

Нов български университет

Курсова работа по СИТВ548 Практика по  
програмиране и интернет технологии

Логистична компания

Изготвена от:

Ангел Тачев F109309

Димитър Димитров F109449

Калоян Анастасов

Мартин Ценев

Преподавател:

гл. ас. д-р Христина Костадинова

## Съдържание

API документация.....	3
1.Авторизация и сигурност .....	3
1.2 Автентикация.....	3
1.3. Управление на пратки .....	5
1.4. Управление на служители .....	6
1.5 Управление на офиси .....	7
1.6 Управление на клиенти .....	10
1.7 Управление на компании .....	11
1.8 Управление на такси за доставки.....	13
1.9 Управление на отчети.....	14
1.10 Управление на приходите .....	15
1.11 Управление на потребители.....	16
1.12 Грешки и статус кодове.....	18

## API документация

API (Application programming interface) е приложно-програмния интерфейс на изходния код, който операционната система или нейните библиотеки от ниско ниво предлагат за поддръжката на заявките от приложния софтуер или компютърните програми. Предоставя един по-абстрактен и опростен план за разработчика на приложения, който би му спестил изучаването на няколко различни слоя от операционната или софтуерната система зад интерфейса. По този начин се достига ефективност и бързина при адаптирането на нови софтуерни технологии.

В нашия случай API служи за управление на логистичната компания, включително за управление на автентикация, на клиенти, на компании, на пратки, на такси за доставка, на служители, на офиси, на отчети, на приходи и на потребители. Използва JWT базирана автентикация и ролеви контрол на достъпа (EMPLOYEE, CLIENT).

### 1. Авторизация и сигурност

Всички заявки(освен /api/auth/register и /api/auth/login) изискват JWT токен – отворен стандарт за сигурно предаване на информация между две страни, който използва JSON формат.

Токенът се подава в Authorization header:

Authorization: Bearer <jwt-token>

EMPLOYEE ще вижда всички пакети, приходи и служители.

CLIENT ще вижда само своите поръчки и пакети

## 1.2 Аутентикация

Базов URL: /api/auth

- Регистрация на потребител: POST /api/auth/register .

Тяло на заявката (JSON):

```
{  
  "username": "client123",  
  "password": "securepassword",  
  "role": "CLIENT"  
  "company_id": 1  
}
```

Примерен отговор(200 OK):

"User registered successfully!"

- Вход за потребител:

POST /api/auth/login – Генерира токена при успешно влизане.

```
{  
  "username": "client123",  
  "password": "securepassword"  
}
```

Отговор: "Bearer <jwt-token>"

- Излизане:

POST /api/auth/logout

Отговор: "Logout successful (token invalidated client-side)."

### 1.3. Управление на пратки

Базов URL: /api/packages

- Извличане на всички пратки - GET /api/packages:

Отговор:

```
[
  {
    "id": 1,
    "senderId": 1,
    "recipientId": 2,
    "courierId": 3,
    "deliveryType": "ADDRESS",
    "deliveryAddress": "123 Main St",
    "weight": 5.5,
    "price": 21.3,
    "status": "SENT",
    "company_id": 3
  }
]
```

- Създаване на нова пратка - POST /api/packages:

Заявка:

```
{
  "senderId": 1,
  "recipientId": 2,
  "courierId": 3,
  "deliveryType": "OFFICE",
  "deliveryAddress": "Courier Office Sofia",
  "weight": 3.2,
  "price": 15.0,
  "status": "SENT",
  "company_id": 3
}
```

Отговор:

```
{
  "id": 5,
  "senderId": 1,
  "recipientId": 2,
  "courierId": 3,
  "deliveryType": "OFFICE",
  "deliveryAddress": "Courier Office Sofia",
  "weight": 3.2,
  "price": 15.0,
}
```

- ```

    "status": "SENT",
    "company_id": 3
  }

```
- Обновяване на пратка - PUT /api/packages/{id}:
 

```

      {
        "weight": 4.0,
        "price": 18.0
      }
      
```
  - Изтриване на пратка:

DELETE /api/packages/{id} – само за EMPLOYEE. – ще връща празен отговор

#### 1.4. Управление на служители

Базов URL: /api/employees

- Списък с всички служители – GET /api/employees:

Отговор:

```

[
  {
    "id": 1,
    "office_id": 2,
    "user_id": 3,
    "company_id": 1
  },
  {
    "id": 2,
    "office_id": 2,
    "user_id": 4,
    "company_id": 1
  }
]

```

- Получаване на служител по ID - GET /api/employees/{id}:

Отговор:

```

{
  "id": 1,
  "office_id": 3,
  "user_id": 4,
  "company_id": 1
}

```

- Създаване на нов служител - POST /api/employees – при регистрация с role: "EMPLOYEE" служителят автоматично се разпределя в EMPLOYEE таблицата.

- Промяна на роля на служител: PUT /api/employees/{id}/role:  

```
{
  "role": "MANAGER"
}
```
- Промяна на потребителско име на служител: PUT /api/employees/{id}/username:  

```
{
  "username": "new.username"
}
```

Отговор:

```
{
  "username": "new.username"
}
```

- Получаване на всички служители за дадена компания: GET /api/employees/company/{companyId}:

Отговор:

```
[
  {
    "id": 1,
    "office_id": 2,
    "user_id": 4,
    "company_id": 1
  },
  {
    "id": 2,
    "office_id": 2,
    "user_id": 4,
    "company_id": 1
  }
]
```

- Изтриване на служител: DELETE /api/employees/{id} - ще връща празен отговор

## 1.5 Управление на офиси

Базов URL: /api/offices

- Получаване на списък със всички офиси - GET /api/offices:

Отговор:

```
[
  {
    "id": 1,
    "name": "Office Sofia",
    "address": "123 Office Street",
    "company_id": 3
  },
  {
    "id": 2,
    "name": "Office Plovdiv",
    "address": "456 Main Blvd",
    "company_id": 1
  }
]
```

- Получаване на информация за офис по ID - GET /api/offices/{id}:

Отговор:

```
{
  "id": 1,
  "name": "Office Sofia",
  "address": "123 Office Street",
  "company_id": 3
}
```

- Създаване на нов офис - POST /api/offices:

Заявка:

```
{
  "name": "Office Varna",
  "address": "789 Sea Blvd",
  "company_id": 2
}
```

Отговор:

```
{
  "id": 3,
  "name": "Office Varna",
  "address": "789 Sea Blvd",
  "company_id": 2
}
```

- Обновяване на информация за офис - PUT /api/offices/{id}:

Заявка:

```
{
```



```
"name": "Updated Office Sofia",
"address": "New Street 456",
}
```

Отговор:

```
{
  "id": 3,
  "name": "Updated Office Sofia",
  "address": "New Street 456",
  "company_id": 3
}
```

- Промяна на име на офис - PUT/api/offices/{id}/name:

```
{
  "name": "New Office Name"
}
```

Отговор:

```
{
  "id": 3,
  "name": "New Office Name",
  "address": "New Street 456",
  "company_id": 3
}
```

- Промяна на адрес на офис – PUT /api/offices/{id}/address:

```
{
  "address": "New Address 789"
}
```

Отговор:

```
{
  "id": 1,
  "name": "New Office Name",
  "address": "New Address 789",
  "company_id": 3
}
```

- Получаване на всички офиси за дадена компания - GET /api/offices/company/{companyId}:

Отговор:

```
[
  {
```

```

    "id": 3,
    "name": "Office Varna",
    "address": "789 Sea Blvd",
    "company_id": 2
  },
  {
    "id": 4,
    "name": "Office Burgas",
    "address": "100 Coast Road",
    "company_id": 2
  }
]

```

- Изтриване на офис – DELETE /api/offices/{id} - ще връща празен отговор.

## 1.6 Управление на клиенти

Базов URL: api/clients

- Получаване на списък с всички клиенти – GET /api/clients:

Отговор:

```

[
  {
    "id": 1,
    "company_id": 1,
    "user_id": 1
  },
  {
    "id": 2,
    "company_id": 1,
    "user_id": 2
  }
]

```

- Получаване на клиент по ID – GET /api/clients/{id}:

Отговор:

```

{
  "id": 2,
  "company_id": 1,
  "user_id": 2
}

```

- Създаване на нов клиент - POST /api/clients: при регистрация с role: "CLIENT" клиентът автоматично се разпределя в CLIENT таблицата.
- Обновяване на информация за клиент - PUT /api/clients/{id}:

Заявка:

```
{
  "user_id": 1
  "company_id": 2
}
```

Отговор:

```
{
  "id": 1,
  "user_id": 1,
  "company_id": 2
}
```

- Изтриване на клиент - DELETE /api/clients/{id} - ще връща празен отговор.

