

Липцкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №2

по Технологии разработки ПО

Проектирование API

Студент

Лобов М.Ю.

Группа АИ-18

Руководитель

Назаркин О.А.

Доцент

Липецк 2021 г.

Задание кафедры

Выбрать предметную область и для данной предметной области спроектировать методы взаимодействия с API с помощью сервиса Swagger.

Ход работы

В качестве приложения был выбран интернет-магазин вина «World of Wine». Для проектирования API был использован сервис SwaggerHub.

Полученный файл world_of_wine.yaml имеет следующее содержание:

```
openapi: 3.0.0
info:
  title: 'world of wine'
  version: '1.0'

paths:
  /wines:
    post:
      description: 'Add wines'
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/winesPostRequest'
      responses:
        '201':
          description: 'Created'
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/winesIds'
        '400':
          description: 'Bad request'
          content:
            application/json:
              schema:
                type: object
                additionalProperties: false
                properties:
                  validation_error:
                    $ref: '#/components/schemas/winesIdsAP'
                required:
                  - validation_error
```

```

/wines/{wine_id}:
  parameters:
    - in: path
      name: wine_id
      required: true
      schema:
        type: integer
  get:
    description: 'Get wine info'
    responses:
      '200':
        description: 'Success'
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/wineGetResponse'
      '404':
        description: 'Not found'

  patch:
    description: 'Update wine by id'
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/wineUpdateRequest'
    responses:
      '200':
        description: 'Created'
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/wineItem'
      '400':
        description: 'Bad request'
      '404':
        description: 'Not found'

/orders:

```

```

post:
  description: 'Add orders'
  requestBody:
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/OrdersPostRequest'
  responses:
    '201':
      description: 'Created'
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/OrdersIds'
    '400':
      description: 'Bad request'
      content:
        application/json:
          schema:
            type: object
            additionalProperties: false
            properties:
              validation_error:
                $ref: '#/components/schemas/OrdersIdsAP'
            required:
              - validation_error

/orders/assign:
  post:
    description: 'Assign orders to a wine by id'
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/OrdersAssignPostRequest'
    responses:
      '200':
        description: 'OK'
        content:

```

```

        application/json:
          schema:
            allOf:
              - $ref: '#/components/schemas/OrdersIds'
              - $ref: '#/components/schemas/AssignTime'
      '400':
        description: 'Bad request'

/orders/complete:
  post:
    description: 'Marks orders as completed'
    requestBody:
      content:
        application/json:
          schema:
            $ref:
'#/components/schemas/OrdersCompletePostRequest'
    responses:
      '200':
        description: 'OK'
        content:
          application/json:
            schema:
              $ref:
'#/components/schemas/OrdersCompletePostResponse'
      '400':
        description: 'Bad request'

components:
  schemas:
    winesPostRequest:
      type: object
      additionalProperties: false
      properties:
        data:
          type: array
          items:
            $ref: '#/components/schemas/wineItem'
      required:

```

- data

wineItem:

- type: object

- additionalProperties: false

- properties:

 - wine_id:

 - type: integer

 - wine_color:

 - type: string

 - enum:

 - white

 - red

 - rose

 - wine_is_sparkling:

 - type: boolean

 - wine_country:

 - type: string

 - enum:

 - Spain

 - France

 - Georgia

 - Russia

 - wine_alcohol:

 - type: integer

- required:

 - wine_id

 - wine_color

 - wine_is_sparkling

 - wine_country

 - wine_alcohol

winesIds:

- type: object

- additionalProperties: false

- properties:

 - wines:

 - type: array

 - items:

```

        type: object
        additionalProperties: false
        properties:
            id:
                type: integer
        required:
            - id
    required:
        - wines

winesIdsAP:
    type: object
    additionalProperties: true
    properties:
        wines:
            type: array
            items:
                type: object
                additionalProperties: true
                properties:
                    id:
                        type: integer
                required:
                    - id
    required:
        - wines

wineGetResponse:
    type: object
    additionalProperties: false
    properties:
        wine_id:
            type: integer
        wine_color:
            type: string
            enum:
                - white
                - red
                - rose

```



```

    wine_is_sparkling:
      type: boolean
    wine_country:
      type: string
      enum:
        - Spain
        - France
        - Georgia
        - Russia
    wine_alcohol:
      type: integer
    rating:
      type: number
  required:
    - wine_id
    - wine_type
    - addresss
    - working_hours
    - earnings

wineUpdateRequest:
  type: object
  additionalProperties: false
  properties:
    wine_color:
      type: string
      enum:
        - white
        - red
        - rose
    wine_is_sparkling:
      type: boolean
    wine_country:
      type: string
      enum:
        - Spain
        - France
        - Georgia
        - Russia

```

```
wine_alcohol:
  type: integer
```

OrdersPostRequest:

```
  type: object
  additionalProperties: false
  properties:
    data:
      type: array
      items:
        $ref: '#/components/schemas/OrderItem'
  required:
    - data
```

OrderItem:

```
  type: object
  additionalProperties: false
  properties:
    order_id:
      type: integer
    amount:
      type: number
    address:
      type: string
  required:
    - order_id
    - amount
    - address
    - delivery_hours
```

OrdersIds:

```
  type: object
  additionalProperties: false
  properties:
    orders:
      type: array
      items:
        type: object
        additionalProperties: false
```

```
        properties:
          id:
            type: integer
        required:
          - id
    required:
      - orders
```

OrdersIdsAP:

```
    type: object
    additionalProperties: true
    properties:
      orders:
        type: array
        items:
          type: object
          additionalProperties: true
          properties:
            id:
              type: integer
          required:
            - id
    required:
      - orders
```

AssignTime:

```
    type: object
    additionalProperties: false
    properties:
      assign_time:
        type: string
        example: '2021-01-10T09:32:14.42Z'
```

OrdersAssignPostRequest:

```
    type: object
    additionalProperties: false
    properties:
      wine_id:
        type: integer
```

```
required:
  - wine_id
```

OrdersCompletePostRequest:

```
type: object
additionalProperties: false
properties:
  wine_id:
    type: integer
  order_id:
    type: integer
  complete_time:
    type: string
    example: '2021-01-10T10:33:01.42Z'
required:
  - wine_id
  - order_id
  - complete_time
```

OrdersCompletePostResponse:

```
type: object
additionalProperties: false
properties:
  order_id:
    type: integer
required:
  - order_id
```

Вывод

В ходе выполнения лабораторной работы были получены навыки описания API приложения с помощью сервиса SwaggerHub.