

Tringle Candidate Backend Project - May 2022

Dear Candidate,

Thank you for your interest joining in Tringle. We would like to know more about you and your technical skills with a simple challenge; a payment rest api project.

In this project, you are free to use any of your favorite language/framework or runtime environment. But, it is required that you should name your routes and parameters exactly as described in the following section.

You should also provide brief documentation about how to run/configure your project. To simplify it, you can use docker but it is not a requirement, it is recommended. Final decision is up to you.

Project Definition

In this challenge, you are asked to create a payment rest api that allows users to create an account (both individual or corporate), deposit/withdraw money and make a payment.

Only individual accounts can deposit or withdraw money. Payments can only be wired from an individual account to a corporate account.

Each money transaction should create an accounting with transaction amount, transaction type and account information. These transactions should be able to be listed as accounting history.

All accounts and accounting transactions should be held in memory, so no database implementation is required.

Each route should have its own input and business validation. Validations should be determined by you.

Here are routes definitions:

1. Account Create Route

POST <http://localhost:5050/account>

```
{
  "accountNumber": "number",
  "currencyCode": { enum: ["TRY", "USD", "EUR"] },
  "ownerName": "string",
  "accountType": { enum: ["individual", "corporate"] }
}
```

2. Account Info Route

GET <http://localhost:5050/account/{accountNumber}>

response:

```
{
  "accountNumber": "number",
  "currencyCode": "string",
  "ownerName": "string",
  "accountType": { enum: ["individual", "corporate"] },
  "balance": "number"
}
```

3. Payment Route

POST <http://localhost:5050/payment>

```
{
  "senderAccount": "number",
  "receiverAccount": "number",
  "amount": "number",
}
```

4. Deposit Route

POST http://localhost:5050/deposit

```
{  
  "accountNumber": "number",  
  "amount": "number"  
}
```

5. Withdraw Route

POST http://localhost:5050/withdraw

```
{  
  "accountNumber": "number",  
  "amount": "number"  
}
```

6. Transaction History

GET http://localhost:5050/accounting/{accountNumber}

response:

```
{  
  "accountNumber": "number",  
  "amount": "number",  
  "transactionType": { enum: ["payment", "deposit", "withdraw"] },  
  "createdAt": "date"  
}
```

Acceptance Criteria:

- API should communicate through REST protocols and use JSON for data exchange.
- No database is required. Only in memory.
- All routes explained in the previous section should be implemented.

- Successful transactions should return http 2xx code.
- All necessary validations should be implemented.
- Validation errors should return http 4xx code.
- Any programming language, framework or runtime environment is allowed.
- Your submission should include instructions to set up or run the application.
- Writing unit tests is a bonus.
- Writing clean code is a bonus.
- Dockerizing the application is a bonus.
- Account numbers should be integer and unique.
- Account balance should be decimal and has 2 decimal places.

Submit: Please send your source code and documentation or github url to candidate@tringle.io.

Deadline: 22th of May 2022

Support: If you have any questions, please contact us on candidate@tringle.io.