-6 записей, отсортированных по атрибуту article\_publish\_date по убыванию (от свежего к старому)
3. Атрибуты из таблицы photos\_in\_report нужно брать из таблицы, сджойненной с photo\_reports по phrep\_id (чтобы в одном item массива photos в ответе API были фото, которые привязаны к конкретному фотоотчету)

**GET** /documents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Объект JSON** | **Атрибут JSON** | **Название**  **таблицы в БД** | **Атрибут в**  **таблице БД** | **Комментарий** |
| documents | doc\_id | foundation\_documents | doc\_id |  |
| doc\_name | doc\_name |  |
| doc\_path | doc\_path |  |

Список документов предоставит заказчик их, нужно будет вручную положить в хранилище и ссылку в БД

**GET** /banking

Название объекта и атрибутов JSON полностью совпадают с названием таблицы и атрибутов в БД.

Документ и реквизиты предоставит заказчик, их нужно будет вручную положить в хранилище и в БД

**GET** /families/{f\_id}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Объект JSON** | **Атрибут JSON** | **Название**  **таблицы в БД** | **Атрибут в**  **таблице БД** | **Комментарий** |
| family | f\_id | family | f\_id |  |
| f\_name | f\_name |  |
| fph\_main\_path | family\_photos | fph\_ path |  |
| fph\_id | fph\_id |  |
| fph\_path | fph\_path |  |
| f\_short\_desc | family | f\_short\_desc |  |
| f\_desc | f\_desc |  |
| f\_sum | f\_sum |  |
| f\_current\_sum | f\_current\_sum |  |
| f\_history | f\_history |  |
| fcomm\_id | family\_comments | fcomm\_id |  |
| fcomm\_name | fcomm\_name |  |
| fcomm\_publish\_date | fcomm\_publish\_date |  |
| fcomm\_text | fcomm\_text |  |
| fcomm\_likes | fcomm\_likes |  |
| other\_families | other\_f\_id | family | f\_id |  |
| other\_f\_name | f\_name |  |
| other\_fph\_path | family\_photos | fph\_path |  |
| other\_f\_created\_at | family | f\_created\_at |  |

В URL запроса передается параметр f\_id – ID семьи, для которой нужно получить данные:

* для объекта family нужно извлекать из БД данные по переданному значению f\_id;
* для объекта other\_families нужно извлекать из БД топ-10 записей, упорядоченных по убыванию f\_created\_at (от самых новых к самым старым), с условием на то, что их f\_id ≠ значению в параметре запроса (т.к. для этого блока нужно получить данные о других семьях)

Значение для fph\_main\_path и other\_fph\_path нужно брать из сджойненных таблиц family и family\_photos (по f\_id) и условием на то, что fph\_main is true. Это нужно для того, чтобы передать в этих атрибутах основную фотографию семьи.

**POST** /families/comment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Объект JSON** | **Атрибут JSON** | **Название**  **таблицы в БД** | **Атрибут в**  **таблице БД** | **Комментарий** |
| AddFamilyComment | f\_id | family\_comments | f\_id |  |
| name | fcomm\_name |  |
|  | email | fcomm\_email |  |
|  | text | fcomm\_text |  |
| CommentAdded | id | fcomm\_id |  |
|  | f\_id | f\_id |  |
|  | name | fcomm\_name |  |
|  | email | fcomm\_email |  |
|  | text | fcomm\_text |  |
|  | created\_at | fcomm\_publish\_date |  |

**POST** /families/comment/like

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Объект JSON** | **Атрибут JSON** | **Название**  **таблицы в БД** | **Атрибут в**  **таблице БД** | **Комментарий** |
| LikeFamilyComment | id | family\_comments | fcomm\_id |  |
| LikeAdded | id | fcomm\_id |  |
|  | likes\_count | fcomm\_likes |  |

При получении этого запроса в таблице family\_comments в запись с указанным в теле запроса id комментария необходимо увеличить значение поля fcomm\_likes на 1.

**POST** /families/personalhelp\_request

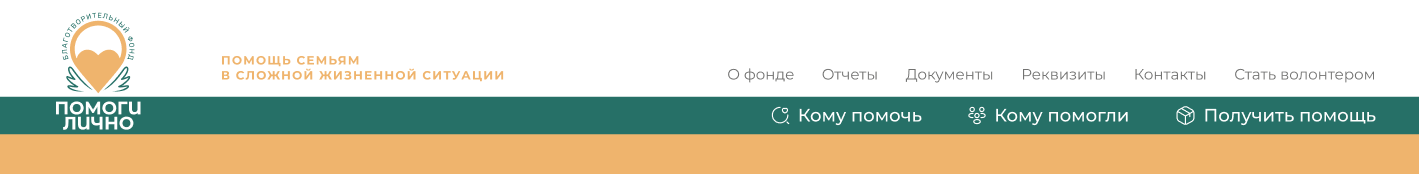
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Объект JSON** | **Атрибут JSON** | **Название таблицы в БД** | **Атрибут в таблице БД** | **Комментарий** |
| PersonalHelp | surname | pers\_help\_request | phreq\_surname |  |
| name | phreq\_name |  |
| patname | phreq\_patname |  |
| phone | phreq\_phone |  |
| email | phreq\_email |  |
| city | phreq\_city |  |
| fname | phreq\_fname |  |
| description | phreq\_description |  |

*Соответствие атрибутов API элементам интерфейса*

[Ссылка](https://www.figma.com/design/MJAVhvXe3Y6zpNz3gYAVnl/sait_pomogi_lichno_-(Copy)?node-id=0-1&t=SYzIRm9YzP9DVKUd-1) на макеты от заказчика

Общее для всех страниц сайта

**Блок с верхней панелью (шапкой)**, одинаковый для всех страниц сайта



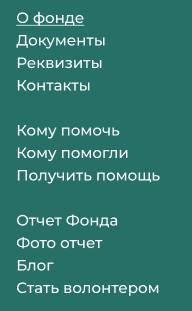
Статические элементы, нужно захардкодить на фронте:

* Логотип «помоги лично», нажатие на логотип с любой страницы сайта должно возвращать пользователя на главную
* Текст «Помощь семьям в сложной жизненной ситуации»
* Кнопки «О фонде», «Отчеты», «Документы», «Реквизиты», «Контакты», «Стать волонтером», «Кому помогли», «Получить помощь»; нажатие на каждую кнопку перенаправляет пользователя на соответствующую отдельную страницу сайта. Кнопка «Кому помочь» не нужна. Справа от «Контакты» нужно добавить кнопку «Блог» для перехода на страницу блога

**Нижний блок**, одинаковый для всех страниц сайта

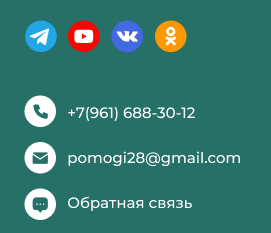


Все элементы блока статичны и должны быть захардкожены на фронте.

1. 

Нажатие на эти кнопки должно перенаправлять пользователя на соответствующие страницы. Коррективы относительно макета:

* Не нужна кнопка «кому помочь»
* Вместо кнопки «Отчет Фонда» -- «Отчеты»
* Не нужна кнопка «Фото отчет»

1. 

