

## ВВЕДЕНИЕ

**Технологии этого модуля:** Клиентское программирование

**Время на выполнение:** 3 часа

К вам обратилась организация, оказывающая помощь в поиске потерянных домашних животных с просьбой разработать сервис GET PET BACK, на котором пользователи могут оставлять информацию о найденных животных, а владельцы, у которых пропал питомец, могут их искать.

Заказчик предоставляет вам полностью готовую верстку со всеми страницами и рабочее API. Вам необходимо использовать все имеющиеся навыки в клиентской разработке для создания Single Page Application, далее SPA.

Заказчик хочет, чтобы API можно было легко поддерживать, поэтому использование JavaScript фреймворков будет плюсом.

**ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!**

## ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – реализовать SPA приложение, которое будет работать с уже разработанным API.

Для вашего удобства во всех URL будет использоваться переменная {host} которая обозначает хост адрес API: <http://??????.wsr.ru/api> – где ?????? Ваш логин.

Для тестирования работы запросов, требующих авторизации пользователя используйте:

**Телефон:** 89111234567

**Пароль:** Password123

Ваше SPA должно состоять из следующих экранов (сверстанные страницы предоставляются):

- Главная
- Страница входа в личный кабинет
- Страница регистрации в личном кабинете
- Страница личного кабинета пользователя
- Страница добавления информации о найденном животном
- Страница поиска животных
- Страница с карточкой найденного животного

Приложение должно обладать следующим функционалом:

### 1. Главная

**Слайдер с объявлениями с животными, у которых были найдены хозяева**, должен работать корректно. До загрузки объявлений вместо слайдера отображается прелоадер. В случае отсутствия найденных хозяев блок не отображается.

Request	Response
<b>URL:</b> {host}/api/pets/slider <b>Method:</b> GET  <i>Для получения пустого массива вы можете воспользоваться запросом</i>  <b>URL:</b> {host}/api/pets/slider/empty <b>Method:</b> GET	<p>-----Successful -----</p> <p><b>Status:</b> 200 <b>Content-Type:</b> application/json <b>Body:</b></p> <pre>{   "data": {     "pets": [       {         "id": 1,         "kind": "Кошка",         "description": "Найдена маленькая кошечка",         "image": "url"       }     ]   } }</pre>

	<p>----- <b>No results</b> -----</p> <p><b>Status:</b> 200  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre>{   "data": {     "pets": []   } }</pre>
--	---

### Быстрый поиск по объявлениям

Поиск осуществляется по объявлениям по полю description. Возвращаются объявления, в описании которых присутствует переданное строковое значение.

При вводе больше 3-х символов должны появиться подсказки по вводу, загруженные с сервера. Также необходимо реализовать функцию debounce (запрос на сервер уходит с задержкой 1000ms после окончания ввода последнего символа для поиска)

Request	Response
<p><b>URL:</b> {host}/api/search  <b>Method:</b> GET</p> <p><b>Query string (GET parameters):</b>  - query</p>	<p>----- <b>Successful</b> -----</p> <p>-----</p> <p><b>Status:</b> 200  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre>{   "data": {     "orders": [       {         "id": 1,         "kind": "кошка",         "photos": '{url}/img1.png',         "description": "Найдена кошка, порода           Сфинкс, очень грустная",         "mark": "VL-0214",         "district": "Василеостровский",         "date": "01-01-1970",       }     ]   } }</pre> <p>----- <b>No results</b> -----</p> <p>-----</p> <p><b>Status:</b> <del>204</del> 200  <b>Content-Type:</b> application/json  <b>Body:</b></p>

	<pre>{   "data": {     "orders": []   } }</pre>
--	---

mark - клеймо

Если результатов нет, появляется соответствующее сообщение.

### Карточки найденных животных

Выводится информация о 6 найденных животных, найденных в последнее время.

Данные в блоке выводятся с сортировкой по убыванию даты.

