

Auth

POST /auth/login

Opis

Logowanie użytkownika. Użytkownik podaje swój email oraz hasło. Zwraca token dostępu oraz informacje o użytkowniku.

Przykładowe zapytanie (Request Body)

```
{
  "email": "jankowalski@example.com",
  "password": "securePassword123"
}
```

Walidacja

- email(String, wymagane): Musi być poprawnym adresem e-mail
- password(String, wymagane): Nie może być puste

Przykładowa odpowiedź (Response Body)

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "message": "Login completed successfully!",
  "user": {
    "id": 1,
    "firstName": "Jan",
    "lastName": "Kowalski",
    "email": "jankowalski@example.com",
    "phoneNumber": "+48123456789",
    "dateOfJoining": "2022-01-01",
    "currentBudget": 1000.0,
    "currency": {
      "currencyId": 1,
      "isoCode": "PLN"
    },
    "darkMode": true,
    "notificationsByEmail": true,
    "notificationsByPhone": false
  }
}
```

POST /auth/register

Opis

Rejestracja nowego użytkownika. Użytkownik podaje swoje dane osobowe, email oraz hasło. Zwraca token dostępu oraz informacje o użytkowniku.

Przykładowe zapytanie (Request Body)

```
{
  "firstName": "Jan",
  "lastName": "Kowalski",
  "email": "jankowalski@example.com",
  "password": "securePassword123"
}
```

Walidacja

- firstName(String, wymagane): Nie może być puste
- lastName(String, wymagane): Nie może być puste
- email(String, wymagane): Musi być poprawnym adresem e-mail
- password(String, wymagane): Nie może być puste, minimum 6 znaków

Przykładowa odpowiedź (Response Body)

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "message": "Registration completed successfully!",
  "user": {
    "id": 1,
    "firstName": "Jan",
    "lastName": "Kowalski",
    "email": "jankowalski@example.com",
    "phoneNumber": "+48123456789",
    "dateOfJoining": "2022-01-01",
    "currentBudget": 1000.0,
    "currency": {
      "currencyId": 1,
      "isoCode": "PLN"
    },
    "darkMode": true,
    "notificationsByEmail": true,
    "notificationsByPhone": false
  }
}
```

Currency

GET /currency/all

Opis

Pobiera listę wszystkich dostępnych walut.

Przykładowe zapytanie

```
GET /currency/all
```

Przykładowa odpowiedź (Response Body)

```
[
  {
    "currencyId": 1,
    "isoCode": "USD"
  },
  {
    "currencyId": 2,
    "isoCode": "EUR"
  }
]
```

General Transaction

POST /general-transactions/plot-data

Opis

Zwraca dane do wykresu budżetu użytkownika w zadanym zakresie dat.

Przykładowe zapytanie (Request Body)

```
{
  "userId": 5,
  "startDate": "2025-01-01",
  "endDate": "2025-01-31"
}
```

Walidacja

- `userId(Integer, wymagane)`: Nie może być null, musi być dodatnie.
- `startDate(Date, wymagane)`: Data po 2000 roku, nie może być null.
- `endDate(Date, wymagane)`: Data po 2000 roku, nie może być null.

Przykładowa odpowiedź (Response Body)

```
[
  {
    "date": "2025-01-01",
    "amount": 250.00
  },
  {
    "date": "2025-01-15",
    "amount": 480.00
  }
]
```

GET /general-transactions/next-transaction/{userId}

Opis

Zwraca informacje o kolejnej transakcji użytkownika (przychód lub wydatek).

Przykładowe zapytanie

```
GET /general-transactions/next-transaction/5?isIncome=true
```

Przykładowa odpowiedź (Response Body)

```
{
  "transactionName": "Salary",
  "date": "2025-06-15",
  "amount": 3500.00,
  "isIncome": true,
  "currencyIsoCode": "PLN"
}
```

GET /general-transactions/estimated-balance/{userId}

Opis

Zwraca przewidywane saldo użytkownika na koniec miesiąca.