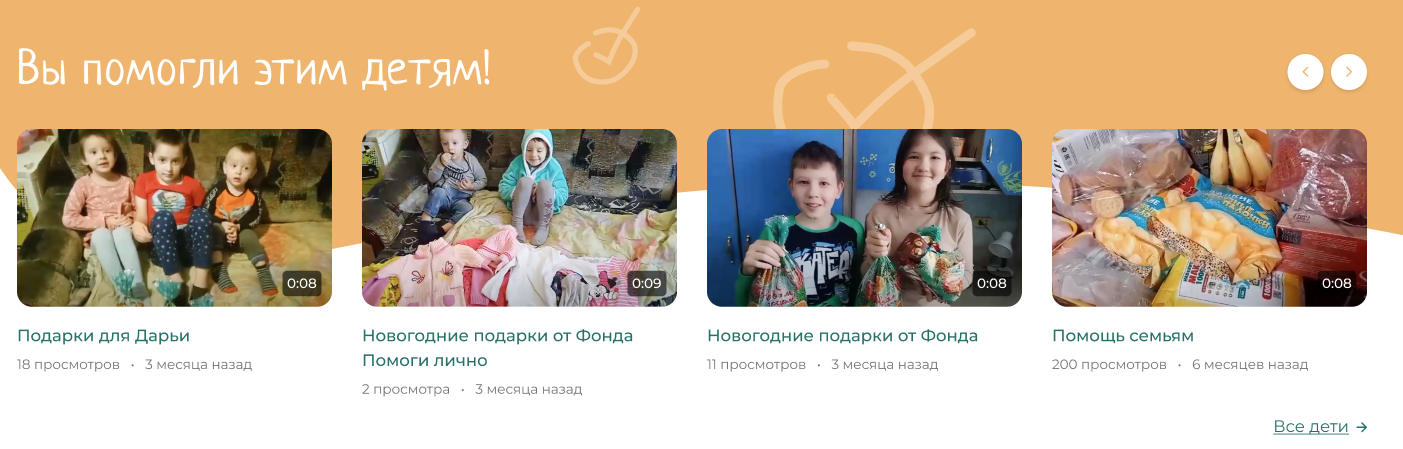
Нажатие на иконки соцсетей должно перенаправлять пользователя на соцсети фонда. Список ссылок будет получен от заказчика.

Коррективы относительно макета:

* Не нужна кнопка и иконка «Обратная связь»

Главная страница сайта

**Блок со списком видео**

****

Статические элементы, нужно захардкодить на фронте:

* Текст заголовка «Вы помогли этим детям!»
* Кнопки < > для пагинации
* Кнопка «Все дети», перенаправляет пользователя на страницу «Кому помогли»

**GET** /video – метод для получения списка видео для блока

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Видео | vrep\_link | Ссылка на видео в ВК, которое необходимо подгрузить на фронт (например, с помощью iframe) |
| Подпись под видео | vrep\_description |  |
|  | vrep\_created\_at | Атрибут, по которому нужно осуществлять сортировку элементов по убыванию от самого нового видео к самому старому |

Необходима пагинация, в методе нужно передавать обязательные параметры запроса:

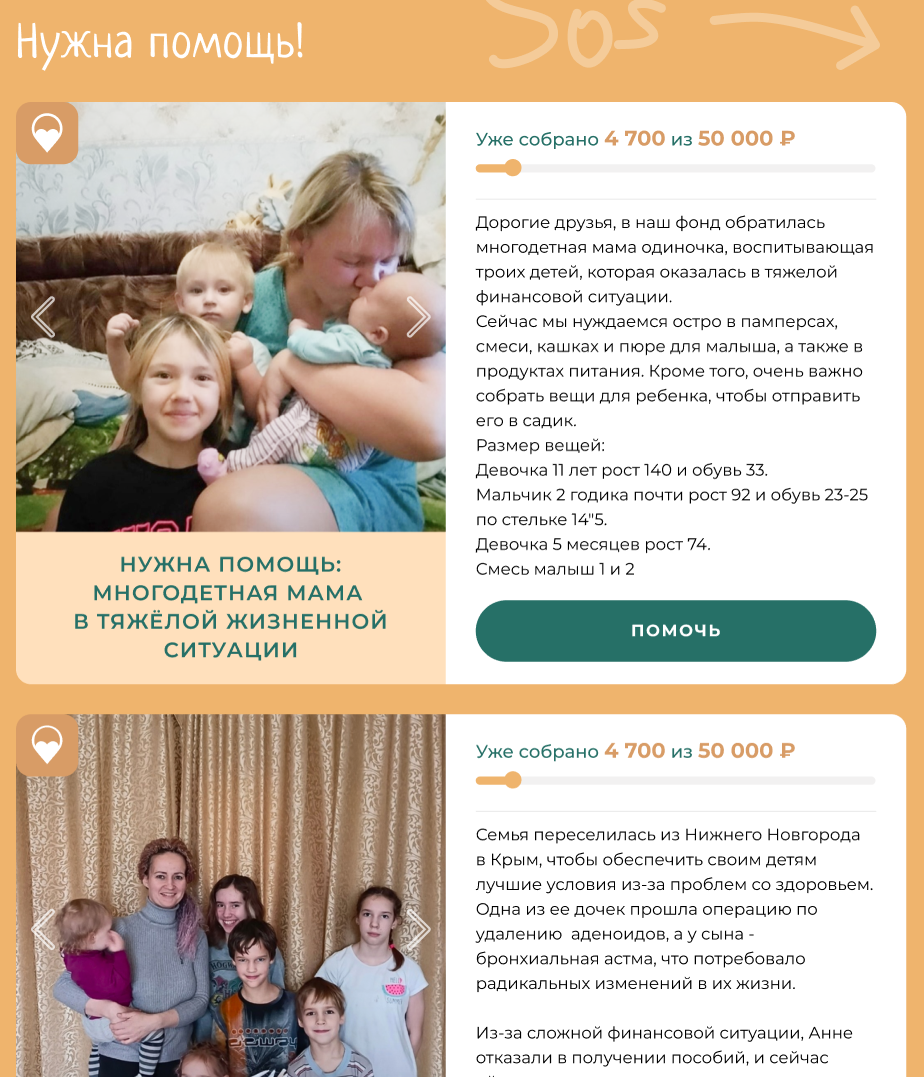
* page (номер страницы), при первом открытии главной страницы =1
* per\_page (количество элементов на странице), всегда должен быть =4

При нажатии на кнопку > отправляется запрос данных для следующей страницы с параметром page=2 для отображения следующей порции видео.

В ответе в объекте pagination будет возвращаться:

|  |  |
| --- | --- |
| **Атрибут** | **Комментарий** |
| current\_page | Текущая страница. Валидация: если current\_page =1, то должна быть заблокирована кнопка < для переключения на предыдущую страницу |
| per\_page | Количество элементов на странице, здесь всегда =4 |
| total | Общее количество элементов в БД |
| last\_page | Номер последней страницы при текущем total в таблице и заданном параметре per\_page. Валидация: если current\_page = last\_page, то должна быть заблокирована кнопка > для переключения на следующую страницу |

**Блок со списком семей**

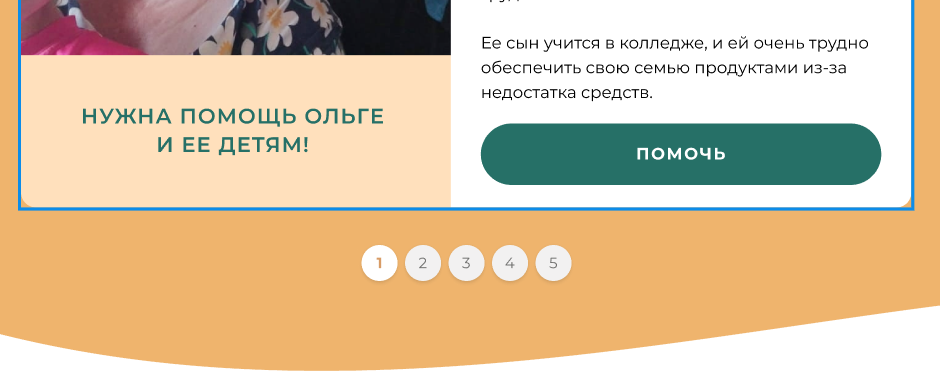


Статические элементы, нужно захардкодить на фронте:

* Текст заголовка «Нужна помощь!»
* Текст «Уже собрано», «из», «Р»
* Кнопка «Помочь», перенаправляет пользователя на страницу подопечного

**GET** /families – метод для получения данных списка семей

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Фото | fph\_path | Путь на фото в файловом хранилище |
| Краткое описание под фото | f\_short\_desc |  |
| Число после «Уже собрано» | f\_current\_sum |  |
| Число после «из» | f\_sum |  |
| Текст описания справа от фото | f\_desc |  |
|  | f\_created\_at | Атрибут, по которому нужно осуществлять сортировку карточек семьи по убыванию от самой новой к самой старой |



Необходима пагинация, в методе нужно передавать обязательные параметры запроса:

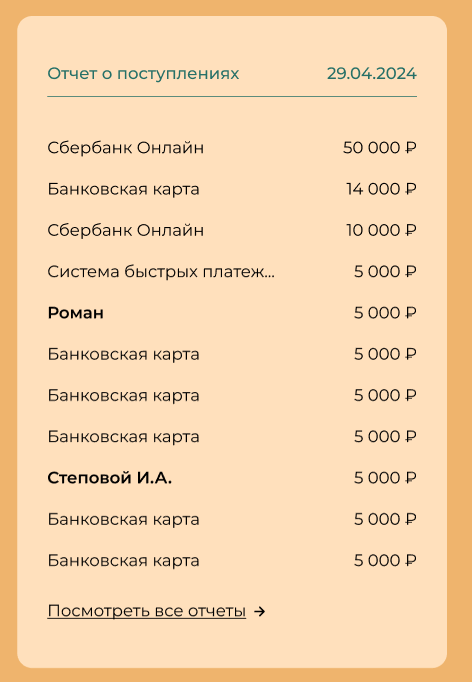
* page (номер страницы), при первом открытии главной страницы =1
* per\_page (количество элементов на странице), всегда должен быть =5

При нажатии на кнопки (1), (2) итд отправляется запрос на страницу с параметром page равным номеру нажатой кнопки для отображения порции карточек.

В ответе в объекте pagination будет возвращаться:

|  |  |
| --- | --- |
| **Атрибут** | **Комментарий** |
| current\_page | Текущая страница |
| per\_page | Количество элементов на странице, здесь всегда =5 |
| total | Общее количество элементов в БД |
| last\_page | Номер последней страницы при текущем total в таблице и заданном параметре per\_page. Количество кнопок для переключения страниц определяется этим атрибутом |

**Блок ленты пожертвований**



Статические элементы, нужно захардкодить на фронте:

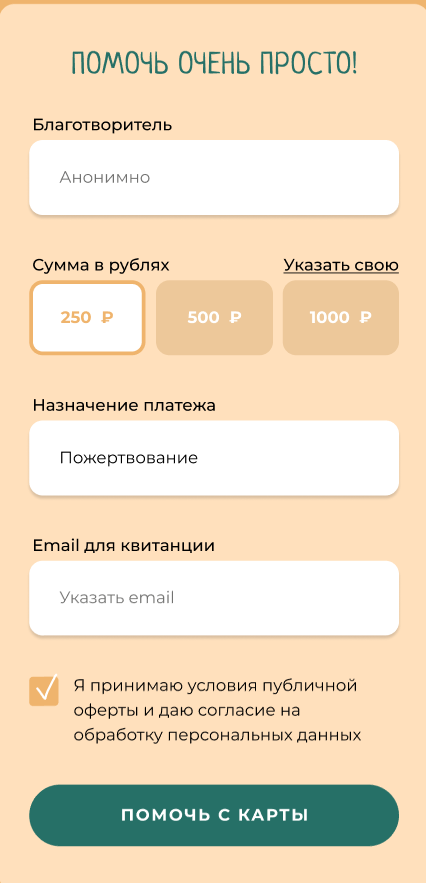
* Заголовок «Лента пожертвований» (вместо «отчет о поступлениях» с макета)
* Текущая дата на момент открытия страницы
* Кнопка «Посмотреть все отчеты», перенаправляет пользователя на страницу очтетов
* Текст «Р» справа от суммы

**GET** /main – метод для получения данных для блоков ленты пожертвований, блога и фотоотчетов

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Имя благотворителя (левый столбец) | don\_name | Если значение в атрибуте пустое, нужно отобразить текст «Аноним» |
| Сумма пожертвования | don\_sum |  |
|  | don\_created\_at | Атрибут, по которому нужно осуществлять сортировку элементов по убыванию от самого нового пожертвования к самому старому |

**Блоки для пожертвования на счет фонда и по QR-коду**

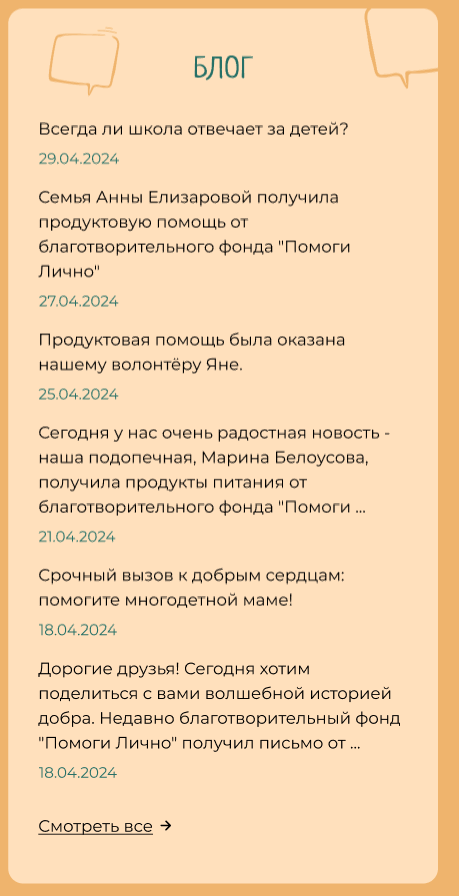
Пока не спроектирована интеграция для оплаты

Статические элементы, нужно захардкодить на фронте:

* Названия заголовков
* Поля и заголовки для ввода значений, кнопка «помочь с карты» (точный набор полей и заголовков возможно будет описать после проектирования интеграции)
* Все элементы блока «сделать добро просто» с оплатой по QR-коду

**Блок статей блога**



Статические элементы, нужно захардкодить на фронте:

* Заголовок «Блог»
* Кнопка «Смотреть все», перенаправляет пользователя на страницу блога

**GET** /main – метод для получения данных для блоков ленты пожертвований, блога и фотоотчетов

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Название статьи | article\_name |  |
| Дата публикации | article\_publish\_date | Нужно переформатировать значение в формат дд.мм.гггг;  по этому атрибуту нужно осуществлять сортировку элементов по убыванию от самой новой статьи к самой старой |

**Блок с фотоотчетами**



Статические элементы, нужно захардкодить на фронте:

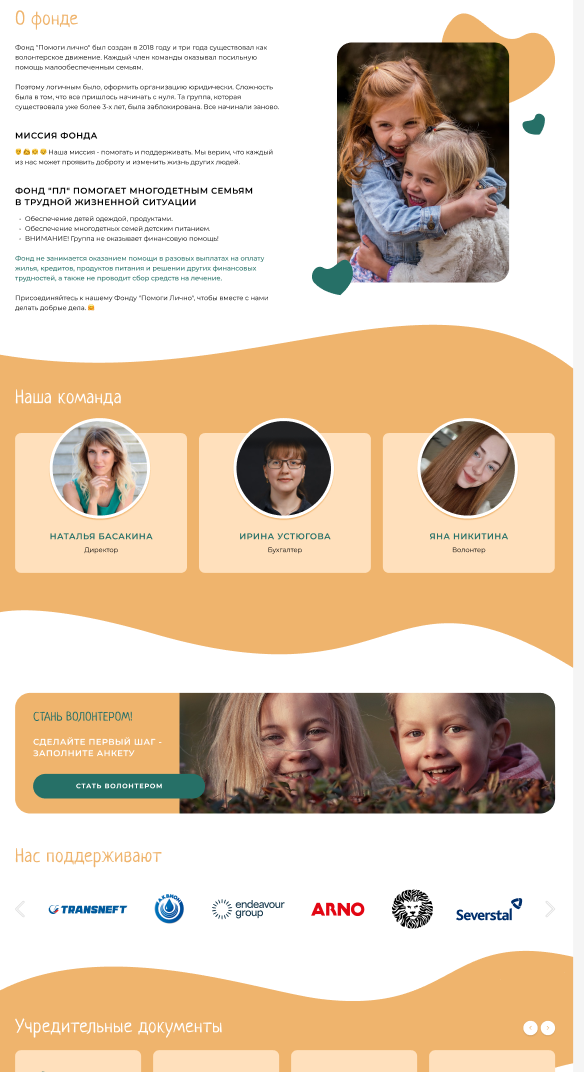
* Заголовок «Фотоотчет»
* Кнопка «Все дети», перенправляет пользователя на страницу «Кому помогли»

**GET** /main – метод для получения данных для блоков ленты пожертвований, блога и фотоотчетов

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Фото | phrep\_path | Если в массиве photos передается несколько phrep\_path, на превью отображать первую из списка |
| Описание под фотографией | phrep\_description |  |
|  | phrep\_created\_at | Атрибут, по которому нужно осуществлять сортировку элементов по убыванию от самого новой карточки фотоотчета к самой старой |

При нажатии на элемент фотоотчета нужно открывать модальное окно просмотрщика фотографии; если фотографий несколько, нужна возможность перелистывать их в модальном окне.

Страница «О фонде»

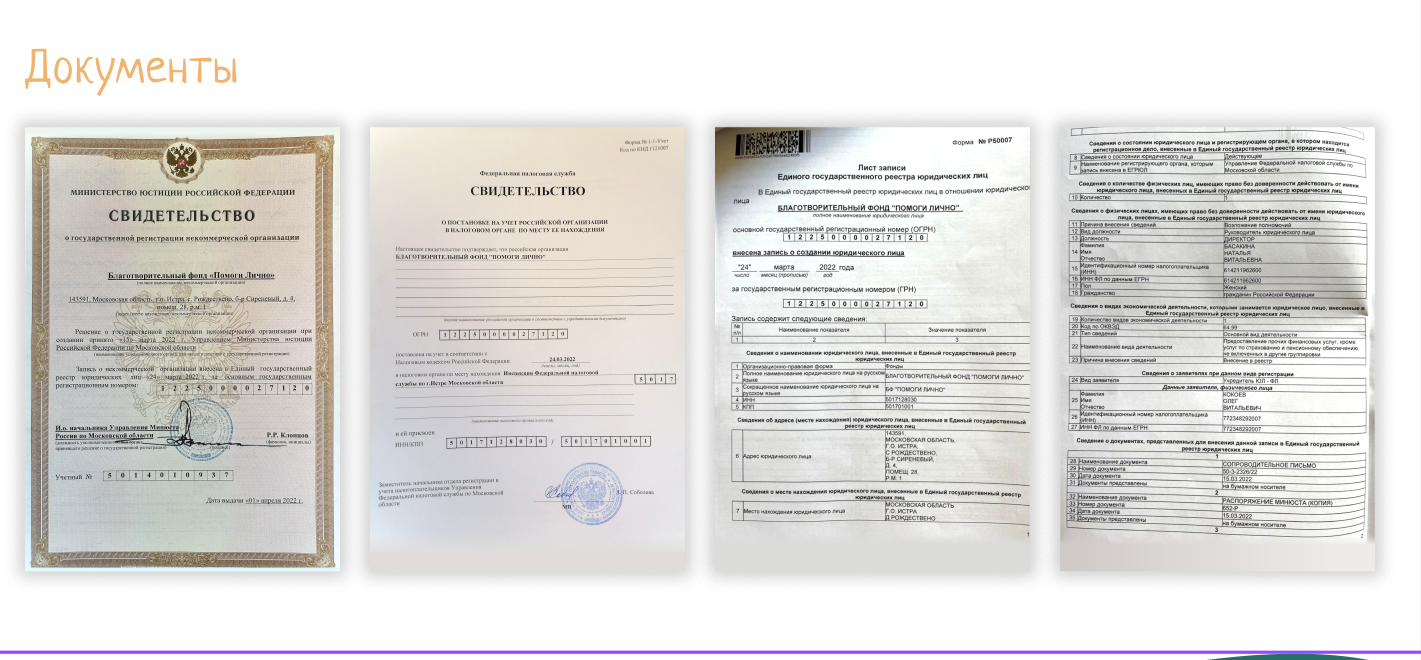


Все элементы страницы статичные, в соответствии с макетами, нужно захардкодить на фронте. Кнопка «Стать волонтером» перенаправляет пользователя на страницу «Стать волонтером».

Изображения и точные тексты нужно запросить у заказчика.

Блок «Учредительные документы» на этой странице НЕ нужен.

Страница «Документы»