Request	Response
<b>URL:</b> {host}/api/pets <b>Method:</b> GET  <b>Query string (GET parameters):</b> <b>Headers</b> - <b>Content-Type:</b> application/json  <b>Body:</b> <pre>{ }</pre>	<p>----- <b>Successful</b> -----</p> <p>-----</p> <p><b>Status:</b> 200  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre>{   "data": {     "orders": [       {         "id": 1,         "phone": "89112345678",         "name": "Иван",         "kind": "кошка",         "photo": '{url}/img1.png',         "description": "Найдена кошка, порода           Сфинкс, очень грустная",         "mark": "VL-0214",         "district": "Василеостровский",         "date": "01-01-1970",         "registred": true       }     ]   } }</pre>

registred - true, если информация о найденном животном была добавлена зарегистрированным пользователем, иначе - false

### Подписка на новости

Форма для подписки на новости, в случае успешной подписки, заменяется на сообщение о успешной подписке, в случае ошибки появляется сообщение об ошибке с текстом ошибки.

email: обязательное

Request	Response
<b>URL:</b> {host}/api/subscription <b>Method:</b> POST <b>Body:</b> <pre>{   "email": "email" }</pre>	<p>----- <b>Successful</b> -----</p> <p>-</p> <p><b>Status:</b> <del>200</del> 204</p> <p>----- <b>Validation error</b> -----</p> <p><b>Status:</b> 422  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre>{   "error": {     "code": 422,     "message": "Validation error",     "errors": {       &lt;key&gt;: &lt;массив ошибок&gt;     }   } }</pre>

## Регистрация

Запрос для регистрации нового пользователя в системе. При отправке запроса необходимо передать объект со следующими свойствами:

- name – обязательное поле, строка, допустимые символы - кириллица, пробел, дефис
- phone – обязательное, строка, только цифры, и знак +
- email – обязательное поле, строка, формат - email
- password – обязательное поле, строка, не менее 7 символов, обязательно: 1 цифра, 1 строчная, 1 заглавная буквы
- password\_confirmation – обязательное поле, строка, значение должно совпадать со значением password
- confirm – обязательное поле, integer 0 или 1 - для элемента, подтверждающего согласие пользователя на обработку персональных данных. Если нет подтверждения согласия на обработку персональных данных, то необходимо вывести соответствующее сообщение в массиве ошибок

*Вся валидация проводится средствами JS!*

*Отправка некорректных данных на сервер невозможна. В случае успешной регистрации отображается соответствующее сообщение, в случае ошибки появляется сообщение с текстом ошибки.*

Request	Response
<b>URL:</b> {host}/api/register <b>Method:</b> POST  <b>Headers</b> - <b>Content-Type:</b> application/json	<p>----- <b>Successful</b> -----</p> <p><b>Status:</b> 204</p> <p>----- <b>Validation error</b> -----</p> <p><b>Status:</b> 422  <b>Content-Type:</b> application/json</p>

<b>Body:</b> <pre>{   "name": "Иван",   "phone": "89001234567",   "email": "user@user.ru",   "password": "paSSword1",   "password_confirmation": "paSSword1",   "confirm": 1, }</pre>	<b>Body:</b> <pre>{   "error": {     "code": 422,     "message": "Validation error",     "errors": {       &lt;key&gt;: &lt;массив ошибок&gt;     }   } }</pre>
--	--

### Страница аутентификации

Запрос для аутентификации пользователя в системе. При отправке запроса необходимо передать объект с логином и паролем.

- email – обязательное
- password – обязательное

Отправка некорректных данных на сервер невозможна. В случае успешной регистрации отображается соответствующее сообщение, в случае ошибки появляется сообщение с текстом ошибки.

