

ВВЕДЕНИЕ

Технологии этого модуля: Клиентское программирование

Время на выполнение: 3 часа

К вам обратилась компания «WhiteTickets» - новая, светлая и энергичная компания, предоставляющая поиск и продажу билетов на любые концерты по Санкт-Петербургу. 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.

Обратите внимание, что в готовой верстке уже присутствуют все необходимые классы для автоматической проверки. Вам запрещается удалять классы, которые начинаются со слова test. Например такие как test-0-logo или test-0-fbs.

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://yddfzaz-m3.wsr.ru~~ <http://tickets.сделай.site>

Ваше SPA должно состоять из следующих экранов:

- Главная
- Вход в личный кабинет
- Регистрация в личном кабинете
- Личный кабинет
- Результаты поиска
- Заказ
- Управление заказом
- Выбор места в зале

Приложение должно обладать следующим функционалом:

1. Домашний экран – на данном экране располагается статическая информация и форма для поиска подходящих концертов со следующими полями:
 - a. Дата от (From) – дата от какой искать концерт, должна быть выполнена маска в формате 10/19/2020
 - b. Дата до (date) – дата до какой искать
 - c. ~~Код концерта (code) – 6 цифр~~ **Тип концерта**
 - d. Количество гостей (guest) – от 1 до 8

При отправке формы пользователю должен отобразиться экран с найденными концертами.

2. Экран с найденными концертами – на данном экране пользователю должен предоставляться список найденных концертов в зависимости от указанных данных, а также адрес мероприятия. Каждый концерт должен содержать:
 - a. Название концерта
 - b. Адрес места мероприятия
 - c. Дата и время концерта
 - d. Тип концерта
 - e. Код концерта
 - f. ~~Кол-во оставшихся билетов~~
 - g. ~~Общую цену, учитывая количество гостей~~

На экране должна быть предусмотрена кнопка «Назад» для возврата на главный экран. Кнопка «Далее» для перехода к оформлению заказа.

На данном экране должна быть предусмотрена фильтрация концертов следующими способами:

- Фильтрация по стоимости концерта (интервал по цене) – список найденных концертов, подходящих в введенный интервал.

Также должна работать сортировка по следующим параметрам:

- ~~Самый популярный – концерты должны отсортироваться по рейтингу (наиболее популярные в начале)~~
- Самый ближайший – концерты должны отсортироваться по дате и времени (самые близкие в начале)

Пользователь не должен иметь возможность выбрать концерт, на который не осталось мест.

Получить список ~~рейсов~~ можно используя следующий запрос:

Request	Response
URL: {host}/api/concert Method: GET Query string (GET parameters): - name (Алые паруса) - code (149593) - type - date1 (2020-12-01) - date2 (2020-12-13) - guest (2)	<p>----- Successful -----</p> <p>Status: 200 Content-Type: application/json Body:</p> <pre>{ "data": { "concert": [{ "concert_id": 34, "concert_code": "374853", "name ": "Алые паруса", "date_concert":"2020-12-01", "time_start":"23:00", "time_finish":"02:00", "duration":"3 часа", "price":"2400 р.", }, { "concert_id": 35, "concert_code": "438745", "name ": "Оркестр", "date_concert":"2020-12-01", "time_start":"04:00", "time_finish":"06:00", "duration":"2 часа", "price":"2400 р.", }] } }</pre>

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

3. Экран оформления заказа – на данном экране пользователь должен видеть информацию о выбранном концерте, а также иметь возможность добавить гостей.

Если пользователь авторизован, то в заказе изначально должен быть указан авторизованный гость.

Добавить гостя можно нажав на кнопку “Добавить гостя”. Для каждого гостя необходимо заполнить следующие поля:

- Имя
- Фамилия
- Дата рождения
- Номер документа

Также должна быть возможность удалить гостя, но нельзя удалить гостя, если он единственный. Также возможность вернуться на предыдущий экран.

Для подтверждения заказа необходимо нажать на кнопку “Подтвердить”. После этого пользователь должен быть перенаправлен на страницу управления заказом.