Przykładowe zapytanie

```
GET /general-transactions/estimated-balance/5
```

Przykładowa odpowiedź (Response Body)

```
{
  "estimatedBalance": 1234.56
}
```

POST /general-transactions/revenues-expenses-in-range

Opis

Zwraca sumę przychodów i wydatków użytkownika w zadanym zakresie dat.

Przykładowe zapytanie (Request Body)

```
{
  "userId": 5,
  "startDate": "2025-01-01",
  "endDate": "2025-01-31"
}
```

Walidacja

- `userId(Integer, wymagane)`: Nie może być null, musi być dodatnie.
- `startDate(LocalDate, wymagane)`: Data po 2000 roku, nie może być null.
- `endDate(LocalDate, wymagane)`: Data po 2000 roku, nie może być null.

Przykładowa odpowiedź (Response Body)

```
{
  "revenues": 1500.00,
  "expenses": 500.00
}
```

GET /general-transactions/monthly-summary/{userId}

Opis

Zwraca podsumowanie przychodów, wydatków i bilansu użytkownika w bieżącym miesiącu.

Przykładowe zapytanie

```
GET /general-transactions/monthly-summary/5
```

Przykładowa odpowiedź (Response Body)

```
{
  "monthlyIncome": 3000.00,
  "monthlyExpenses": 1200.00,
  "monthlyBalance": 1800.00
}
```

GET /general-transactions/current-balance/{userId}

Opis

Zwraca aktualne saldo użytkownika.

Przykładowe zapytanie

```
GET /general-transactions/current-balance/5
```

Przykładowa odpowiedź (Response Body)

```
{
  "currentBalance": 987.65
}
```

One Time Transaction

POST /onetime-transaction

Opis

Tworzy nową jednorazową transakcję.

Przykładowe zapytanie (Request Body)

```
{
  "userId": 5,
  "name": "Zakupy spożywcze",
  "amount": 150.00,
  "currencyId": 1,
  "isIncome": false,
  "date": "2025-05-01",
  "description": "Zakupy w supermarkecie"
}
```

Walidacja

- `userId(Integer, wymagane)`: Nie może być null, musi być dodatnie.
- `name(String, wymagane)`: Nie może być puste, maksymalnie 512 znaków.
- `amount(BigDecimal, wymagane)`: Nie może być null, musi być dodatnie.
- `currencyId(Integer, wymagane)`: Nie może być null.
- `isIncome(Boolean, wymagane)`: Nie może być null.
- `date(Date, wymagane)`: Nie może być null, data po 2000 roku.
- `description(String, opcjonalne)`: Maksymalnie 512 znaków.

Przykładowa odpowiedź (Response Body)

```
{
  "transactionId": 101,
  "userId": 5,
  "name": "Zakupy spożywcze",
  "amount": 150.00,
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "isIncome": false,
  "date": "2025-05-01",
  "description": "Zakupy w supermarkecie"
}
```

GET /onetime-transaction/{id}

Opis

Zwraca szczegóły jednorazowej transakcji.

Przykładowe zapytanie

```
GET /onetime-transaction/101
```

Przykładowa odpowiedź (Response Body)

```
{
  "transactionId": 101,
  "userId": 5,
  "name": "Zakupy spożywcze",
  "amount": 150.00,
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "isIncome": false,
```

```
"date": "2025-05-01",  
"description": "Zakupy w supermarkecie"  
}
```

PUT /onetime-transaction/{id}

Opis

Aktualizuje dane jednorazowej transakcji.

Przykładowe zapytanie (Request Body)

```
{  
  "userId": 5,  
  "name": "Zakupy spożywcze",  
  "amount": 155.00,  
  "currencyId": 1,  
  "isIncome": false,  
  "date": "2025-05-01",  
  "description": "Zakupy w supermarkecie"  
}
```

Walidacja

- `userId(Integer, wymagane)`: Nie może być null, musi być dodatnie.
- `name(String, wymagane)`: Nie może być puste, maksymalnie 512 znaków.
- `amount(BigDecimal, wymagane)`: Nie może być null, musi być dodatnie.
- `currencyId(Integer, wymagane)`: Nie może być null.
- `isIncome(Boolean, wymagane)`: Nie może być null.
- `date(Date, wymagane)`: Nie może być null, data po 2000 roku.
- `description(String, opcjonalne)`: Maksymalnie 512 znaków.