Request	Response
<b>URL:</b> {host}/api/login <b>Method:</b> POST  <b>Headers</b> - <b>Content-Type:</b> application/json  <b>Body:</b> <pre>{   "email": "user@user.ru",   "password": "paSSword1" }</pre>	<p>----- <b>Successful</b> -----</p> <p><b>Status:</b> 200  <b>Content-Type:</b> application/json  <b>Body:</b>  <pre>{   "data": {     "token": &lt;сгенерированный token&gt;   } }</pre></p> <p>----- <b>Validation error</b> -----</p> <p><b>Status:</b> 422  <b>Content-Type:</b> application/json  <b>Body:</b>  <pre>{   "error": {     "code": 422,     "message": "Validation error",     "errors": {       &lt;key&gt;: &lt;массив ошибок&gt;     }   } }</pre></p>

	<p>----- <b>Unauthorized</b> -----</p> <p><b>Status:</b> 401</p> <p><b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{   "error": {     "code": 401,     "message": "Unauthorized",     "errors": {       "phone": [ "phone or password incorrect" ]     }   } }</pre>
--	--

### Страница поиска по объявлениям

Поиск осуществляется по объявлениям по переданным значениям: район, вид животного.

Должно быть передано на сервер хотя бы одно из указанных значений.

Поиск по полю "район" - строгий (полное соответствие, например, "Центральный"),

по полю "вид" - нестрогий (например, для "ко" должны отобразиться объявления, в которых указан вид "кошка", "кот", "котик", "собачко" и т.д.)

Необходимо отобразить все объявления, в полях которых присутствуют введенные параметры.

Вам следует реализовать пагинацию для отображения результатов поиска, позволяющую просматривать другие результаты поиска - не более 10 животных на одной странице.

Если результатов поиска нет, необходимо отобразить соответствующую информацию.

Request	Response
<p><b>URL:</b> {host}/api/search/order/  <b>Method:</b> GET</p> <p><b>Query string (GET parameters):</b>  - query  (/?district=Василеостровский&amp;kind=ко)</p>	<p>Response</p> <p>----- <b>Successful</b> -----</p> <p>-----</p> <p><b>Status:</b> 200</p> <p><b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{   "data": {     "orders": [       {         "id": 1,         "phone": "89112345678",         "email": "user@user.ru",         "name": "Иван",         "kind": "кошка",         "photos": '{url}/img1.png',         "description": "Найдена кошка, порода Сфинкс, очень грустная",         "mark": "VL-0214", </pre>

	<pre>         "district": "Василеостровский",         "date": "01-01-1970",       }     ]   } } </pre> <p>----- <b>No results</b> -----</p> <p>-----</p> <p><b>Status:</b> 204  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre> {   "data": {     "orders": []   } } </pre>
--	---

### Личный кабинет

Должна быть возможность выйти из личного кабинета.

Необходимо позаботиться об уведомлении пользователей о каких-либо действиях (ошибки валидации, подтверждения и т.п).

В личный кабинет возможен вход только авторизованного пользователя. Если пользователь не авторизован, должна происходить переадресация на страницу с аутентификацией.

### Функциональные возможности личного кабинета:

#### - Информация о пользователе

Только для авторизованного пользователя.

Количество дней, которое прошло с момента регистрации пользователя на сайте рассчитывается на стороне клиента относительно текущей даты и полученного значения registrationDate

В случае запроса неавторизованным пользователем должна происходить переадресация на страницу с аутентификацией.

При запросе информации о другом пользователе необходимо отобразить сообщение об ошибке доступа.