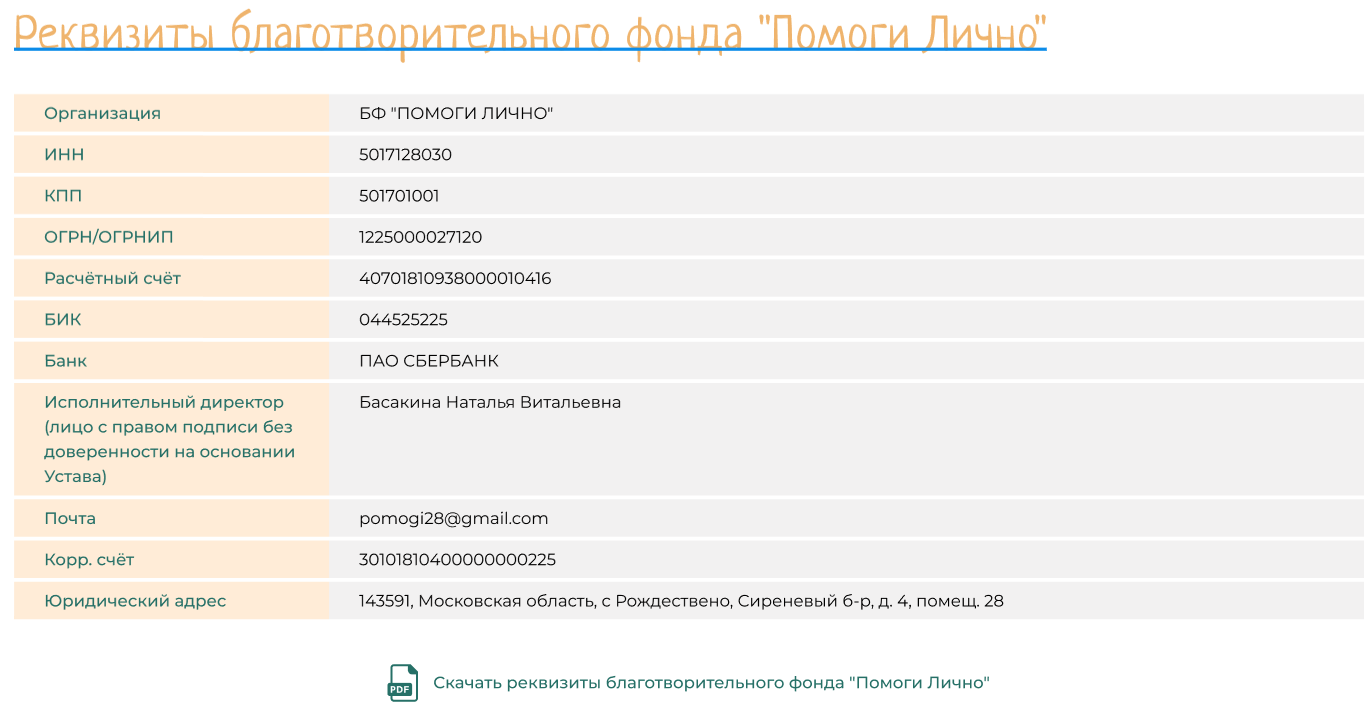
Статические элементы, нужно захардкодить на фронте: заголовок «Документы»

**GET** /documents – метод для получения списка документов

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Иконка документа | doc\_path | Ссылка на документ в хранилище, документы в формате pdf. Визуальный вид иконки можно взять с макета «о фонде» из блока учредительных документов |
| Название документа под иконкой | doc\_name |  |

При нажатии на иконку документа нужно открывать просмотрщик pdf.

Страница «Реквизиты»

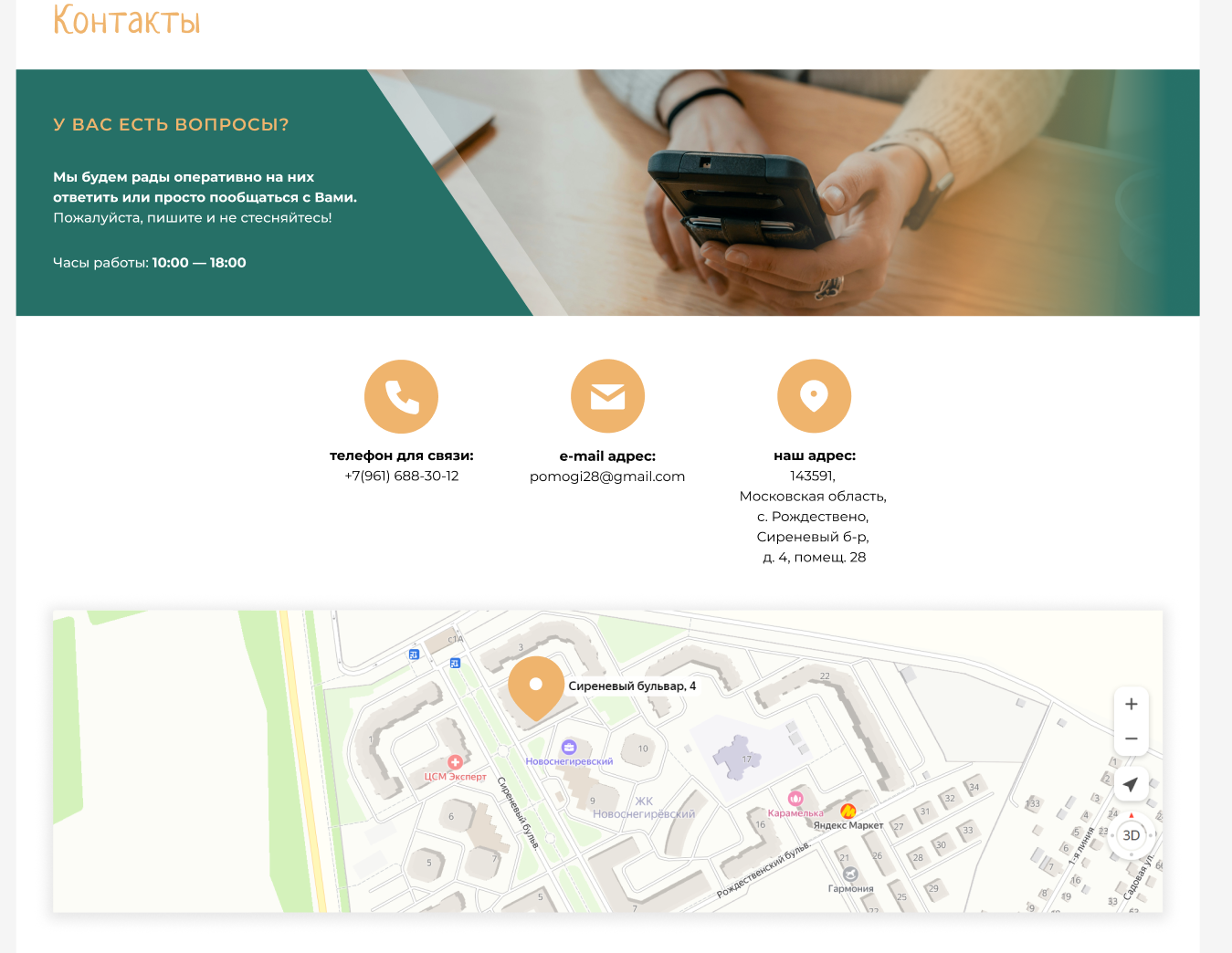


Статические элементы, нужно захардкодить на фронте: заголовок таблицы «Реквизиты благотворительного…», названия строк из левого столбца таблицы и текст для ссылки для скачивания.

**GET** /banking – метод для получения реквизитов

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Организация | organization |  |
| ИНН | inn |  |
| КПП | kpp |  |
| ОГРН/ОГРНИП | ogrn |  |
| Расчетный счет | account |  |
| БИК | bik |  |
| Банк | bank |  |
| Исполнительный директор | director |  |
| Почта | email |  |
| Корр. счет | corr\_account |  |
| Юридический адрес | address |  |
| «Скачать реквизиты…» | doc\_path | Ссылка на документ в хранилище, документы в формате pdf |

Страница «Контакты»

****

Статические элементы, нужно захардкодить на фронте: все заголовки и тексты, включая телефон, email и адрес, в соответствии с макетом.