## 1.7 Управление на компании

Базов URL: /api/companies

- Създаване (обновяване на компания) - POST /api/companies:

Заявка:

```
{
  "id": 1,
  "name": "Courier Express",
  "address": "123 Delivery St., Cityville",
  "created": 2025-01-31 15:44:29.471197,
  "updated": 2025-02-01 15:44:29.471197,
  "email": contact1@courierexpress.com,
  "phone": "+123456789"
}
```

Отговор:

```
{
  "id": 1,
  "name": "Courier Express",
  "address": "123 Delivery St., Cityville",
  "created": 2025-02-01 15:44:29.471197,
  "updated": 2025-02-02 15:44:29.471197,
}
```

```
"email": contact1@courierexpress.com,  
"phone": "+123456789"  
}
```

- Получаване на списък за всички компании - GET /api/companies:

Отговор:

```
[  
  {  
    "id": 1,  
    "name": "Courier Express",  
    "address": "123 Delivery St., Cityville",  
    "created": 2025-01-31 15:44:29.471197,  
    "updated": 2025-01-02 15:44:29.471197,  
    "email": contact1@courierexpress.com,  
    "phone": "+123456789"  
  },  
  {  
    "id": 2,  
    "name": "Example Company",  
    "address": "1,123 Example Street",  
    "created": 2025-02-01 15:44:29.471197,  
    "updated": 2025-02-02 15:44:29.471197,  
    "email": contact@techlogistics.com,  
    "phone": "+1-555-789-1234"  
  }  
]
```

- Получаване на информация за компания по ID – GET /api/companies/{id}:

Отговор:

```
{  
  "id": 1,  
  "name": "Courier Express",  
  "address": "123 Delivery St., Cityville",  
  "created": 2025-01-31 15:44:29.471197,  
  "updated": 2025-02-01 15:44:29.471197,  
  "email": contact1@courierexpress.com,  
  "phone": "+123456789"  
}
```

- Изтриване на компания - DELETE /api/companies/{id} - ще връща празен отговор.

## 1.8 Управление на такси за доставки

Базов URL: /api/deliveryfees

- Създаване на такса – POST /api/deliveryfees:

Заявка:

```
{  
  "company_id": 1,  
  "price_per_kg": 5.0,  
  "price_per_kg_office": 2.50,  
  "price_per_kg_address": 4.00  
}
```

Отговор:

```
{  
  "id": 1  
  "company_id": 1,  
  "price_per_kg": 5.0,  
  "price_per_kg_office": 2.50,  
  "price_per_kg_address": 4.00  
}
```

- Извличане на всички такси за доставка – GET /api/deliveryfees:

Отговор:

```
[  
  {  
    "id": 1  
    "company_id": 1,  
    "price_per_kg": 5.0,  
    "price_per_kg_office": 2.50,  
    "price_per_kg_address": 4.00  
  },  
  {  
    "id": 2,  
    "company_id": 2,  
    "price_per_kg": 3.0,  
    "price_per_kg_office": 3.00,  
    "price_per_kg_address": 5.50  
  }  
]
```

- Получаване на такса по ID - GET /api/deliveryfees/{id}:

Отговор:

```
{
  "id": 1
  "company_id": 1,
  "price_per_kg": 5.0,
  "price_per_kg_office": 2.50,
  "price_per_kg_address": 4.00
}
```

- Обновяване на таксата за доставка – PUT /api/deliveryfees/{id}:

Заявка:

```
{
  "price_per_kg": 6.0,
  "price_per_kg_office": 3.00,
  "price_per_kg_address": 4.50
}
```

Отговор:

```
{
  "id": 1
  "company_id": 1,
  "weight_per_kg": 6.0,
  "price_per_kg_office": 3.00,
  "price_per_kg_address": 4.50
}
```

- Изтриване на такса – DELETE /api/deliveryfees/{id} - ще връща празен отговор.