Przykładowa odpowiedź (Response Body)

```
{  
  "transactionId": 101,  
  "userId": 5,  
  "name": "Zakupy spożywcze",  
  "amount": 155.00,  
  "currency": {  
    "currencyId": 1,  
    "isoCode": "PLN"  
  },  
  "isIncome": false,  
  "date": "2025-05-01",  
  "description": "Zakupy w supermarkecie"  
}
```


DELETE /onetime-transaction/{id}

Opis

Usuwa jednorazową transakcję.

Przykładowe zapytanie

```
DELETE /onetime-transaction/101
```

Przykładowa odpowiedź

```
Status: 204 No Content
```

GET /onetime-transaction/all/{userId}

Opis

Zwraca wszystkie jednorazowe transakcje użytkownika.

Przykładowe zapytanie

```
GET /onetime-transaction/all/5
```

Przykładowa odpowiedź (Response Body)

```
[
  {
    "transactionId": 101,
    "userId": 5,
    "name": "Zakupy spożywcze",
    "amount": 150.00,
    "currency": {
      "currencyId": 1,
      "isoCode": "PLN"
    },
    "isIncome": false,
    "date": "2025-05-01",
    "description": "Zakupy w supermarkecie"
  },
  {
    "transactionId": 102,
    "userId": 5,
    "name": "Bilet do kina",
```

```
[
  {
    "amount": 50.00,
    "currency": {
      "currencyId": 1,
      "isoCode": "PLN"
    },
    "isIncome": false,
    "date": "2025-05-02",
    "description": "Kino z przyjaciółmi"
  }
]
```

Period

GET /period/all

Opis

Zwraca listę wszystkich dostępnych okresów.

Przykładowe zapytanie

```
GET /period/all
```

Przykładowa odpowiedź (Response Body)

```
[
  {
    "name": "Dzienny",
    "code": "P1D"
  },
  {
    "name": "Tygodniowy",
    "code": "P1W"
  },
  {
    "name": "Miesięczny",
    "code": "P1M"
  },
  {
    "name": "Roczny",
    "code": "P1Y"
  }
]
```

Plan

POST /plan

Opis

Tworzy nowe marzenie.

Przykładowe zapytanie (Request Body)

```
{
  "userId": 5,
  "name": "Oszczędności wakacyjne",
  "amount": 5000.00,
  "currencyId": 1,
  "description": "Plan na wyjazd wakacyjny"
}
```

Walidacja

- `userId(Integer, wymagane)`: Nie może być null, musi być dodatnie.
- `name(String, wymagane)`: Nie może być puste, maksymalnie 512 znaków.
- `amount(BigDecimal, wymagane)`: Nie może być null, musi być dodatnie.
- `currencyId(Integer, wymagane)`: Nie może być null.
- `description(String, opcjonalne)`: Maksymalnie 512 znaków.

Przykładowa odpowiedź (Response Body)

```
{
  "planId": 201,
  "userId": 5,
  "name": "Oszczędności wakacyjne",
  "description": "Plan na wyjazd wakacyjny",
  "amount": 5000.00,
  "date": "2025-05-05",
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "completed": false,
  "actualAmount": 0.00,
  "canBeCompleted": false,
  "subplansCompleted": 0.0,
  "estimatedCompletionDate": "2025-10-10"
}
```

GET /plan/{id}

Opis

Zwraca szczegóły marzenia.