Request	Response
<p><b>URL:</b> {host}/api/users/{id}</p> <p><b>Method:</b> GET</p> <p><b>Headers</b></p> <ul style="list-style-type: none"> <li>- <b>Content-Type:</b> application/json</li> <li>- <b>Authorization:</b> Bearer {token}</li> </ul> <p><b>Body:</b></p>	<p>----- <b>Successful</b> -----</p> <p>-----</p> <p><b>Status:</b> 200  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre> {   "data": {     "user": [ </pre>



<pre>{ }</pre>	<pre>{   "id": 1,   "phone": "89112345678",   "email": "mail@email.ru",   "name": "Иван",   "registrationDate": "01-01-1970",   "ordersCount": 4,   "petsCount": 2 }</pre> <p>----- <b>Unauthorized</b> -----</p> <p><b>Status:</b> 401  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre>{   "error": {     "code": 401,     "message": "Unauthorized"   } }</pre>
----------------	---

registrationDate - дата регистрации пользователя

ordersCount - количество объявлений, добавленных пользователем

petsCount - количество животных, у которых нашлись хозяева из числа тех, объявления о которых были добавлены текущим пользователем

- Изменение номера телефона
- phone – обязательное

Request	Response
<p><b>URL:</b> {host}/api/users/{id}/phone  <b>Method:</b> PATCH</p> <p><b>Headers</b></p> <ul style="list-style-type: none"> <li>- <b>Content-Type:</b> application/json</li> <li>- <b>Authorization:</b> Bearer {token}</li> </ul> <p><b>Body:</b></p> <pre>{   "phone": "+79112345678", }</pre>	<p>----- <b>Successful</b> -----</p> <p>-----</p> <p><b>Status:</b> 200  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre>{   "data": {     "status": "ok"   } }</pre> <p>----- <b>Unauthorized</b> -----</p> <p><b>Status:</b> 401  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre>{</pre>

	<pre> "error": {   "code": 401,   "message": "Unauthorized" } } </pre> <p>----- <b>Validation error</b> -----  (Если phone пустое)</p> <p><b>Status:</b> 422  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre> {   "error": {     "code": 422,     "message": "Validation error",     "error":       "The phone should not be empty."   } } </pre>
--	---

- Изменение адреса электронной почты

- email – обязательное, изменение на строку, не являющуюся адресом электронной почты, недопустимо

Request	Response
<p><b>URL:</b> {host}/api/users/{id}/email  <b>Method:</b> PATCH</p> <p><b>Headers</b>  - <b>Content-Type:</b> application/json  - <b>Authorization:</b> Bearer {token}</p> <p><b>Body:</b></p> <pre> {   "email": "newmail@user.ru", } </pre>	<p>----- <b>Successful</b> -----  -----</p> <p><b>Status:</b> 200  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre> {   "data": {     "status": "ok"   } } </pre> <p>----- <b>Unauthorized</b> -----</p> <p><b>Status:</b> 401  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre> {   "error": {     "code": 401,     "message": "Unauthorized"   } } </pre>

	<p>----- Validation error -----</p> <p>(Если email пустое)</p> <p><b>Status:</b> 422</p> <p><b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{   "error": {     "code": 422,     "message": "Validation error",     "error":       "The email should not be empty."   } }</pre>
--	---

- Объявления, добавленные пользователем

Статусы объявлений

active - активное

wasFound - хозяин найден

onModeration - на модерации

archive - в архиве

Request	Response
<p><b>URL:</b> {host}/api/users/orders/{id}</p> <p><b>Method:</b> GET</p> <p><b>Headers</b></p> <ul style="list-style-type: none"> <li>- <b>Content-Type:</b> application/json</li> <li>- <b>Authorization:</b> Bearer {token}</li> </ul> <p><b>Body:</b></p> <pre>{ }</pre>	<p>----- Successful -----</p> <p>-----</p> <p><b>Status:</b> 200</p> <p><b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{   "data": {     "orders": [       {         "id": 1,         "kind": "кошка",         "photos": '{url}/img1.png',         "description": "Найдена кошка, порода           Сфинкс, очень грустная",         "mark": "VL-0214",         "district": "Василеостровский",         "date": "01-01-1970",       },       ...     ],   } }</pre>

	<p>----- <b>No results</b> -----</p> <p>-----</p> <p><b>Status:</b> 204</p> <p><b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{   "data": {     "orders": []   } }</pre>
--	--

### Удаление пользователем объявления

Удаление возможно только своих объявлений со статусом active и onModeration

Перед удалением необходимо подтверждение пользователем.

