



# HOMEWORK #2.1

ОТКРЫТАЯ ШКОЛА  
СИСТЕМНЫХ АНАЛИТИКОВ



[CLICK HERE](#)



# TASKS

**Создайте спецификацию со следующими методами:**

- Метод для получения описания выбранного отеля.  
В результате ответа вы должны получить:
  - идентификатор отеля;
  - полное наименование отеля;
  - изображение отеля (URL);
  - полное описание отеля;
  - обратная связь для гостей;
  - список доступных номеров.
- Метод для заказа номера в отеле.





# TASK #1.1

Метод для получения описания выбранного отеля.

Swagger UI

## Просмотр информации по выбранному отелю ^

GET

/hotels/{hotelID} Получить информацию об определенном отеле

^

Возвращает полную информацию об отеле по его уникальному идентификатору

### Parameters

Try it out

Name	Description
------	-------------

**hotelID** \* required Идентификатор выбранного отеля

integer(\$int64)  
(path) Default value : 1

Example : 75

75



# TASK #1.2

Метод для получения описания выбранного отеля.

## JSON OpenAPI Specification

```
"/hotels/{hotelID}": {
    "get": {
        Scan | Audit
        "parameters": [
            {
                "$ref": "#/components/parameters/hotelID"
            }
        ],
        "summary": "Получить информацию об определенном отеле",
        "description": "Возвращает полную информацию об отеле по его уникальному идентификатору",
        "operationId": "getHotelByID",
        "tags": [
            "Просмотр информации по выбранному отелю"
        ],
        "responses": {
            "200": {
                "description": "Информация об отеле успешно получена",
                "content": {
                    "application/json": {
                        "schema": {
                            "$ref": "#/components/schemas/Hotel"
                        }
                    }
                }
            },
            "400": { ... },
            "404": { ... },
            "500": { ... },
            "503": { ... }
        }
    }
},
```



# TASK #1.3

Метод для заказа номера в отеле.

Swagger UI

## Бронирование номера в отеле

**GET** /orders Получить список бронирований

**POST** /orders Забронировать номер

Забронировать выбранный номер в отеле

**Parameters**

No parameters

**Request body** required

application/json

**Examples:**

Создание бронирования

Example Value | Schema

```
{  
    "roomID": 615,  
    "clientID": 362,  
    "dateTime": "2025-01-15T14:00:00Z"  
}
```



# TASK #1.4

Метод для получения описания выбранного отеля.

## JSON OpenAPI Specification

```
"/orders": {
    "get": { ... },
    "post": {
        Scan | Audit
        "summary": "Забронировать номер",
        "description": "Забронировать выбранный номер в отеле",
        "tags": [
            "Бронирование номера в отеле"
        ],
        "operationId": "createOrder",
        "requestBody": {
            "required": true,
            "content": {
                "application/json": {
                    "schema": {
                        "$ref": "#/components/schemas/Order"
                    },
                    "examples": {
                        "newOrder": {
                            "summary": "Создание бронирования",
                            "value": {
                                "roomID": 615,
                                "clientID": 362,
                                "dateTime": "2025-01-15T14:00:00Z"
                            }
                        }
                    }
                }
            }
        }
    }
},
```



## TASK #2.1

В рамках данного домашнего задания была предложена готовая API спецификация (документ для самопроверки). При проведении финальной сверки разработанной API спецификации с представленным эталонным документом были выявлены следующие различия:

### Убран параметр nullable при описании схем

Согласно стандарту OpenAPI все поля по умолчанию трактуются как обязательные.

### Уточнены наименования endpoint'ов и параметров

Исправления обеспечивают общепринятые соглашения RestAPI о наименованиях.

**Изменения:** /Orders → /orders, id → hotelID.

### Детализированы параметры запроса

Для критически важных параметров добавлены ограничения и значения по умолчанию, что усиливает валидацию данных и улучшает документацию.

### Добавлена полная обработка ошибок

Спецификация теперь явно декларирует ответы для всех предусмотренных сценариев (коды 400, 404, 409, 500, 503 и др.), что позволяет корректно реагировать на любые ситуации.

Помимо этого уточнена универсальная схема Error.





## TASK #2.2

В рамках данного домашнего задания была предложена готовая API спецификация (документ для самопроверки). При проведении финальной сверки разработанной API спецификации с представленным эталонным документом были выявлены следующие различия:

### Добавлены многочисленные примеры

Для запросов и ответов добавлены наглядные примеры, что значительно упрощает интеграцию и снижает количество ошибок при разработке.

### Реализована полная поддержка CRUD операций

API теперь покрывает весь жизненный цикл сущностей (Create, Read, Update, Delete), предоставляя клиентам полный набор функций для управления данными



### Доступ к разработанной API спецификации

Для получения доступа к разработанной API спецификации можно воспользоваться:

- Google Drive: [GoogleDrive](#).
- GitHub Repository: [GitHub](#).
- Приложенный JSON файл.



**СПАСИБО  
ЗА ВНИМАНИЕ  
И ВАШ ТРУД**

**CLICK HERE**