Для отображения карты предлагаем использовать JavaScript API Яндекс-карт, документация по интеграции здесь: https://yandex.ru/maps-api/docs/js-api/index.html

Страница подопечного

**Карточка семьи**



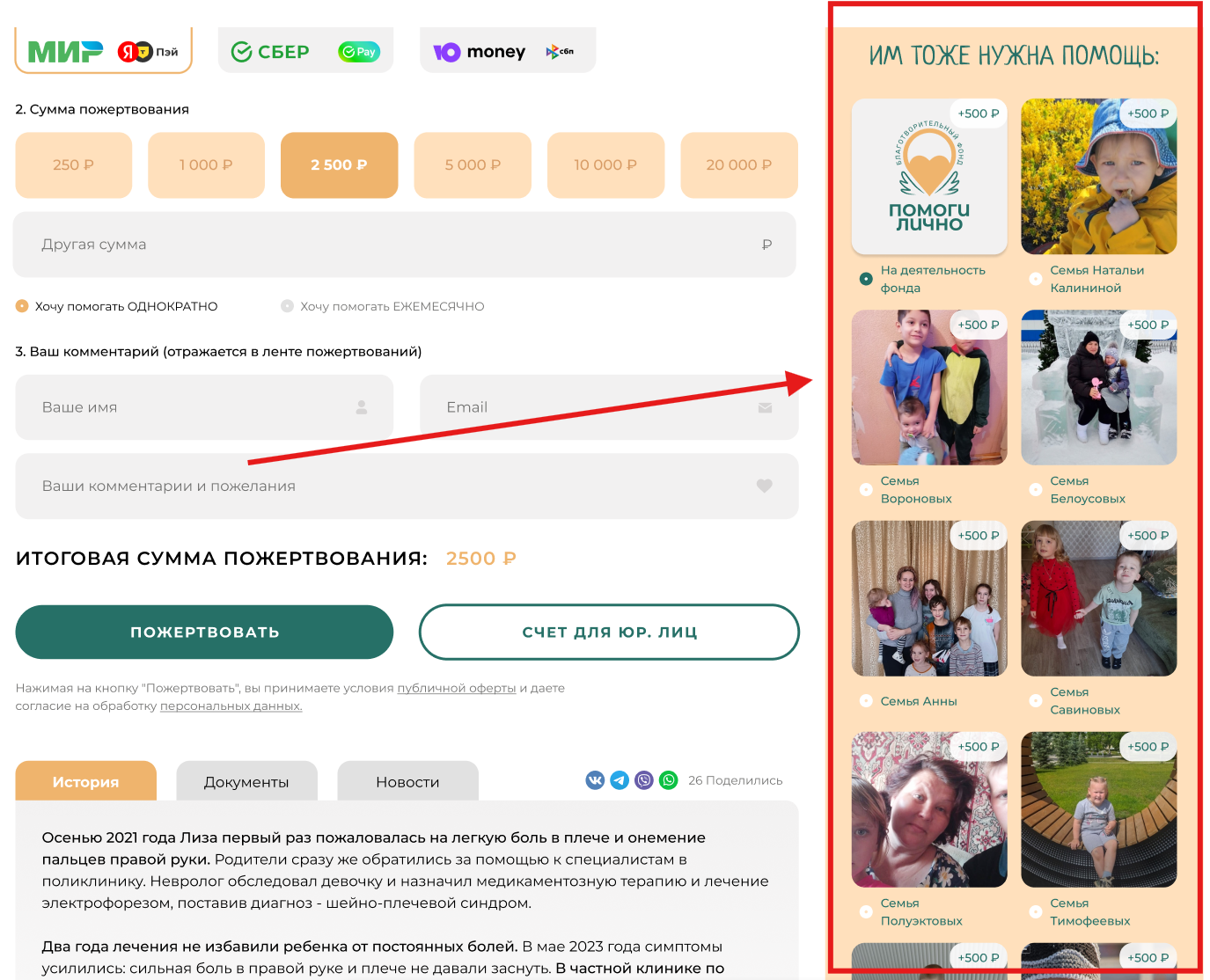
Статические элементы, нужно захардкодить на фронте: тексты «Уже собрано», «из», «Р».

**GET** /families/{f\_id} – метод для получения данных для страницы конкретной семьи (карточка семьи, история семьи, лента комментариев, список других семей, которым можно помочь). В параметре необходимо передавать id семьи (f\_id) с которой был совершен переход к ее странице

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Название семьи в заголовке блока | f\_name |  |
| Основное фото с крупным превью | fph\_main\_path | Ссылка на фото в хранилище |
| Остальные фото | family\_photos:  fph\_path | Ссылки на фото, могут быть пустые |
| Число после «Уже собрано» | f\_current\_sum |  |
| Число после «из» | f\_sum |  |
| Краткое описание под суммами | f\_short\_desc |  |
| Полное описание | f\_desc |  |

При нажатии на превью фото нужно открывать модальное окно просмотрщика фотографии; если фотографий несколько, нужна возможность перелистывать их в модальном окне

**Блок «Им тоже нужна помощь»**



Статические элементы, нужно захардкодить: заголовок блока «Им тоже нужна помощь». Элементы радиокнопки и «+500р» не нужны.

**GET** /families/{f\_id}

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Фото семьи | other\_fph\_path | Ссылка на фото для превью |
| Название семьи | other\_f\_name |  |
|  | other\_f\_created\_at | По этому атрибуту нужно делать сортировку элементов по убыванию (от новых к старым) |
|  | other\_f\_id |  |

Нажатие на фотографию в этом блоке должно перенаправлять пользователя на страницу соответствующей семьи, подставляя значение other\_f\_id в параметр запроса

**Блок История/Документы/Новости**



Статические элементы, нужно захардкодить на фронте: названия заголовка «История»; «Документы» и «Новости» не нужны

**GET** /families/{f\_id}

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Текст истории | f\_history |  |

**Блок с лентой комментариев**

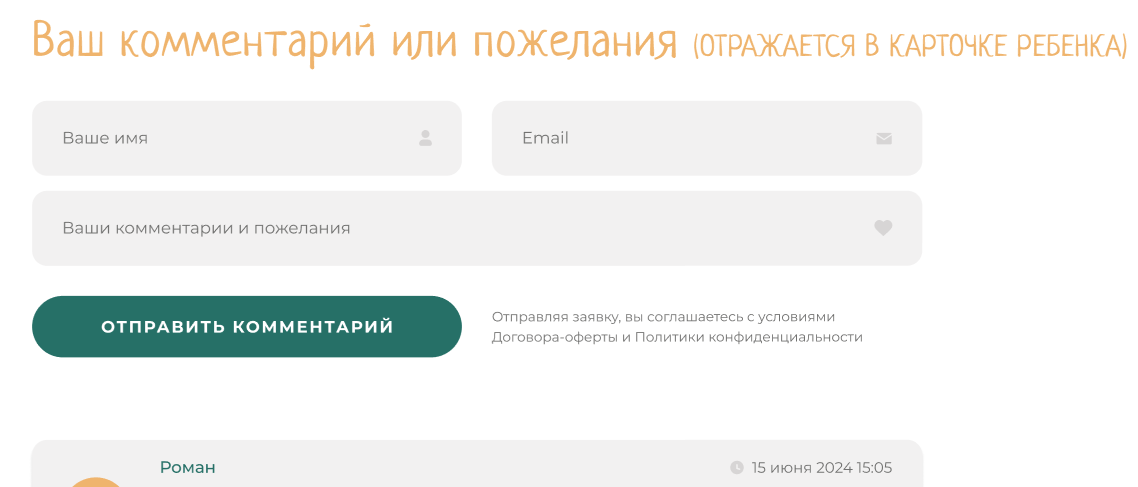


**GET** /families/{f\_id}

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Имя комментатора | fcomm\_name |  |
| Дата и время публикации комментария | fcomm\_publish\_date | Значение нужно преобразовать в формат, в соответствии с макетом: дд <месяца> гггг чч:мм |
| Текст комментария | fcomm\_text |  |
| Количество лайков | fcomm\_likes |  |