## 1.9 Управление на отчети

Базов URL: /api/reports

- Получаване на отчети за представянето на служителите – GET /api/reports/employees:

Отговор:

```
[
  {
    "employeeId": 1,
    "employeeUsername": "john_doe",
    "totalPackages": 45
  },
  {
    "employeeId": 2,
```

```
    "employeeName": "jane_smith",  
    "totalPackages": 30  
  }  
]
```

- Извличане на всички пакети, които са изпратени, но не са получени – GET /api/reports/unreceived:

Отговор:

```
[  
  {  
    "id": 1,  
    "senderId": 5,  
    "receiverId": 12,  
    "status": "SENT",  
    "courierId": 3,  
    "price": 15.50  
  },  
  {  
    "id": 2,  
    "senderId": 8,  
    "receiverId": 20,  
    "status": "SENT",  
    "courierId": 2,  
    "price": 25.75  
  }  
]
```

- Изчисляване на приходите на компания за даден период – GET /api/reports/revenue:

Отговор:

```
{  
  "revenue": 10500.75  
}
```

## 1.10 Управление на приходите

Базов URL: /api/revenues

- Получаване на всички приходи – GET /api/revenues:

Отговор:

```
[  
  {  
    "id": 1,  
    "company_id": 3,  
    "package_id": 10,  
    "amount": 25.50,  
    "status": "PAID",  
    "date": "2017-01-01",  
    "employeeName": "jane_smith",  
    "totalPackages": 30  
  }  
]
```

```

    "recordDate": "2025-01-28",
    "createdAt": "2025-01-28",
    "updatedAt": "2025-01-28"
  },
  {
    "id": 2,
    "company_id": 1,
    "package_id": 5,
    "amount": 15.75,
    "recordDate": "2025-01-29",
    "createdAt": "2025-01-29",
    "updatedAt": "2025-01-29"
  }
]

```

- Получаване на приход по ID – GET /api/revenues/{id}:

Отговор:

```

{
  "id": 1,
  "company_id": 3,
  "package_id": 10,
  "amount": 25.50,
  "recordDate": "2025-01-28",
  "createdAt": "2025-01-28",
  "updatedAt": "2025-01-28"
}

```

- Изтриване на приход – DELETE /api/revenues/{id} - ще връща празен отговор.

## 1.11 Управление на потребители

Базов URL: /api/users

- Получаване на списък със всички потребители – GET api/users:

Отговор:

```

[
  {
    "id": 1,
    "username": "ivan.petroff",
    "role": "CLIENT",
    "company_id": 1,
    "createdAt": "2025-01-01",
    "updatedAt": "2025-01-15"
  },
  {
    "id": 2,

```



```
    "username": "maria.ilieva",
    "role": "EMPLOYEE",
    "company_id": 1,
    "createdAt": "2025-01-02",
    "updatedAt": "2025-01-16"
  }
]
```

- Получаване на потребител по ID – GET /api/users/{id}:

Отговор:

```
{
  "id": 1,
  "username": "ivan.petroff",
  "role": "CLIENT",
  "company_id": 1,
  "createdAt": "2025-01-01",
  "updatedAt": "2025-01-15"
}
```

- Създаване на нов потребител – POST /api/users: създаването на потребителя се случва при post заявката с auth/register и потребителят отива в users таблицата автоматично след регистрация.
- Обновяване на потребител: PUT /api/users/{id}:

Заявка:

```
{
  "username": "new.username",
  "password": "newSecurePassword123",
  "role": "EMPLOYEE",
}
```

Отговор:

```
{
  "id": 3,
  "username": "new.username",
  "role": "EMPLOYEE",
  "createdAt": "2025-01-03",
  "updatedAt": "2025-01-20",
  "companyId": 1
}
```

- Изтриване на потребител – DELETE /api/users/{id} - ще връща празен отговор.

