BancoUP Backend API Documentation

✨ Overview

BancoUP is a backend system for managing users, accounts, and transactions securely and efficiently. Built with Node.js and MySQL via Sequelize ORM.

⚙️ Technologies

- **Node.js** (Express framework)
- **MySQL** (with Sequelize ORM)
- **JWT** for user authentication
- **bcrypt** for password encryption
- **RESTful** API structure

🔐 Authentication

`POST /user/login`

Authenticates a user.

Request Body:

```json
{
  "email": "user@example.com",
  "password": "SecurePass123!"
}
```

Responses:

- `200 OK`: Returns JWT token and user info.
- `401 Unauthorized`: Invalid credentials.
- `403 Forbidden`: User is blocked.

JWT Middleware: `auth.middleware.js`

Validates Bearer token. Injects user data (`req.usuario`).

👤 User Endpoints

`POST /user/insert`

Creates a new user.

Request Body:

```
{
  "email": "user@example.com",
  "password": "SecurePass123!",
  "type": "normal"
}
```

`GET /user/profile` *(Auth required)*

Returns authenticated user profile and account info.

`GET /user/users?blocked=0|1`

Lists users filtered by blocked status.

`POST /user/unlock`

Unlocks a specific user.

Request Body:

```
{
  "id": 1
}
```

`POST /user/block`

Blocks a specific user.

Request Body:

```
{
  "id": 1
}
```

💳 Account Endpoints

`POST /account/create`

Creates a bank account for authenticated user.

Request Body:

```
{
  "userID": 1
}
```

`POST /account/activate`

Activates an account by `accountID`.

Request Body:

```json
{
  "accountID": 10001
}
```

`POST /account/deactivate`

Deactivates an account by `accountID`.

Request Body:

```json
{
  "accountID": 10001
}
```

💸 Transaction Endpoints

`POST /transaction/`

Transfers funds between accounts.

Request Body:

```json
{
  "senderId": 10001,
  "recipientAccountNumber": 10002,
  "amount": 500,
  "concept": "Payment"
}
```

Validations:

- Amount: $500 – $10,000
- Cannot transfer to self
- Daily limit: $10,000
- Max balance: $50,000

`GET /transaction/account/:accountId`

Retrieves all transactions involving the given account.

🗃️ Database Schema

Tables

- **user**: id, email, password, type, failedAttempts, blocked
- **account**: id, userID, balance, active
- **transaction**: id, senderID, receiverID, amount, status, concept, date
- **totalsentperday** / **totalreceivedperday**: Daily tracking per account

Scheduled Events

- `totalsentperday_event`
- `totalreceivedperday_event`

🛠️ Project Setup

Environment Variables

- `PORT`: App port (default 3000)
- `JWT_SECRET`: Secret key for signing JWT

Entry Point

```
npm install
npm run dev
```

🔐 Security

- JWT protects sensitive endpoints
- Passwords hashed with bcrypt
- Account operations are protected with auth middleware