

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа №4

Выполнил:
Куцало Александр
Группа
М3221d

Проверил:
Добряков Д. И.

Санкт-Петербург

Задача

Создать документацию ЛР2 средствами Swagger

Ход работы

Для создания документации был создан файл swagger.json:

```
{
  "openapi": "3.0.0",
  "info": {
    "title": "User API",
    "description": "API for managing users",
    "version": "1.0.0"
  },
  "tags": [
    {
      "name": "User",
      "description": "Operations related to users creation and usage"
    },
    {
      "name": "Shop",
      "description": "Operations related to the shop"
    }
  ],
  "paths": {
    "/users/id/{id}": {
      "get": {
        "summary": "Get user by ID",
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "number"
            }
          }
        ]
      }
    },
    "/users": {
      "get": {
        "summary": "Get all users",
        "parameters": [
          {
            "name": "page",
            "in": "query",
            "required": false,
            "schema": {
              "type": "number"
            }
          }
        ]
      },
      "post": {
        "summary": "Create new user",
        "parameters": [
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": {
              "$ref": "#/components/schemas/User"
            }
          }
        ]
      }
    },
    "/users/{id}": {
      "put": {
        "summary": "Update user",
        "parameters": [
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": {
              "$ref": "#/components/schemas/User"
            }
          }
        ]
      },
      "delete": {
        "summary": "Delete user",
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "number"
            }
          }
        ]
      }
    }
  },
  "tags": ["User"],
  "responses": {
    "200": {
      "description": "Success",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/User"
          }
        }
      }
    },
    "404": {
      "description": "User not found"
    }
  }
}
```

```

"/users/create": {
  "post": {
    "summary": "Create a new user",
    "requestBody": {
      "required": true,
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/User"
          }
        }
      }
    },
    "tags": ["User"],
    "responses": {
      "200": {
        "description": "User created successfully"
      },
      "400": {
        "description": "Bad request"
      }
    }
  }
},
"/users/login": {
  "post": {
    "summary": "User login",
    "requestBody": {
      "required": true,
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/LoginCredentials"
          }
        }
      }
    },
    "tags": ["User"],
    "responses": {
      "200": {
        "description": "Login successful"
      },
      "403": {
        "description": "Unauthorized"
      }
    }
  }
},
"/users/auth_only": {
  "get": {
    "summary": "Access private page (requires authentication)",
    "security": [
      {
        "bearerAuth": []
      }
    ],
    "tags": ["User"],
    "responses": {
      "200": {
        "description": "Access granted"
      }
    }
  }
},

```

```

        "403": {
            "description": "Unauthorized"
        }
    },
    "/users/logout": {
        "post": {
            "summary": "Logout",
            "responses": {
                "200": {
                    "description": "Logout successful"
                }
            },
            "tags": ["User"]
        }
    },
    "/shop/": {
        "get": {
            "summary": "Test frontend homepage",
            "responses": {
                "200": {
                    "description": "Success"
                }
            },
            "tags": ["Shop"]
        }
    },
    "/shop/item": {
        "get": {
            "summary": "Get all items",
            "parameters": [
                {
                    "name": "tagId",
                    "in": "query",
                    "description": "Array of tag IDs",
                    "required": false,
                    "schema": {
                        "type": "array",
                        "items": {
                            "type": "integer"
                        }
                    }
                }
            ],
            "tags": ["Shop"],
            "responses": {
                "200": {
                    "description": "Success",
                    "content": {
                        "application/json": {
                            "schema": {
                                "$ref": "#/components/schemas/Item"
                            }
                        }
                    }
                }
            }
        }
    },
    "/shop/item/{id}": {

