

ВВЕДЕНИЕ

Технологии этого модуля: Клиентское программирование

Время на выполнение: 3 часа

К вам обратилась компания «Крылья» - новая авиакомпания, предоставляющая услуги пассажирских авиаперевозок. Заказчик предоставляет вам полностью готовую верстку со всеми страницами и рабочее API. Вам необходимо использовать все имеющиеся навыки в клиентской разработке для создания SPA (Single Page Application).

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://wsr2021.irgups.ru/xxxxxxx-m3/>

Ваше SPA должно состоять из следующих экранов:

- Главная
- Регистрация в личном кабинете
- Вход в личный кабинет
- Личный кабинет
- Результаты поиска
- Бронирование
- Управление бронированием
- Выбор места в салоне воздушного судна

Приложение должно обладать следующим функционалом:

1. Домашний экран – на данном экране располагается статическая информация и форма для поиска подходящих рейсов со следующими полями:
 - a. Откуда (From where) – город или аэропорт вылета
 - b. Куда (To where) – город или аэропорт прилета
 - c. Туда (Departing) – дата вылета
 - d. Обратно (Returning) – дата возвращения обратно
 - e. Количество пассажиров (Passengers) – от 1 до 8

При отправке формы пользователю должен отобразиться экран с найденными рейсами. Получить список аэропортов по введенном запросу можно с помощью следующего запроса:

Request	Response
URL: {host}/api/airport Method: GET Query string (GET parameters): - query	<p>----- Successful -----</p> <p>Status: 200 Content-Type: application/json Body:</p> <pre>{ "data": { "items": [{ "name": "Иркутск", "iata": "IKT" }] } }</pre>

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

2. Экран с найденными рейсами – на данном экране пользователю должен предоставляться список найденных рейсов в зависимости от указанных данных, а также город отправления и прибытия. Каждый рейс должен содержать:
- a. Номер рейса (Flight)
 - b. Самолет (Aircraft)
 - c. Дата и время отправления (Date and time of departure)
 - d. Время прибытия (Arrival time)
 - e. Время в полете (Flight time)
 - f. Общую цену, учитывая количество пассажиров (Cost)

Если пользователь не указал дату возвращения обратно (Returning), то список рейсов должен включать в себя только перелеты из точки А в точку В. Если пользователь указал дату возвращения обратно (Returning), то список рейсов должен включать в себя и перелеты из точки А в В и из В в А на указанные даты.

Пользователь должен выбрать 1 рейс туда и 1 обратно (если была указана обратная дата), которые ему подходят и нажать на кнопку “Go to booking” для перехода на экран с оформлением бронирования.

На данном экране должна быть предусмотрена фильтрация рейсов следующими способами:

- Фильтрация по времени вылета – должны отображаться рейсы, время вылета которых находится внутри выбранного диапазона.

Также должна работать сортировка по следующим параметрам:

- Самый быстрый — рейсы должны сортироваться по времени в пути (наиболее быстрые в начале)
- Самый дешевый – рейсы должны сортироваться по стоимости (наиболее дешевые в начале)

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

Получить список рейсов можно используя следующий запрос:

Request	Response
<p>URL: {host}/api/flight Method: GET</p> <p>Query string (GET parameters):</p> <ul style="list-style-type: none">- from (<i>IKT</i>)- to (<i>BTK</i>)- date1 (<i>2020-10-01</i>)- date2 (<i>2020-10-13</i>)- passengers (<i>1</i>)	<p>----- Successful -----</p> <p>Status: 200 Content-Type: application/json Body:</p> <pre>{ "data": { "flights_to": [{ "flight_id": 2, "flight_code": "FP1200", "from": { "city": "Братск", "airport": "Братск", "iata": "BTK", "date": "2020-10-01", "time": "12:00" }, "to": { "city": "Иркутск", "airport": "Иркутск", "iata": "IKT", "date": "2020-10-01", "time": "13:35" }, "cost": 9500, "availability": 156 }, { "flight_id": 14, "flight_code": "FP 1201", "from": { "city": "Братск", "airport": "Братск", "iata": "BTK", "date": "2020-10-01", "time": "08:35" }, "to": { "city": "Иркутск", "airport": "Иркутск", "iata": "IKT", "date": "2020-10-01",</pre>

```
        "time": "10:05"
      },
      "cost": 10500,
      "availability": 156
    }
  ],
  "flights_back": [
    {
      "flight_id": 1,
      "flight_code": "FP 2100",
      "from": {
        "city": "Иркутск",
        "airport": "Иркутск",
        "iata": "ИКТ",
        "date": "2020-10-10",
        "time": "08:35"
      },
      "to": {
        "city": "Братск",
        "airport": "Братск",
        "iata": "ВТК",
        "date": "2020-10-10",
        "time": "10:05"
      },
      "cost": 10500,
      "availability": 156
    },
    {
      "flight_id": 13,
      "flight_code": "FP 2101",
      "from": {
        "city": "Иркутск",
        "airport": "Иркутск",
        "iata": "ИКТ",
        "date": "2020-10-10",
        "time": "12:00"
      },
      "to": {
        "city": "Братск",
        "airport": "Братск",
        "iata": "ВТК",
        "date": "2020-10-10",
        "time": "13:35"
      },
      "cost": 12500,
      "availability": 156
    }
  ]
}
```