Przykładowe zapytanie

```
GET /plan/201
```

Przykładowa odpowiedź (Response Body)

```
{
  "planId": 201,
  "userId": 5,
  "name": "Oszczędności wakacyjne",
  "description": "Plan na wyjazd wakacyjny",
  "amount": 5000.00,
  "date": "2025-05-05",
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "completed": false,
  "actualAmount": 0.00,
  "canBeCompleted": false,
  "subplansCompleted": 0.0,
  "estimatedCompletionDate": "2025-10-10"
}
```

PUT /plan/{id}

Opis

Aktualizuje dane dla marzenia.

Przykładowe zapytanie (Request Body)

```
{
  "userId": 5,
  "name": "Oszczędności wakacyjne",
  "amount": 5200.00,
  "currencyId": 1,
  "description": "Plan na wyjazd wakacyjny"
}
```

Walidacja

- `userId(Integer, wymagane)`: Nie może być null, musi być dodatnie.
- `name(String, wymagane)`: Nie może być puste, maksymalnie 512 znaków.

- amount(BigDecimal, wymagane): Nie może być null, musi być dodatnie.
- currencyId(Integer, wymagane): Nie może być null.
- description(String, opcjonalne): Maksymalnie 512 znaków.

Przykładowa odpowiedź (Response Body)

```
{
  "planId": 201,
  "userId": 5,
  "name": "Oszczędności wakacyjne",
  "description": "Plan na wyjazd wakacyjny",
  "amount": 5200.00,
  "date": "2025-05-05",
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "completed": false,
  "actualAmount": 0.00,
  "canBeCompleted": false,
  "subplansCompleted": 0.0,
  "estimatedCompletionDate": "2025-10-10"
}
```

POST /plan/complete/{id}

Opis

Oznacza marzenie jako zakończone.

Przykładowe zapytanie

```
POST /plan/complete/201
```

Przykładowa odpowiedź (Response Body)

```
{
  "planId": 201,
  "userId": 5,
  "name": "Oszczędności wakacyjne",
  "description": "Plan na wyjazd wakacyjny",
  "amount": 5000.00,
  "date": "2025-05-05",
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
}
```

```
"completed": true,  
"actualAmount": 5000.00,  
"canBeCompleted": true,  
"subplansCompleted": 1.0,  
"estimatedCompletionDate": "2025-05-05"  
}
```

Ustawianie innej daty

Aby ustawić inną datę realizacji marzenia niż bieżący dzień, można opcjonalnie przekazać parametr `completionDate` w zapytaniu:

```
POST /plan/complete/201?completionDate=2025-04-04
```

POST /plan/uncomplete/{id}

Opis

Cofa oznaczenie marzenia jako zakończone.

Przykładowe zapytanie

```
POST /plan/uncomplete/201
```

Przykładowa odpowiedź (Response Body)

```
{  
  "planId": 201,  
  "userId": 5,  
  "name": "Oszczędności wakacyjne",  
  "description": "Plan na wyjazd wakacyjny",  
  "amount": 5000.00,  
  "date": null,  
  "currency": {  
    "currencyId": 1,  
    "isoCode": "PLN"  
  },  
  "completed": true,  
  "actualAmount": 5000.00,  
  "canBeCompleted": true,  
  "subplansCompleted": 1.0,  
  "estimatedCompletionDate": "2025-05-25"  
}
```

GET /plan/all/{userId}

Opis

Zwraca wszystkie marzenia użytkownika.

Przykładowe zapytanie

```
GET /plan/all/5
```

Przykładowa odpowiedź (Response Body)

```
[
  {
    "planId": 201,
    "userId": 5,
    "name": "Oszczędności wakacyjne",
    "description": "Plan na wyjazd wakacyjny",
    "amount": 5000.00,
    "date": "2025-05-05",
    "currency": {
      "currencyId": 1,
      "isoCode": "PLN"
    },
    "completed": false,
    "actualAmount": 0.00,
    "canBeCompleted": false,
    "subplansCompleted": 0.0,
    "estimatedCompletionDate": "2025-05-05"
  }
]
```

DELETE /plan/{id}

Opis

Usuwa marzenie.