### 1.12 Грешки и статус кодове

- 200 OK – успешно извършена операция
- 201 Created – успешно създаден ресурс
- 400 Bad Request – грешни или липсващи данни
- 401 Unauthorized – няма JWT токен или е невалиден
- 403 Forbidden – няма права за достъп
- 404 Not Found – ресурсът не е намерен

Нов български университет

Курсова работа по СИТВ548 Практика по  
програмиране и интернет технологии

Логистична компания

Изготвена от:

Ангел Тачев F109309

Димитър Димитров F109449

Калоян Анастасов

Мартин Ценев

Преподавател:

гл. ас. д-р Христина Костадинова

## Съдържание

|                                            |   |
|--------------------------------------------|---|
| Документация на базата данни .....         | 3 |
| Какви са връзките между таблиците: .....   | 5 |
| 2. Компания (company).....                 | 5 |
| 3. Служител (employees).....               | 6 |
| 4. Клиент (clients).....                   | 6 |
| 5. Офис (office) .....                     | 7 |
| 6. Пакети (packages).....                  | 7 |
| 7. Приходи (revenues).....                 | 8 |
| 8. Такса за доставка (delivery_fees) ..... | 8 |

# Документация на базата данни

Документацията се състои от това опише таблиците на базата от данни, която е свързана с приложението „Логистична компания“.

Базата от данни се състои от общо 8 таблици:

Clients- там се разпределят клиенти на компанията

Company- информация за компанията- име, адрес, кога е създадена и т.н.

delivery\_fees - цената за доставка до офис/адрес.

employees - таблицата за служителите съдържаща тип на служителя, id на компанията, office\_id, user\_id.

Office - id, address на офиса, име на офиса и id на компанията

Packages – id, company\_id- на компанията, courier\_id- id на

куриера, created\_at- създадена на , delivery\_addr- адрес,

delivery\_fee\_id- id на цената за доставка до офис/адрес,

delivery\_type- OFFICE/ADDRESS(спрямо enum-a), price,

recipient\_id - получател на доставка, sender\_id- изпращащ

пратката, status- SENT/DELIVERED (спрямо enum-a) дали е

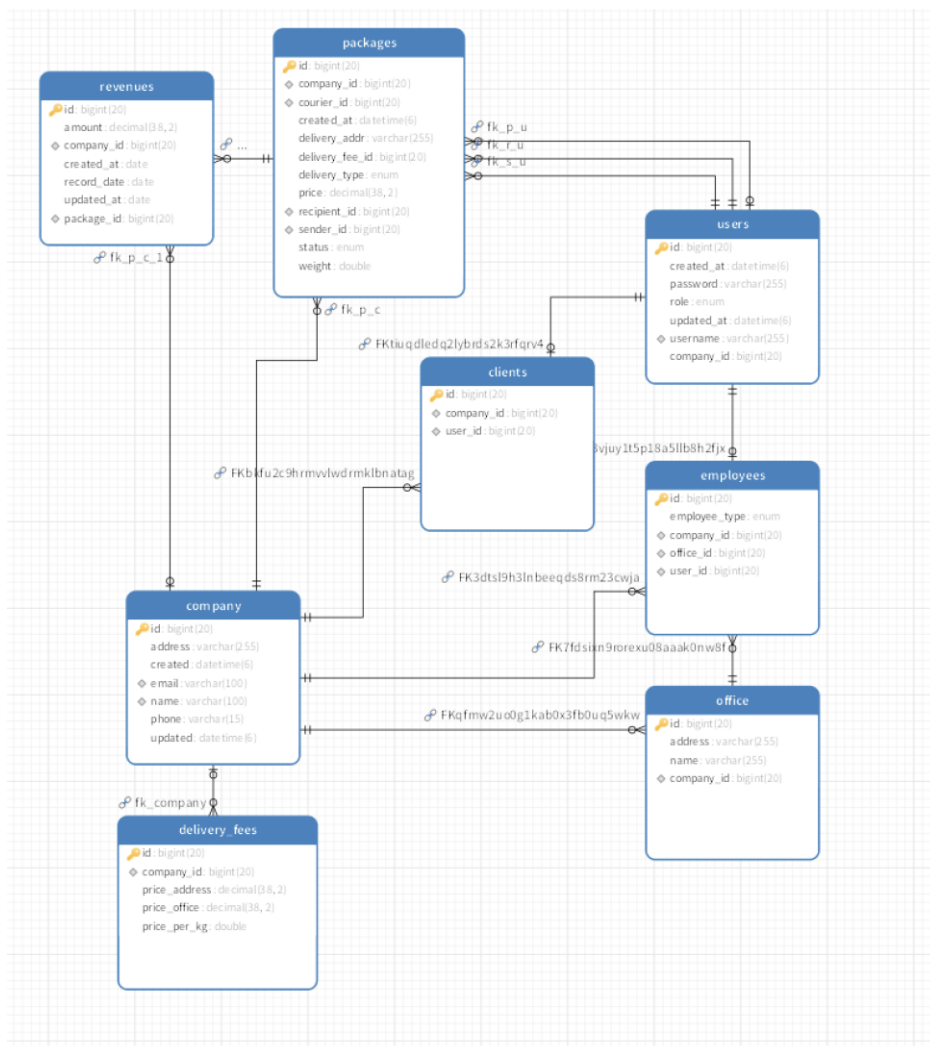
изпратена или е и доставена, weight- тежест на пратката