1. Комментарии должны быть упорядочены по убыванию по полю fcomm\_publish\_date (от самых новых к самым старым)
2. Иконку автора комментария можно реализовать в соответствии с макетом, используя в качестве буквы первый символ из поля fcomm\_name

**Блок добавить комментарий в ленту комментариев семьи**



Статические элементы, нужно захардкодить на фронте:

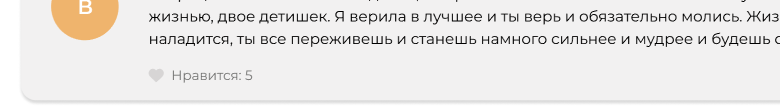
* Заголовок «Ваш комментарий или пожелания», текст в скобках не нужен
* Поля с текстами для ввода данных
* Текст справа от кнопки «отправляя заявку, вы соглашаетесь…» не нужен

**POST** /families/comment

|  |  |  |
| --- | --- | --- |
| **Поле, из которого надо брать значение для атрибута в тело запроса** | **Атрибут** | **Комментарий** |
| Ваше имя | name |  |
| Email | email |  |
| Ваши комментарии и пожелания | text |  |

Также в теле нужно передать id семьи, для которой добавляется комментарий. После получения ответа об успешном добавлении комментария в базу нужно отобразить новый комментарий вверху списка

**Проставление лайка комментарию** (по нажатию на кнопку «нравится»)

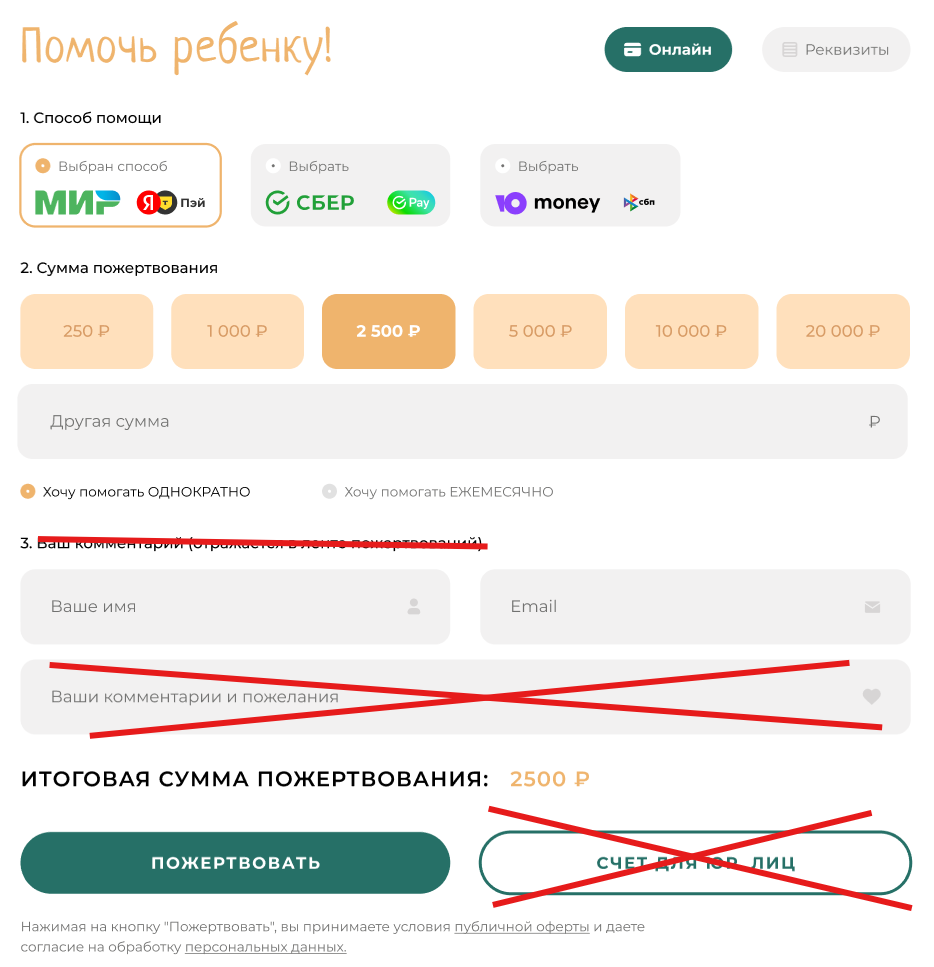
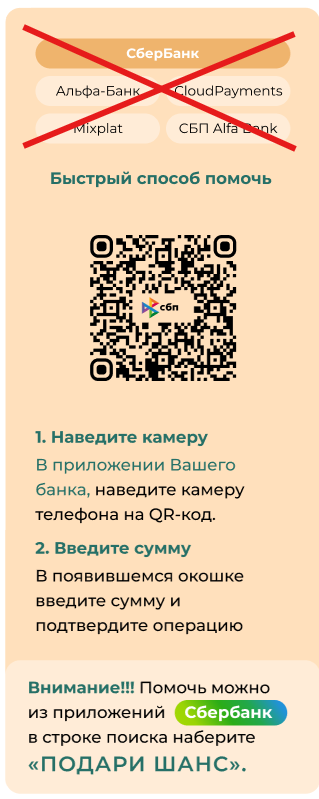


**POST** /families/comment/like

В теле запроса необходимо передать id комментария, которому проставляется лайк

**Блоки для пожертвования на счет фонда и по QR-коду**

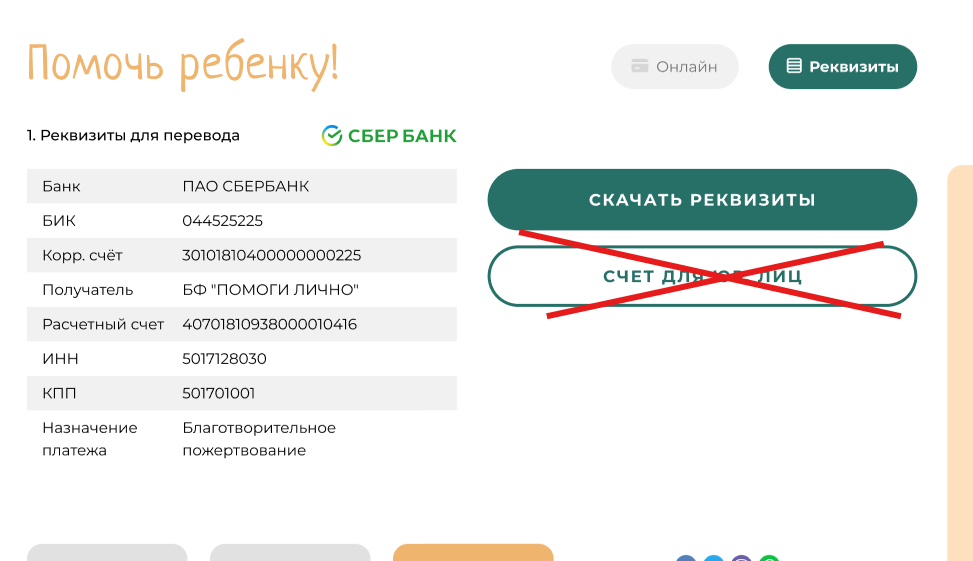
Пока не спроектирована интеграция для оплаты

Статические элементы, нужно захардкодить на фронте:

* Название заголовка, поменять на «Помочь семье!»
* Кнопки «Онлайн», «Реквизиты» + новая кнопка рядом с ними «Помочь лично»
* Поля и заголовки для ввода значений для оплаты онлайн, кнопка «Пожертвовать» (точный набор полей и заголовков возможно будет описать после проектирования интеграции)
* Текст «Нажимая на кнопку…» под кнопкой «Пожертвовать». В текст «персональных данных» нужно положить ссылку на страницу стороннего веб-сайта, будет получена от заказчика
* Все элементы блока «быстрый способ помочь» с оплатой по QR-коду

**Разрез «Реквизиты»:**

****

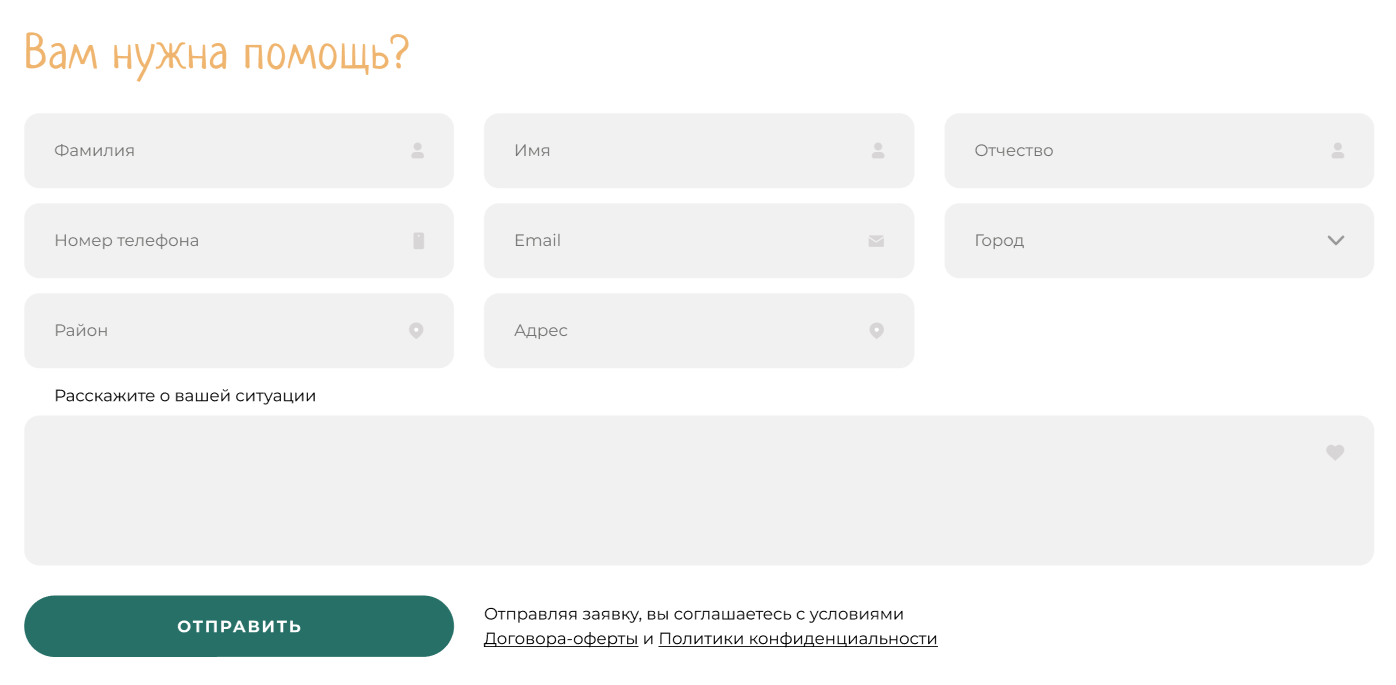
Статические элементы, нужно захардкодить на фронте: список наименований в левом столбце, кнопка «Скачать реквизиты»

**GET** /banking – метод для получения реквизитов

|  |  |  |
| --- | --- | --- |
| **Элемент интерфейса** | **Атрибут** | **Комментарий** |
| Банк | bank |  |
| БИК | bik |  |
| Корр. счет | corr\_account |  |
| Получатель | organization |  |
| Расчетный счет | account |  |
| ИНН | inn |  |
| КПП | kpp |  |
| Назначение платежа | purpose |  |
| «Скачать реквизиты» | doc\_path | Ссылка на документ в хранилище, документы в формате pdf |

**Отправить заявку на личную помощь семье**

Элемент не отрисован на макетах: по нажатию на кнопку «Помочь лично» (должна находиться справа от кнопок «Онлайн» и «Реквизиты») нужно открывать модальное окно с полями для заполнения заявки, по аналогии с заявками на получение помощи и волонтерство.



В модальном окне должен быть статичный заголовок «Хотите помочь семье лично? \nОтправьте нам заявку, мы свяжемся с вами и расскажем, как это можно сделать», поля для заполнения, кнопка «Отправить» и текст «Отправляя заявку, вы соглашаетесь…» со ссылками на страницы стороннего веб-сайта (будут получены от заказчика).

Список полей для заполнения:

* Фамилия (текст)
* Имя (текст)
* Отчество (текст)
* Номер телефона (текст)
* Email (текст)
* Город (текст)
* Какой семье хотите помочь (текст)
* Расскажите, почему и как бы вы хотели помочь (текст)

**POST** /families/personalhelp\_request

|  |  |  |
| --- | --- | --- |
| **Поле, из которого надо брать значение для атрибута в тело запроса** | **Атрибут** | **Комментарий** |
| Фамилия | surname |  |
| Имя | name |  |
| Отчество | patname | Необязательное поле |
| Номер телефона | phone |  |
| Email | email |  |
| Город | city |  |
| Какой семье хотите помочь | fname |  |
| Расскажите, почему и как бы вы хотели помочь | description |  |

**POST** /email/messages

Метод внешней системы рассылки писем NotiSend (ссылка на [документацию](https://notisend.ru/dev/email/api/?_ga=2.184028064.702638767.1741103576-1887278183.1741103576#TOC_d7a6319e563f08691be55897faac38c2) на сайте).

Для того чтобы получить API-ключ для заголовка «Authorization: Bearer» необходимо зарегистрироваться на сайте, ключ будет в личном кабинете.

Заполнение атрибутов в теле запроса:

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| to | Email получателя |
| from\_email | Email отправителя, необходимо запросить у заказчика, будет один постоянный |
| from\_name | Имя отправителя, необходимо запросить у заказчика, будет одно постоянное |
| subject | Тема письма |
| text | Тело письма |

После того как пользователь оставляет заявку необходимо два раза вызвать запрос:

1) для отправки заявки на почту фонда

2) для отправки уведомления о том, что заявка принята, пользователю, который оставил заявку.

*Заявка на почту фонда*:

- в атрибут получателя «**to**» нужно добавить почту фонда (необходимо запросить у заказчика, будет одна постоянная)

- в атрибут темы письма «**subject**» нужно добавить текст «Новая заявка на личную помощь»

- в атрибут тела письма «**text**» нужно добавить значения из всех полей заявки по шаблону:

*С сайта поступила новая заявка от человека, который хочет помочь семье лично.*

*<Фамилия>: <Иванов>*

*<Поле заявки2>: <Значение, которое пользователь ввел в поле>*

*<Поле заявки3>: <Значение, которое пользователь ввел в поле>*

*…*

*Уведомление пользователю, который оставил заявку*:

- в атрибут получателя «**to**» нужно добавить почту, которую пользователь ввел в поле «Email»

- в атрибут темы письма «**subject**» нужно добавить текст «Уведомление по заявке в Фонд «Помоги лично»»

- в атрибут тела письма «**text**» нужно добавить текст «Ваша заявка находится в обработке»