Przykładowe zapytanie

```
DELETE /plan/201
```

Przykładowa odpowiedź

```
Status: 204 No Content
```

Recurring Transaction

POST /recurring-transaction

Opis

Tworzy nową cykliczną transakcję.

Przykładowe zapytanie (Request Body)

```
{
  "userId": 5,
  "name": "Abonament Spotify",
  "amount": 29.99,
  "currencyId": 1,
  "isIncome": false,
  "interval": "P1M",
  "firstPaymentDate": "2025-05-01",
  "finalPaymentDate": "2025-12-01",
  "description": "Miesięczna subskrypcja"
}
```

Walidacja

- `userId(Integer, wymagane)`: Nie może być null, musi być dodatnie.
- `name(String, wymagane)`: Nie może być puste, maksymalnie 512 znaków.
- `amount(BigDecimal, wymagane)`: Nie może być null, musi być dodatnie.
- `currencyId(Integer, wymagane)`: Nie może być null.
- `isIncome(Boolean, wymagane)`: Nie może być null.
- `interval (String, wymagane)`: Nie może być puste, wymagany poprawny format okresu.
- `firstPaymentDate(Date, wymagane)`: Nie może być null, data po 2000 roku.
- `finalPaymentDate(Date, wymagane)`: Nie może być null, data po 2000 roku.
- `description(String, opcjonalne)`: Maksymalnie 512 znaków.

Przykładowa odpowiedź (Response Body)

```
{
  "transactionId": 301,
  "userId": 5,
  "name": "Abonament Spotify",
  "amount": 29.99,
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "isIncome": false,
  "firstPaymentDate": "2025-05-01",
}
```



```
"nextPaymentDate": "2025-06-01",
"finalPaymentDate": "2025-12-01",
"interval": "MONTHLY",
"description": "Miesięczna subskrypcja"
}
```

GET /recurring-transaction/{id}

Opis

Zwraca szczegóły cyklicznej transakcji.

Przykładowe zapytanie

```
GET /recurring-transaction/301
```

Przykładowa odpowiedź (Response Body)

```
{
  "transactionId": 301,
  "userId": 5,
  "name": "Abonament Spotify",
  "amount": 29.99,
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "isIncome": false,
  "firstPaymentDate": "2025-05-01",
  "nextPaymentDate": "2025-06-01",
  "finalPaymentDate": "2025-12-01",
  "interval": "MONTHLY",
  "description": "Miesięczna subskrypcja"
}
```

PUT /recurring-transaction/{id}

Opis

Aktualizuje dane cyklicznej transakcji.

Przykładowe zapytanie (Request Body)

```
{
  "userId": 5,
  "name": "Abonament Spotify (rodzinny)",

```

```
"amount": 39.99,
"currencyId": 1,
"isIncome": false,
"interval": "P1M",
"firstPaymentDate": "2025-05-01",
"finalPaymentDate": "2025-11-01",
"description": "Subskrypcja rodzinna"
}
```

Walidacja

- `userId(Integer, wymagane)`: Nie może być null, musi być dodatnie.
- `name(String, wymagane)`: Nie może być puste, maksymalnie 512 znaków.
- `amount(BigDecimal, wymagane)`: Nie może być null, musi być dodatnie.
- `currencyId(Integer, wymagane)`: Nie może być null.
- `isIncome(Boolean, wymagane)`: Nie może być null.
- `interval (String, wymagane)`: Nie może być puste, wymagany poprawny format okresu.
- `firstPaymentDate(Date, wymagane)`: Nie może być null, data po 2000 roku.
- `finalPaymentDate(Date, wymagane)`: Nie może być null, data po 2000 roku.
- `description(String, opcjonalne)`: Maksymalnie 512 znaków.