После запроса на сервер, объявление на страницах сайта не отображается

Request	Response
<p><b>URL:</b> {host}/api/users/orders/{id}</p> <p><b>Method:</b> DELETE</p> <p><b>Headers</b></p> <ul style="list-style-type: none"> <li>- <b>Content-Type:</b> application/json</li> <li>- <b>Authorization:</b> Bearer {token}</li> </ul> <p><b>Body:</b></p> <pre>{ }</pre> <p>-----order not found -----</p> <pre>{   "error": {     "code": 404,     "message": "Order doesn't exist"   } }</pre>	<p>----- <b>Successful</b> -----</p> <p>-----</p> <p><b>Status:</b> 200</p> <p><b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{   "data": {     "status": "ok"   } }</pre> <p>----- <b>Unauthorized</b> -----</p> <p><b>Status:</b> 401</p> <p><b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{   "error": {     "code": 401,     "message": "Unauthorized"   } }</pre> <p>----- <b>Validation error</b> -----</p> <p>(например, для объявлений с другими статусами)</p> <p><b>Status:</b> 403</p> <p><b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{   "error": {</pre>

	<pre> "code": 403, "message": "Access denied", } </pre>
--	---

### Редактирование пользователем объявления

Редактирование возможно только объявлений со статусом active и onModeration.

Пользователь может редактировать только свои объявления.

Поля, доступные для редактирования:

- photo1, photo2, photo3 - изображения, в формате png, photo1 - обязательное поле
- mark - клеймо, необязательное строковое поле в произвольном формате
- description - текст, описание, дополнительная информация

Request	Response
<b>URL:</b> {host}/api/pets/{id} <b>Method:</b> <del>PATCH</del> POST  <b>Headers</b> - <b>Content-Type:</b> multipart/form-data - <b>Authorization:</b> Bearer {token}  <b>Body:</b> <pre> {   "kind": "кошка",   "photo1": FILE,   "photo2": FILE,   "photo3": FILE,   "mark": "VL-1250",   "description": "Веселая собачка породы бультерьер", } </pre> <p>-----order not found -----</p> <pre> {   "error": {     "code": 404,     "message": "Order doesn't exist"   } } </pre>	<p>----- <b>Successful</b> -----</p> <p><b>Status:</b> 200  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre> {   "data": {     "status": "ok"   } } </pre> <p>----- <b>Unauthorized</b> -----</p> <p><b>Status:</b> 401  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre> {   "error": {     "code": 401,     "message": "Unauthorized"   } } </pre> <p>----- <b>Validation error</b> -----  (например, для объявлений с другими статусами)</p> <p><b>Status:</b> 403  <b>Content-Type:</b> application/json  <b>Body:</b></p> <pre> {   "error": {     "code": 403,     "message": "Access denied",   } } </pre>

	<p>----- Validation error -----</p> <p><b>Status:</b> 422</p> <p><b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{   "error": {     "code": 422,     "message": "Validation error",     "errors": {       &lt;key&gt;: &lt;массив ошибок&gt;     }   } }</pre>
--	---

### Страница с карточкой одного животного

Request	Response
<p><b>URL:</b> {host}/api/pets/{id}</p> <p><b>Method:</b> GET</p> <p><b>Headers</b></p> <p>- <b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{ }</pre>	<p>----- Successful -----</p> <p>-----</p> <p><b>Status:</b> 200</p> <p><b>Content-Type:</b> application/json</p> <p><b>Body:</b></p> <pre>{   "data": {     "pet": [       {         "id": 1,         "phone": "89112345678",         "email": "user@user.ru",         "name": "Иван",         "kind": "кошка",         "photos":           ['{url}/img1.png', '{url}/img2.png',            '{url}/img3.png'],         "description": "Найдена кошка, порода                       Сфинкс, очень грустная",         "mark": "VL-0214",         "district": "Василеостровский",         "date": "01-01-1970",       }     ]   } }</pre> <p>----- No results -----</p>