```

```
"get": {
  "summary": "Get item by ID",
  "parameters": [
    {
      "name": "id",
      "in": "path",
      "required": true,
      "schema": {
        "type": "string"
      }
    }
  ],
  "tags": ["Shop"],
  "responses": {
    "200": {
      "description": "Success",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/Item"
          }
        }
      }
    },
    "404": {
      "description": "Item not found"
    }
  }
},
"post": {
  "summary": "Add item to shopping cart",
  "parameters": [
    {
      "name": "id",
      "in": "path",
      "required": true,
      "schema": {
        "type": "string"
      }
    }
  ],
  "tags": ["Shop"],
  "requestBody": {
    "required": true,
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/AddToCartRequest"
        }
      }
    }
  },
  "security": [
    {
      "bearerAuth": []
    }
  ],
  "responses": {
    "200": {
      "description": "Item added to shopping cart",
      "content": {
```

```

        "application/json": {
            "schema": {
                "$ref": "#/components/schemas/ShoppingCartItem"
            }
        }
    },
    "401": {
        "description": "Unauthorized"
    }
}

"/shop/cart": {
    "get": {
        "summary": "Get shopping cart contents",
        "security": [
            {
                "bearerAuth": []
            }
        ],
        "tags": ["Shop"],
        "responses": {
            "200": {
                "description": "Success"
            },
            "401": {
                "description": "Unauthorized"
            }
        }
    }
}

},
"components": {
    "schemas": {
        "AddToCartRequest": {
            "type": "object",
            "properties": {
                "quantity": {
                    "type": "integer",
                    "example": 12
                },
                "itemId": {
                    "type": "integer",
                    "example": 1
                }
            }
        },
        "User": {
            "type": "object",
            "properties": {
                "name": {
                    "type": "string",
                    "example": "John Doe"
                },
                "password": {
                    "type": "string",
                    "description": "Hashed password",
                    "example":
"$2b$05$qalSs7dD6Vjxi.frJk05UOagBRwvPKxwqMgToFaVZabIl80n.nxwq"
                }
            }
        }
    }
}

```

```
    "email": {
      "type": "string",
      "example": "johndoe@example.com"
    }
  },
  "required": ["email", "password"]
},
"LoginCredentials": {
  "type": "object",
  "properties": {
    "email": {
      "type": "string",
      "example": "example@example.com"
    },
    "password": {
      "type": "string",
      "example": "password123"
    }
  },
  "required": [
    "password",
    "email"
  ]
},
"Item": {
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "example": "Shoes"
    },
    "description": {
      "type": "string",
      "example": "These shoes are really good"
    },
    "price": {
      "type": "number",
      "example": 1000
    },
    "imageUrl": {
      "type": "string",
      "example": "https://example.com/example.png"
    },
    "quantity": {
      "type": "number",
      "example": 12
    }
  },
  "required": [
    "name",
    "price"
  ]
},
"ItemTag": {
  "type": "object",
  "properties": {
    "itemId": {
      "type": "number",
      "example": 1
    },
    "tagId": {
```

```

        "type": "number",
        "example": 2
    },
    },
    "required": [
        "tagId",
        "itemId"
    ]
},
"Tag": {
    "type": "object",
    "properties": {
        "name": {
            "type": "string",
            "example": "Hoodies"
        }
    },
    "required": [
        "name"
    ]
},
"ShoppingCartItem": {
    "type": "object",
    "properties": {
        "userId": {
            "type": "number",
            "example": 1
        },
        "itemId": {
            "type": "number",
            "example": 2
        },
        "quantity": {
            "type": "number",
            "example": 12
        }
    },
    "required": [
        "userId",
        "itemId"
    ]
}
},
"securitySchemes": {
    "bearerAuth": {
        "type": "http",
        "scheme": "bearer",
        "bearerFormat": "JWT"
    }
}
}
}

```

В результате чего была автоматически сгенерирована удобная, легко читаемая документация кода:

 Powered by [Redocly](#)

User API (1.0.0)

API for managing users

User

Operations related to users creation and usage

Get user by ID

GET /users/id/{id} >

Try it

Request >

Responses >

Response samples >

Create a new user

POST /users/create >

Try it

Request >

Request >

Responses >

Request samples >

User login

POST /users/login >

Try it

Request >

Responses >

Request samples >

Access private page (requires authentication)

GET /users/auth_only >

Try it

Responses >

Logout

POST /users/logout >

Try it

Responses >

Shop

Operations related to the shop

Test frontend homepage

GET /shop/ >

Try it

Responses >

Get all items

GET /shop/item >

Try it

Request >

Responses >

Response samples >

Get item by ID

GET /shop/item/{id} >

Try it

Request >

Responses >

Response samples >

Add item to shopping cart

POST /shop/item/{id} >

Try it

Request >

Responses >

Request samples ▾

Payload

application/json

Copy

```
{
  "quantity": 12,
  "itemId": 1
}
```

Response samples ▾

200

application/json

Response samples ▾

200

application/json

Copy

```
{  "userId": 1,  "itemId": 2,  "quantity": 12}
```

Get shopping cart contents

GET /shop/cart ▾

Try it

Copy

http://localhost:63342/shop/cart

Responses >

Вывод

В ходе выполнения данной работы был изучен и применен новый способ документации кода