BanKING

Created and documented by: Hugo Korte



Inhoud

[Backend: System Description 3](#_Toc106706889)

[Layers: 3](#_Toc106706890)

[Controllers: 3](#_Toc106706891)

[Services: 3](#_Toc106706892)

[Repositories: 3](#_Toc106706893)

[Backend: Class Diagrams 4](#_Toc106706894)

[Class Diagram: BanKING 4](#_Toc106706895)

[AppUser: 4](#_Toc106706896)

[BankAccount: 4](#_Toc106706897)

[Transaction: 4](#_Toc106706898)

[Backend: Rest Specification 5](#_Toc106706899)

[Backend: Sequence Diagrams 6](#_Toc106706900)

[Sequence: Sign Up 6](#_Toc106706901)

[Sequence: Deposit 7](#_Toc106706902)

[Backend: Implementation Choices 8](#_Toc106706903)

[Frontend: Functionality Description 9](#_Toc106706904)

[Frontend: Wireframes 10](#_Toc106706905)

[Frontend: Visual Representations 11](#_Toc106706906)

# Backend: System Description

## Layers:

1. Controllers
2. Services
3. Repositories

### Controllers:

De controllers zijn het start en eindpunt van elke request, deze controllers hebben als functie om alle requests op te vangen en als response een van de functies in de service klassen aan te roepen die dan de request kan verwerken en bijbehorende response kan genereren.

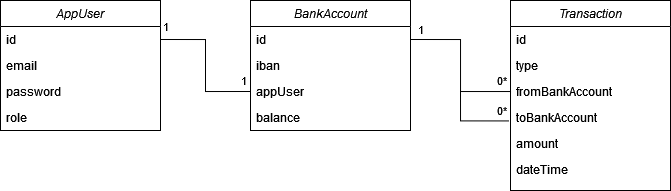
Services:  
De services zijn de klassen die alle ‘Business logic’ hanteren. De services hebben meestal een aantal methodes die relevant zijn aan de service en nemen een request als input (met soms parameters) en die dit request verwerken tot een netjes pakketje die ze dan weer terugsturen naar de controller.

### Repositories:

De repositories dienen als tussen-klassen waardoor de services met de database kan communiceren. Dit betekent dus dat de repositories een aantal methoden hebben (denk aan: Find By, Save en Delete) en dat die methoden worden omgezet in SQL statements waardoor de CRUD functionaliteit mogelijk is op de database.

# Backend: Class Diagrams

## Class Diagram: BanKING



Het klasse diagram bestaat uit 3 klassen:

1. *AppUser:*
2. *BankAccount*
3. *Transaction*

### AppUser:

De *AppUser-*klasserepresenteert een gebruiker van de applicatie.

### BankAccount:

De *BankAccount-*klasse representeert een bank rekening en heeft een verplichte One-To-One relatie met de *AppUser* (dit kun je zien als de eigenaar van de bank rekening). Wanneer een *AppUser* wordt aangemaakt wordt dan ook al direct een *BankAccount* aangemaakt en gekoppeld.

### Transaction:

De *Transaction*-klasse representeert een transactie tussen 2 *BankAccount-*klassen en heeft dus een One-To-Many relatie met een *BankAccount-*klasse, omdat een transactie (vaak) tussen 2 bank rekeningen is kan 1 transactie 2 relaties hebben met 2 verschillende *BankAccount-*klassen.

# Backend: Rest Specification

Omdat het domein en protocol verschillend kan zijn zal ik alleen de resource paths noteren en niet de volledige uri’s