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

3. Экран оформления бронирования – на данном экране пользователь должен видеть информацию о выбранных рейсах, а также иметь возможность добавить пассажиров.

Если пользователь авторизован, то в бронировании изначально должен быть указан авторизованный пассажир.

Добавить пассажира можно нажав на кнопку “Add passenger”. Для каждого пассажира необходимо заполнить следующие поля:

- a. Имя (First name)
- b. Фамилия (Last name)
- c. Дата рождения (Date of Birth)
- d. Номер документа (Document number)

Также должна быть возможность удалить пассажира, но нельзя удалить пассажира, если он единственный.

Для подтверждения бронирования необходимо нажать на кнопку “Confirm”. После этого пользователь должен быть перенаправлен на страницу управления бронированием.

Для оформления бронирования используйте следующий запрос:

Request	Response
<p>URL: {host}/api/booking Method: POST Body: { "flight_from": { "id": 1, "date": "2020-09-20" }, }</p>	<p>----- Successful ----- Status: 201 Content-Type: application/json Body: { "data": { "code": "QSASE" } }</p>

<pre> "flight_back": { "id": 2, "date": "2020-09-30" }, "passengers": [{ "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 бронированием – на данном экране необходимо отобразить всю имеющуюся информацию о бронировании:
- a. Информация о бронировании
 - i. Код бронирования
 - ii. Стоимость бронирования
 - b. Информация о перелете (для каждого рейса) – обратите внимание, что если бронирование оформлено “туда-обратно”, то рейса должно быть два (туда и обратно):
 - i. Номер рейса (Flight)
 - ii. Самолет (Aircraft)
 - iii. Откуда (From where)
 - iv. Дата и время отправления (Date and time of departure)
 - v. Куда (To where)
 - vi. Время прибытия (Arrival time)
 - vii. Время в полете (Flight time)
 - c. Информация о пассажирах
 - i. Имя (First name)
 - ii. Фамилия (Last name)
 - iii. Дата рождения (Date of Birth)
 - iv. Номер документа (Document number)
 - v. Место (Seat)
 - d. 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
<p>URL: {host}/api/booking/{code} Method: GET</p>	<p>Status: 200 Content-Type: application/json Body:</p> <pre data-bbox="766 392 1212 2038">{ "data": { "code": "AKIJF", "cost": 40000, "flights": [{ "flight_id": 1, "flight_code": "FP2100", "from": { "city": "Иркутск", "airport": "Иркутск", "iata": "ИКТ", "date": "2020-10-01", "time": "08:35" }, "to": { "city": "Братск", "airport": "Братск", "iata": "ВТК", "date": "2020-10-01", "time": "10:05" }, "cost": 10500, "availability": 56 }, { "flight_id": 2, "flight_code": "FP1200", "from": { "city": "Братск", "airport": "Братск", "iata": "ВТК", "date": "2020-10-12", "time": "12:00" }, "to": { "city": "Иркутск", "airport": "Иркутск", "iata": "ИКТ", "date": "2020-10-12", "time": "13:35" }, "cost": 9500, </pre>

	<pre> "availability": 56 }], "passengers": [{ "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 самолета и список пассажиров. Должна быть возможность выбрать пассажира, а после для него место. При нажатии на кнопку “Back” экран должен смениться на предыдущий без сохранения выбранных мест. При нажатии на кнопку “Confirm” 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
<p>URL: {host}/api/booking/{code}/seat</p> <p>Method: PATCH</p> <p>Headers - Content-Type: application/json</p> <p>Body: { "passenger": 1,</p>	<p>----- Success -----</p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body: { "data": { "id": 1, "first_name": "Ivan", "last_name": "Ivanov", "birth_date": "1990-02-20",</p>