	<pre> ----- Status: 204 404 Content-Type: application/json Body: {   "data": {     "pets": []   } </pre>
--	--

### Страница добавления нового объявления

Добавление нового объявления. При отправке запроса необходимо передать объект со следующими свойствами:

- name – имя пользователя - обязательное поле, строка, допустимые символы - кириллица, пробел, дефис
  - phone – обязательное, строка, только цифры, и знак +
  - email – обязательное поле, строка, формат - email
  - register - Элемент выбора опции автоматической регистрации пользователя при добавлении объявления о найденном животном. Если пользователь выбрал вариант с регистрацией, то появляется блок с полями для ввода пароля и его подтверждения (по умолчанию он скрыт). Если пользователь решил после этого отказаться от регистрации, блок с полями для ввода паролей скрывается (также без использования js).
    - password – обязательное поле, строка, не менее 7 символов, обязательно: 1 цифра, 1 строчная, 1 заглавная буквы
    - password\_confirmation – обязательное поле, строка, значение должно совпадать со значением password
  - photo1, photo2, photo3 - изображения, photo1 - обязательное поле
  - mark - клеймо, необязательное строковое поле в произвольном формате
  - description - текст, описание, дополнительная информация
  - confirm – обязательное поле, integer 0 или 1 - для элемента, подтверждающего согласие пользователя на обработку персональных данных. Если нет подтверждения согласия на обработку персональных данных, то необходимо вывести соответствующее сообщение в массиве ошибок
- Если пользователь авторизован, то поля с телефоном, именем, e-mail заполняются автоматически.

Вся валидация проводится средствами JS!

Отправка некорректных данных на сервер невозможна. В случае успешного добавления объявления отображается соответствующее сообщение, в случае ошибки появляется сообщение с текстом ошибки.

Request	Response
<b>URL:</b> {host}/api/pets/new <b>Method:</b> POST	<pre> ----- Successful ----- Status: 200 Content-Type: application/json </pre>

<b>Headers</b> <b>- Content-Type: multipart/form-data</b>  <b>Body:</b> <pre>{   "name": "Иван",   "phone": "89001234567",   "email": "user@user.ru",   "password": "paSSword1",   "password_confirmation": "paSSword1",   "confirm": 1,   "kind": "собака",   "photo1": FILE,   "photo2": FILE,   "photo3": FILE,   "mark": "VL-1250",   "description": "Веселая собачка породы сенбернар",   "confirm": 1, }</pre>	<b>Body:</b> <pre>{   "data": {     "status": "ok",     "id": 10,   } }</pre> <p>----- Validation error -----</p> <b>Status:</b> 422 <b>Content-Type:</b> application/json <b>Body:</b> <pre>{   "error": {     "code": 422,     "message": "Validation error",     "errors": {       &lt;key&gt;: &lt;массив ошибок&gt;     }   } }</pre>
--	--

## ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Разработанное приложение должно быть доступно по адресу <http://xxxxxx-m2.wsr.ru/>, где xxxxxx - логин участника (указан на индивидуальной карточке).

Проверка работы будет производиться в браузере Google Chrome.

На страницах должны использоваться анимации и микроанимации, реализованные с помощью JS. Созданные анимированные эффекты должны способствовать формированию положительного впечатления о сайте.

**ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!**

## СИСТЕМА ОЦЕНКИ

Секция	Критерий	Сумма
A	Организация работы и управление	1,00
B	Коммуникация и навыки межличностного общения	1,00
C	Графический дизайн	1,00
D	Верстка	0,00
E	Программирование на стороне клиента	18,00



F	Программирование на стороне сервера	0,00
G	CMS	0,00
<b>Всего</b>		21,00