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **POST** | **/api/app-user** | | | |
| Dit request zal een app-user aanmaken en registreren in de database | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *Add a \* to the name of required parameters.* | | app-user\* | body | Email en wachtwoord van app-user |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 201 | app-user is aangemaakt | |
| 409 | app-user met die email bestaat al | |
| 422 | De request mist/bevat verkeerde informatie | |
|  |  | |
|  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GET** | **/api/app-user/{email}** | | | |
| [description here. Explain what the request does when executed successfully.] | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *Add a \* to the name of required parameters.* | | app-user\* | path | Email van app-user die opgehaald moet worden |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 200 | app-user was opgehaald | |
| 401 | request heeft niet benodigde rechten om app-user op te halen | |
| 404 | app-user was niet gevonden | |
|  |  | |
|  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PUT** | **/api/app-user/{email}** | | | |
| Update app-user password by specifying email from app-user | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *Add a \* to the name of required parameters.* | | email\* | path | app-user dat veranderd moet worden |
| Wachtwoord\* | body | wachtwoord dat veranderd moet worden |
|  |  |  |
|  |  |  |
|  |  |  |
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 204 | app-user wachtwoord is veranderd | |
| 401 | request heeft niet benodigde rechten om app-user te veranderen | |
| 404 | app-user met die email kon niet worden gevonden | |
| 422 | wachtwoord voldoet niet tot criteria | |
|  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DELETE** | **/api/app-user/{email}** | | | |
| Delete app-user by specifying email from app-user | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *Add a \* to the name of required parameters.* | | email\* | path | email van app-user die verwijderd moet worden |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 204 | app-user was verwijderd | |
| 401 | geen toegang tot deze resource | |
| 404 | email was niet gevonden | |
|  |  | |
|  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **POST** | **/api/app-user/authenticate** | | | |
| Authenticate user and get JSON Web Tokens (JWT) to access any other resources without re-authenticing | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *Add a \* to the name of required parameters.* | | email\* | body | email van app-user |
| password\* | body | wachtwoord van app-user |
|  |  |  |
|  |  |  |
|  |  |  |
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 200 | response json bevat access en refresh token | |
| 401 | incorrect wachtwoord | |
| 404 | email was niet gevonden | |
|  |  | |
|  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **POST** | **/api/bank-account/{iban}/transactions/{TransactionType}** | | | |
| Create a transaction that allows the user to alter the balance of their bankaccount | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *Add a \* to the name of required parameters.* | | iban\* | path | email van app-user |
| transactietype\* | path | type transactie (storten, ophalen, etc) |
| amount\* | body | Aantal euro’s |
| bankAccTo | body | iban waar verzend requests heen gaan |
|  |  |  |
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 201 | transactie was successvol | |
| 401 | geen toegang tot deze resource | |
| 404 | iban was niet gevonden | |
| 422 | aantal of iban is incorrect | |
|  |  | |

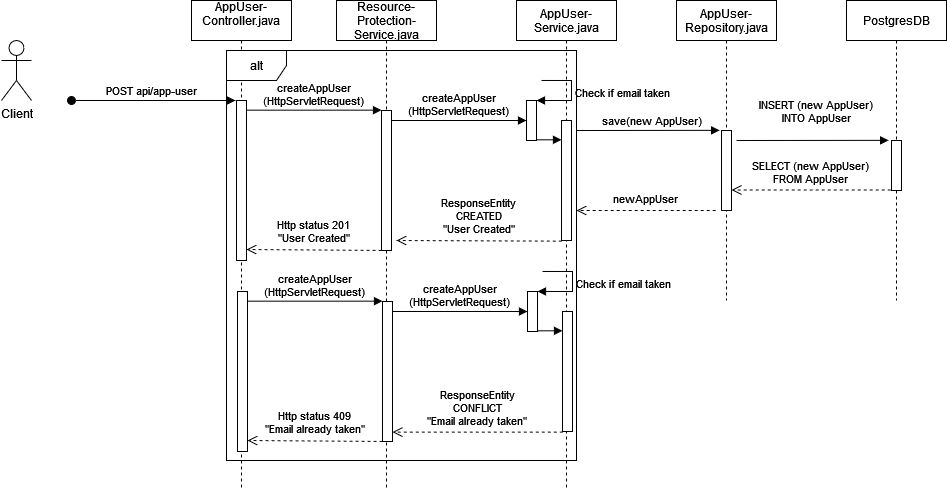
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GET** | **/api/bank-account /{iban}/transactions?limit={limit}&sortBy={sortBy]** | | | |
| [description here. Explain what the request does when executed successfully.] | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *Add a \* to the name of required parameters.* | | iban\* | path | transactions of iban |
| limit | query | aantal transacties |
| sortBy | query | sorteer algoritme |
|  |  |  |
|  |  |  |
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 200 | json met alle transactie’s | |
| 401 | geen toegang tot deze resource | |
| 404 | iban niet gevonden | |
|  |  | |
|  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PUT** | **/api/bank-account/{iban}/transaction/{id}** | | | |
| Update app-user password by specifying email from app-user | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *Add a \* to the name of required parameters.* | | iban\* | path | app-user dat veranderd moet worden |
| id\* | path | wachtwoord dat veranderd moet worden |
| fromBankAcc\* | body | bank rekening van zender |
| toBankAcc\* | body | bank rekening van ontvanger |
| amount\* | body | aantal euro’s |
|  | | timestamp\* | body | tijd en datum |
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 204 | transactie is veranderd | |
| 401 | geen toegang tot deze resource | |
| 404 | transactie nog niet gevonden | |
| 422 | zender/ontvanger/aantal/tijdendatum was niet goed | |
|  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DELETE** | **/api/app-user/{iban}/transactions** | | | |
| [description here. Explain what the request does when executed successfully.] | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *Add a \* to the name of required parameters.* | |  | path |  |
|  | path |  |
|  | path |  |
|  | path |  |
|  | path |  |
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | |  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |

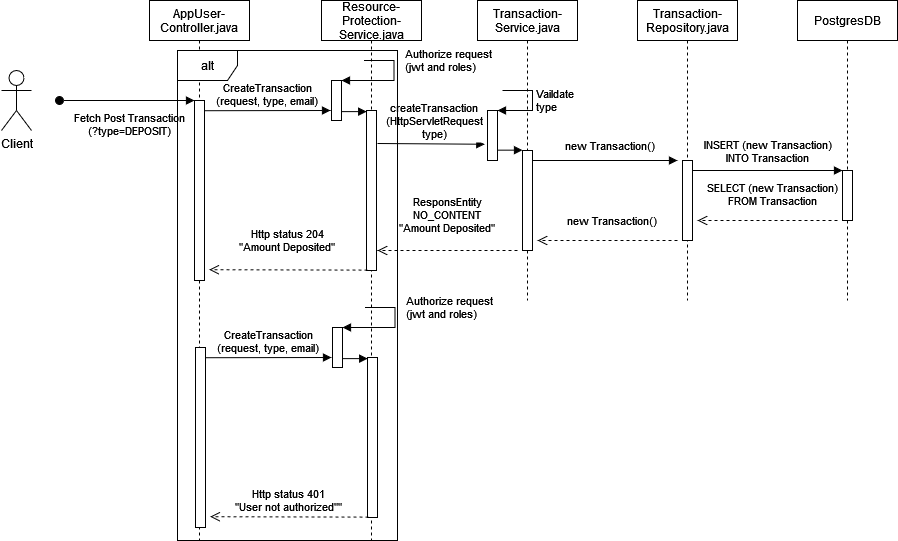
# Backend: Sequence Diagrams

## Sequence: Sign Up



In dit sequence diagram wordt gevisualiseerd hoe een gebruiker een account aan kan maken (sign-up) met mijn applicatie. Om het proces goed te visualiseren is bij elke laag neergezet hoe ze de request verwerken en wat ze als response teruggeven. Voor elk request dat binnenkomt bij AppUser-Service.java word een validatie check gedaan zodat alleen geldige email adressen zich kunnen aanmelden.

## Sequence: Deposit



In dit sequence diagram wordt gevisualiseerd hoe een gebruiker geld kan storten op zijn/haar bank rekening via Deposit. Wanneer een request aankomt in de ‘ProtectedResourceService’, word elke request, indien dat nodig is, geauthoriseert door middel van de rol uit de access token te halen en kijken of die rol de benodigde ‘LevelOfClearance’ bevat om toegang tot die resource te krijgen.

# Backend: Implementation Choices

# Frontend: Functionality Description

# Frontend: Wireframes

# Frontend: Visual Representations