Для оформления заказа используйте следующий запрос:

Request	Response
<p>URL: {host}/api/order</p> <p>Method: POST</p> <p>Body:</p> <pre>{ "concert": { "id": 1, "date": "2020-12-20" }, }</pre>	<p>----- Successful -----</p> <p>Status: 201</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "data": { "code": "QSASE" } }</pre>

<pre> "guest": [{ "first_name": "Ivan", "last_name": "Ivanov", "birth_date": "1990-02-20", "document_number": "1234567890" }, { "first_name": "Ivan", "last_name": "Gorbunov", "birth_date": "1990-03-20", "document_number": "1224567890" }] </pre>	<pre> } ----- Validation error ----- Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Validation error", "errors": { <key>: <массив ошибок> } } } </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4. Экран управления заказом – на данном экране необходимо отобразить всю имеющуюся информацию о заказе:
- Информация о заказе
 - Код заказа
 - Стоимость заказа
 - Информация о концерте
 - Номер концерта
 - Название концерта
 - Место проведения концерта
 - Дата и время проведения концерта
 - Длительность концерта
 - Информация о гостях
 - Имя
 - Фамилия
 - Дата рождения
 - Номер документа
 - Место
 - Также на этом экране присутствует кнопка для выбора места “Select seats”, при нажатии на которую пользователь должен перейти на экран с выбором места для пассажиров в бронировании.

Получить информацию о бронировании можно используя следующий запрос:

Request	Response
URL: {host}/api/order/{code} Method: GET	Status: 200 Content-Type: application/json Body: <pre>{</pre>

	<pre> "data": { "code": "AKIJF", "cost": 40000, "tickets": [{ "tickets_id": 34, "tickets_code": "374853", "name_concert": "Алые паруса", "date_concert": "2020-12-01", "time_start": "23:00", "time_finish": "02:00", "duration": "3 часа", "venue": "Большой театр России", }] "guest": [{ "id": 1, "first_name": "Ivan", "last_name": "Ivanov", "birth_date": "1990-02-20", "document_number": "1234567890", "place_from": "7B", "place_back": null }, { "id": 2, "first_name": "Ivan", "last_name": "Larin", "birth_date": "1990-03-20", "document_number": "1224567890", "place_from": null, "place_back": null }] } </pre>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5. Экран с выбором места – на данном экране должна быть представлена схема зала и список гостей. Должна быть возможность выбрать гостя, а после для него место. При нажатии на кнопку “Назад” экран должен смениться на предыдущий без сохранения выбранных мест. При нажатии на кнопку “Сохранить” информация о выбранных местах должна быть сохранена и пользователь должен быть перенаправлен на предыдущую страницу.

Для сохранения выбранного места нужно использовать следующий запрос:

Request	Response
URL: {host}/api/order/{code}/seat Method: PATCH Headers - Content-Type: application/json Body: { "guest_id": 1, "seat": "7 row 56 seat", }	----- Success ----- Status: 200 Content-Type: application/json Body: { "data": { "id": 1, "first_name": "Ivan", "last_name": "Ivanov", "birth_date": "1990-02-20", "document_number": "1234567890", "place_from": "7 row 56 seat", } } ----- Seat is occupied ----- Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Seat is occupied", } } ----- Forbidden ----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Guest does not apply to booking" } } ----- Validation error ----- Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Validation error", } }

	<pre> "errors": { <key>: <массив ошибок> } } } </pre>
--	---------------------------------------------------------------------

6. Регистрация – неавторизованный пользователь должен иметь возможность зарегистрироваться в системе на странице с регистрацией.

Для регистрации пользователя предусмотрен следующий запрос:

Request	Response
URL: {host}/api/register Method: POST Headers - Content-Type: application/json Body: <pre> { "first_name": "Ivan", "last_name": "Ivanov", "phone": "89001234567", "document_number": "7567999222", "password": "paSSword" } </pre>	<pre> ----- Successful ----- Status: 204 ----- Validation error ----- Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Validation error", "errors": { <key>: <массив ошибок> } } } </pre>

7. Вход в личный кабинет – неавторизованный пользователь должен иметь возможность войти в систему на странице со входом.

