

Задание 8

- Спроектировать API-методы для сервиса туроператора.
- Для каждого метода: конечная точка, REST-метод, контракт (входные и выходные параметры, заголовки), ожидаемый HTTP-статус ответа, обработка ошибок.
- Удобнее всего – взять образец на слайде 7 из видео-лекции и сделать сначала список необходимых методов для покрытия use-cases, и далее описать контракты для каждого метода.

Решение

- API использует REST-принципы.
- Базовый URL: /api/v1
- Аутентификация: JWT-токен в заголовке Authorization: Bearer <token>
- Для публичных операций (поиск) аутентификация не требуется.

Сопоставление use-cases, маршрутов, методов и аутентификации:

Действующее лицо	Use Case	Маршрут	HTTP-метод	Аутентификация
Клиент	UC1.1 Поиск доступных рейсов/туров	/flights	GET	Нет
Клиент	UC1.2 Создание бронирования	/bookings	POST	Да
Клиент	UC1.3 Просмотр деталей бронирования	/bookings/{id}	GET	Да
Клиент	UC1.4 Оплата бронирования	/payments	POST	Да

Клиент	UC1.5 Получение билета и подтверждения	/tickets/{id}	GET	Да
Клиент	UC1.6 Отмена бронирования	/bookings/{id}	DELETE	Да
Менеджер	UC2.1 Добавление рейса	/flights	POST	Да
Менеджер	UC2.2 Обновление рейса	/flights/{id}	PUT	Да
Менеджер	UC2.3 Удаление рейса	/flights/{id}	DELETE	Да
Менеджер	UC2.4 Управление уведомлениями	/notifications	POST	Да
Клиент/Менеджер	UC3.1 Регистрация пользователя	/users/register	POST	Нет
Клиент/Менеджер	UC3.2 Авторизация	/users/login	POST	Нет
Клиент/Менеджер	UC3.3 Просмотр профиля	/users/{id}	GET	Да

Схемы запросов и ответов:

- Авторизация и пользователи

- POST /users/register

- Вход:

JSON body:

```
{
  "name": "string (обяз.)",
  "email": "string (обяз.)",
```

```
"password": "string (обяз.)",
"phone": "string",
"passportData": "string",
"type": "Client или Manager"
}
```

■ Заголовки:

Content-Type: application/json

■ Выход:

```
{
  "userId": "UUID",
  "message": "User registered"
}
Статус: 201 Created
```

■ Ошибки:

- 400 Bad Request – некорректные данные
- 409 Conflict – email существует

○ POST /users/login

■ Вход:

JSON body:

```
{
  "email": "string (обяз.)",
  "password": "string (обяз.)"
}
```

■ Заголовки:

Content-Type: application/json

■ Выход:

```
{
  "token": "string (JWT)",
  "userId": "UUID"
}
Статус: 200 OK
```

■ Ошибки:

- 401 Unauthorized – неверные credentials

○ GET /users/{id}

■ Вход:

Path param id: UUID

■ Заголовки:

Authorization: Bearer <token>

■ Выход:

```
{  
    "name": "string",  
    "email": "string",  
    "phone": "string",  
    "type": "string",  
    "orderHistory": "string"  
}  
Статус: 200 OK
```

■ Ошибки:

- 401 Unauthorized
- 404 Not Found – пользователь не найден
- Рейсы и туры
 - GET /flights

■ Вход:

Query params:

direction: string

departureDate: date

arrivalDate: date

class: string

seats: integer

■ Заголовки: Нет

■ Выход:

```
[  
{
```

```
        "flightId": "UUID",
        "airline": "string",
        "departureTime": "datetime",
        "arrivalTime": "datetime",
        "price": "decimal",
        "availableSeats": "integer"
    }
]
```

Статус: 200 OK

■ Ошибки:

- 400 Bad Request – некорректные параметры
 - 404 Not Found – рейсы не найдены
- POST /flights (для менеджера)

■ Вход:

JSON body:

```
{
    "airline": "string (обяз.)",
    "departureTime": "datetime (обяз.)",
    "arrivalTime": "datetime (обяз.)",
    "direction": "string (обяз.)",
    "availableSeats": "integer (обяз.)",
    "price": "decimal (обяз.)",
    "class": "string"
}
```

■ Заголовки:

- Authorization: Bearer <token>
- Content-Type: application/json

■ Выход:

```
{
    "flightId": "UUID",
    "message": "Flight created"
}
```

Статус: 201 Created

■ Ошибки:

- 401 Unauthorized
- 403 Forbidden – не менеджер

- 400 Bad Request
- PUT /flights/{id} (для менеджера)
 - Вход:
 - Path param id: UUID
 - JSON body (частичные обновления, как в POST)
 - Заголовки:
 - Authorization: Bearer <token>
 - Content-Type: application/json
 - Выход:

```
{  
    "message": "Flight updated"  
}
```

Статус: 200 OK
 - Ошибки:
 - 401 Unauthorized
 - 403 Forbidden
 - 404 Not Found
 - 400 Bad Request
 - DELETE /flights/{id} (для менеджера)
 - Вход:
 - Path param id: UUID
 - Заголовки:
 - Authorization: Bearer <token>
 - Выход:

```
{  
    "message": "Flight deleted"  
}
```

Статус: 200 OK

■ Ошибки:

- 401 Unauthorized
- 403 Forbidden
- 404 Not Found

● Бронирования

- POST /bookings

■ Вход:

JSON body:

```
{  
    "tourId": "UUID (обяз.)",  
    "passengers": [  
        {  
            "name": "string (обяз.)",  
            "passportData": "string (обяз.)",  
            "age": "integer",  
            "contact": "string"  
        }  
    ],  
    "class": "string",  
    "additionalServices": "string (JSON)"  
}
```

■ Заголовки:

- Authorization: Bearer <token>
- Content-Type: application/json

■ Выход:

```
{  
    "bookingId": "UUID",  
    "status": "pending"  
}
```

Статус: 201 Created

■ Ошибки:

- 401 Unauthorized
- 400 Bad Request – недостаточно мест

- 404 Not Found – тур не найден
- GET /bookings/{id}

■ Вход:

- Path param id: UUID

■ Заголовки:

- Authorization: Bearer <token>

■ Выход:

```
{  
    "bookingDate": "datetime",  
    "status": "string",  
    "confirmation": "string",  
    "passengers": [],  
    "paymentId": "UUID"  
}
```

Статус: 200 OK

■ Ошибки:

- 401 Unauthorized
- 404 Not Found

- DELETE /bookings/{id}

■ Вход:

- Path param id: UUID

■ Заголовки:

- Authorization: Bearer <token>

■ Выход:

```
{  
    "message": "Booking cancelled"  
}
```

Статус: 200 OK

■ Ошибки:

- 401 Unauthorized

- 404 Not Found
 - 409 Conflict – нельзя отменить подтвержденное
- Оплата
 - POST /payments
 - Вход:
JSON body:

```
{  
    "bookingId": "UUID (обяз.)",  
    "amount": "decimal (обяз.)",  
    "method": "string (обяз.)"  
}
```
 - Заголовки:
 - Authorization: Bearer <token>
 - Content-Type: application/json
 - Выход:

```
{  
    "paymentId": "UUID",  
    "status": "success"  
}
```

(интеграция с шлюзом)
Статус: 201 Created
 - Ошибки:
 - 401 Unauthorized
 - 400 Bad Request – недостаточно средств
 - 404 Not Found – бронь не найдена
 - Билеты
 - GET /tickets/{id}
 - Вход:
 - Path param id: UUID (ticketId)
 - Заголовки:

- Authorization: Bearer <token>

■ Вход:

```
{  
    "flightDetails": "string",  
    "passengerData": "string",  
    "eDocument": "string",  
    "status": "string"  
}  
Статус: 200 OK
```

■ Ошибки:

- 401 Unauthorized
- 404 Not Found

● Уведомления

○ POST /notifications (для менеджера)

■ Вход:

JSON body:

```
{  
    "userId": "UUID (обяз.)",  
    "bookingId": "UUID",  
    "type": "string (обяз.)",  
    "message": "string"  
}
```

■ Заголовки:

- Authorization: Bearer <token>
- Content-Type: application/json

■ Выход:

```
{  
    "notificationId": "UUID"  
}  
Статус: 201 Created
```

■ Ошибки:

- 401 Unauthorized
- 403 Forbidden

- 400 Bad Request

Обязательные поля и статус-коды ошибок:

- Обязательные поля:
 - Указаны в схемах (например, email и password для login).
- Валидация на backend:
 - email – валидный формат
 - password – минимум 8 символов
 - даты – ISO формат
- Статус-коды ошибок:

Код	Описание
200 OK	Успех для GET/PUT/DELETE
201 Created	Успех для POST
400 Bad Request	Некорректные данные/параметры (детали в JSON { "error": "description" })
401 Unauthorized	Нет/недействительный токен
403 Forbidden	Недостаточно прав (например, клиент пытается добавить рейс)
404 Not Found	Ресурс не существует
409 Conflict	Конфликт (например, email занят, места закончились)
500 Internal Server Error	Внутренняя ошибка (логировать, не раскрывать детали)