<pre>"seat": "7B", "type": "from/back" }</pre>	<pre>"document_number": "1234567890", "place_from": "7B", "place_back": null } }</pre> <p>----- Seat is occupied -----</p> <p>Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Seat is occupied", } }</p> <p>----- Forbidden -----</p> <p>Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Passenger does not apply to booking" } }</p> <p>----- Validation error -----</p> <p>Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Validation error", "errors": { <key>: <массив ошибок> } } }</p>
--	--

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	----- Successful ----- Status: 204

Method: POST Headers - Content-Type: application/json Body: <pre>{ "first_name": "Ivan", "last_name": "Ivanov", "phone": "89001234567", "document_number": "7567999222", "password": "paSSword" }</pre>	<p style="text-align: center;">----- Validation error -----</p> Status: 422 Content-Type: application/json Body: <pre>{ "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>	<p style="text-align: center;">----- Successful -----</p> Status: 200 Content-Type: application/json Body: <pre>{ "data": { "token": <сгенерированный token> } }</pre> <p style="text-align: center;">----- Validation error -----</p> Status: 422 Content-Type: application/json Body: <pre>{ "error": { "code": 422, "message": "Validation error", "errors": { <key>: <массив ошибок> } } }</pre> <p style="text-align: center;">----- Unauthorized -----</p> Status: 401

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

8. Личный кабинет – на странице с профилем пользователь должен увидеть следующую информацию:
- a. О себе
 - i. Имя
 - ii. Фамилия
 - iii. Количество прошедших полетов
 - b. Избранные рейсы. Каждый рейс должен отображать следующие данные:
 - i. Город вылета
 - ii. Город прибытия
 - iii. Время отправления
 - iv. Время прибытия
 - v. Кнопка "Book now" – при нажатии на эту кнопку экран должен смениться на "Экран бронирования" для оформления нового бронирования. На экране бронирования автоматически должен быть заполнен авторизованный пассажир, а в блоке с рейсами должна быть возможность выбрать дату для рейса.
 - vi. Кнопка "Remove from favorites" – при нажатии на эту кнопку рейс должен быть удален из избранного.
 - c. Прошедшие рейсы. Каждый рейс должен отображать следующие данные:
 - i. Номер рейса
 - ii. Город вылета
 - iii. Город прибытия
 - iv. Время отправления
 - v. Время прибытия
 - vi. Кнопка "Add to favorites" – при нажатии на эту кнопку рейс должен быть добавлен в избранное.
 - d. Предстоящие бронирования. Каждое бронирование должно содержать следующую информацию:
 - i. Код бронирования
 - ii. Дата вылета
 - iii. Время вылета
 - iv. Время прилета
 - v. Откуда
 - vi. Куда

При нажатии на код бронирования пользователь должен перейти на страницу управления бронированием.

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}	----- Success ----- Status: 200 Content-Type: application/json Body: { "first_name": "Ivan", "last_name": "Ivanov", "phone": "89001234567", "document_number": "1224567890" } ----- Unauthorized ----- Status: 401 Content-Type: application/json Body: { "error": { "code": 401, "message": "Unauthorized" } }

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

Request	Response
URL: {host}/user/booking Method: GET Headers - Content-Type: application/json - Authorization: Bearer {token}	----- Success ----- Status: 200 Content-Type: application/json Body: { "data": { "items": [{ "code": "MGERS", "cost": 40000, "flights": [{ "flight_id": 1, "flight_code": "FP2100", }] }] } }

```
"from": {
  "city": "Иркутск",
  "airport": "Иркутск",
  "iata": "ИКТ",
  "date": "2020-10-01",
  "time": "08:35"
},
"to": {
  "city": "Братск",
  "airport": "Братск",
  "iata": "ВТК",
  "date": "2020-10-01",
  "time": "10:05"
},
"cost": 10500,
"availability": 58
},
{
  "flight_id": 2,
  "flight_code": "FP1200",
  "from": {
    "city": "Братск",
    "airport": "Братск",
    "iata": "ВТК",
    "date": "2020-10-12",
    "time": "12:00"
  },
  "to": {
    "city": "Иркутск",
    "airport": "Иркутск",
    "iata": "ИКТ",
    "date": "2020-10-12",
    "time": "13:35"
  },
  "cost": 9500,
  "availability": 58
}
],
"passengers": [
  {
    "id": 1,
    "first_name": "Ivan",
    "last_name": "Ivanov",
    "birth_date": "1990-02-20",
    "document_number": "1234567890",
    "place_from": null,
    "place_back": null
  },
  ],
```