Для аутентификации пользователя предусмотрен следующий запрос:

Request	Response
URL: {host}/api/login Method: POST Headers - Content-Type: application/json Body: <pre> { "phone": "89001234567", "password": "paSSword" } </pre>	<pre> ----- Successful ----- Status: 200 Content-Type: application/json Body: { "data": { "token": <сгенерированный token> } } </pre>

}	<p>----- Validation error -----</p> <p>Status: 422 Content-Type: application/json Body:</p> <pre>{ "error": { "code": 422, "message": "Validation error", "errors": { <key>: <массив ошибок> } } }</pre> <p>----- Unauthorized -----</p> <p>Status: 401 Content-Type: application/json Body:</p> <pre>{ "error": { "code": 401, "message": "Unauthorized", "errors": { "phone": ["phone or password incorrect"] } } }</pre>
---	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

8. Личный кабинет – на странице с профилем пользователь должен увидеть следующую информацию:
- a. О себе
 - i. Имя
 - ii. Фамилия
 - iii. Количество побывавших концертов
 - b. Избранные концерты. Каждый концерт должен отображать следующие данные:
 - i. Название концерта
 - ii. Код концерта
 - iii. Дата и время концерта
 - iv. Тип концерта
 - v. Место проведения концерта
 - vi. Кнопка “бронировать сейчас” – при нажатии на эту кнопку экран должен смениться на “Экран заказа” для оформления нового заказа. На экране заказа автоматически должен быть заполнен авторизованный гость, а в блоке с концертами должна быть возможность выбрать дату для концерта.
 - vii. Кнопка “Удалить из избранного” – при нажатии на эту кнопку концерт должен быть удален из избранного.

- с. Предстоящие заказы. Каждый заказ должен содержать следующую информацию:
- i. Название концерта
 - ii. Код заказа
 - iii. Дата и время концерта
 - iv. Тип концерта
 - v. Место проведения концерта

При нажатии на код заказа пользователь должен перейти на страницу управления заказом.

Для получения информации о пользователе вы можете использовать следующий запрос:

Request	Response
URL: {host}/user Method: GET Headers - Content-Type: application/json - Authorization: Bearer {token}	<p>----- Success -----</p> <p>Status: 200 Content-Type: application/json Body:</p> <pre>{ "first_name": "Ivan", "last_name": "Ivanov", "phone": "89001234567", "document_number": "1224567890" }</pre> <p>----- Unauthorized -----</p> <p>Status: 401 Content-Type: application/json Body:</p> <pre>{ "error": { "code": 401, "message": "Unauthorized" } }</pre>

Для получения информации о заказах пользователя вы можете использовать следующий запрос:

Request	Response
URL: {host}/user/booking Method: GET Headers - Content-Type: application/json - Authorization: Bearer {token}	<p>----- Success -----</p> <p>Status: 200 Content-Type: application/json Body:</p> <pre>Sheremetievo, { "data": { "items": [{</pre>

```
"code": "MGERS",
"cost": 40000,
"tickets": [
  {
    "tickets_id": 34,
    "tickets_code": "374853",
    "name_concert": "Алые паруса",
    "date_concert": "2020-12-01",
    "time_start": "23:00",
    "time_finish": "02:00",
    "duration": "3 часа",
    "price": "2400 р.",
    "bonuses": "120"
  }
],
"guest": [
  {
    "id": 1,
    "first_name": "Ivan",
    "last_name": "Ivanov",
    "birth_date": "1990-02-20",
    "document_number": "1234567890",
    "place_from": "7 row 55 seat",
  },
  {
    "id": 2,
    "first_name": "Ivan",
    "last_name": "Sergeev",
    "birth_date": "1990-03-20",
    "document_number": "1224567890",
    "place_from": "7 row 56 seat",
  }
]
]
```

----- Unauthorized -----

Status: 401

Content-Type: application/json

Body:

```
{
  "error": {
    "code": 401,
    "message": "Unauthorized"
  }
}
```

9. Должна быть возможность выйти из личного кабинета.
10. Необходимо позаботиться об уведомлении пользователей о каких-либо действиях (ошибки валидации, подтверждения и т.п).

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Разработанное приложение должно быть доступно по адресу <http://xxxxxx-m3.wsr.ru/>, где xxxxxx - логин участника (указан на индивидуальной карточке).

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	2,00
D	Верстка	1,00
E	Программирование на стороне клиента	15,00
F	Программирование на стороне сервера	0,00
G	CMS	0,00
Всего		20,00