revenues - id на revenue, amount- стойността на прихода генериран от дадения package, company\_id, created\_at - създадено на, record\_date- дата, на която е отчетено, че се е случило revenue-то, updated\_at- дата, на която се е случило промяна с това revenue

package\_id - id на пакета, от който идва прихода

users - id, created\_at- кога е създаден user-а, password, role- CLIENT/EMPLOYEE, updated\_at – дата ако е променено нещо по user-а, username- име на user, company\_id- id на компания, с която е асоцииран клиента

## Диаграма на базата



Какви са връзките между таблиците:

### 1. Потребители (users)

Представява всички потребители в системата.

Връзки:

(Всеки потребител принадлежи към една компания.)

One-to-One Users → Client (Потребител може да бъде клиент.)

Потребители от типа „един към един“ → Служител (даден потребител може да бъде служител.)

### 2. Компания (company)

Представява куриерската фирма.

Връзки:

(Всеки потребител принадлежи на една компания.)

One-to-Many Company → Employees (Служителите работят за дадена компания.)

One-to-Many Company → Clients (Клиентите принадлежат на дадена компания.)

One-to-Many Company → Offices (Една компания има няколко офиса.)

One-to-Many Company → Packages (Пакети принадлежат на дадена компания.)

One-to-Many Company → Revenue (Приходите се проследяват за всяка компания.)

### 3. Служител (employees)

Представява служителите на дадена компания.

Връзки:

Служител → Потребители (Всеки служител има потребителски акаунт.)

Many-to-One Employee → Office (Един служител работи в един офис.)

Many-to-One Employee → Company (Служител принадлежи към една компания.)

### 4. Клиент (clients)

Представява клиентите, които използват куриерските услуги.

Връзки:

One-to-One Client → Users (Всеки клиент има потребителски акаунт.)

Many-to-One Client → Company (Един клиент принадлежи на една компания.)

One-to-Many Client → Packages (Клиентът може да изпраща няколко пакета.)



## 5. Офис (office)

Представява офиси, в които се оставят или обработват пакети.

Взаимоотношения:

(Всеки офис принадлежи на една компания.)

One-to-Many Office → Employees (Един офис има няколко служители.)

## 6. Пакети (packages)

Представява пратките, обработвани от компанията.

Връзки:

(Всеки пакет има изпращач.)

Много към един пакет → Получател (Клиент) (Всеки пакет има получател.)

Много към един пакет → Куриер (Служител) (Пакетът е разпределен на куриер.)

Many-to-One Package → DeliveryFee (Всеки пакет има такса за доставка.)

Many-to-One Package → Company (Пакетът се обработва от фирма.)

One-to-Many Package → Revenue (Всеки пакет генерира приходи.)

## 7. Приходи (revenues)

Представява приходите, генерирани от доставките на пакети.

Връзки: Връзките между пакетите са следните:

(Всеки запис за приходи е свързан с пакет.)

Many-to-One Revenue → Company (Приходът принадлежи на компания.)

## 8. Такса за доставка (delivery\_fees)

Представява цената на доставката въз основа на теглото и цената за доставката до офис или адрес.

Връзки:

(Всяка такса се определя от фирма.)

One-to-Many DeliveryFee → Package (Цените на пакетите се определят въз основа на таксите за доставка.)