Przykładowa odpowiedź (Response Body)

```
{
  "transactionId": 301,
  "userId": 5,
  "name": "Abonament Spotify (rodzinny)",
  "amount": 39.99,
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "isIncome": false,
  "firstPaymentDate": "2025-05-01",
  "nextPaymentDate": "2025-06-01",
  "finalPaymentDate": "2025-11-01",
  "interval": "MONTHLY",
  "description": "Subskrypcja rodzinna"
}
```

DELETE /recurring-transaction/{id}

Opis

Usuwa cykliczną transakcję.

Przykładowe zapytanie

```
DELETE /recurring-transaction/301
```

Przykładowa odpowiedź

Status: 204 No Content

Subplan

POST /subplan

Opis

Tworzy nowe pod-marzenie w ramach marzenia.

Przykładowe zapytanie (Request Body)

```
{
  "userId": 5,
  "name": "Zakupy sprzętu",
  "amount": 1200.00,
  "currencyId": 1,
  "description": "Kupno nowego laptopa",
  "planId": 201
}
```

Walidacja

- `userId(Integer, wymagane)`: Nie może być null, musi być dodatnie.
- `name(String, wymagane)`: Nie może być puste, maksymalnie 512 znaków.
- `amount(BigDecimal, wymagane)`: Nie może być null, musi być dodatnie.
- `currencyId(Integer, wymagane)`: Nie może być null.
- `description(String, opcjonalne)`: Maksymalnie 512 znaków.
- `planId(Integer, wymagane)`: Nie może być null, musi być dodatnie.

Przykładowa odpowiedź (Response Body)

```
{
  "planId": 201,
  "subplanId": 501,
  "name": "Zakupy sprzętu",
  "description": "Kupno nowego laptopa",
  "amount": 1200.00,
  "currency": {
```

```
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "completed": false,
  "actualAmount": 0.00,
  "canBeCompleted": false,
  "date": "2025-05-05",
  "estimatedCompletionDate": "2025-05-05"
}
```

GET /subplan/{id}

Opis

Zwraca szczegóły pod-marzenia.

Przykładowe zapytanie

```
GET /subplan/501
```

Przykładowa odpowiedź (Response Body)

```
{
  "planId": 201,
  "subplanId": 501,
  "name": "Zakupy sprzętu",
  "description": "Kupno nowego laptopa",
  "amount": 1200.00,
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "completed": false,
  "actualAmount": 0.00,
  "canBeCompleted": false,
  "date": "2025-05-05",
  "estimatedCompletionDate": "2025-05-05"
}
```

PUT /subplan/{id}

Opis

Aktualizuje dane pod-marzenia.

Przykładowe zapytanie (Request Body)

```
{
  "userId": 5,
  "name": "Zakupy sprzętu",
  "amount": 1300.00,
  "currencyId": 1,
  "description": "Kupno laptopa + myszki",
  "planId": 201
}
```

Walidacja

- `userId(Integer, wymagane)`: Nie może być null, musi być dodatnie.
- `name(String, wymagane)`: Nie może być puste, maksymalnie 512 znaków.
- `amount(BigDecimal, wymagane)`: Nie może być null, musi być dodatnie.
- `currencyId(Integer, wymagane)`: Nie może być null.
- `description(String, opcjonalne)`: Maksymalnie 512 znaków.
- `planId(Integer, wymagane)`: Nie może być null, musi być dodatnie.

Przykładowa odpowiedź (Response Body)

```
{
  "planId": 201,
  "subplanId": 501,
  "name": "Zakupy sprzętu",
  "description": "Kupno laptopa + myszki",
  "amount": 1300.00,
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "completed": false,
  "actualAmount": 0.00,
  "canBeCompleted": false,
  "date": "2025-05-05",
  "estimatedCompletionDate": "2025-05-05"
}
```

POST /subplan/complete/{id}

Opis

Oznacza pod-marzenie jako zakończone.

Przykładowe zapytanie

```
POST /subplan/complete/501
```

Przykładowa odpowiedź (Response Body)

```
{
  "planId": 201,
  "subplanId": 501,
  "name": "Zakupy sprzętu",
  "description": "Kupno nowego laptopa",
  "amount": 1200.00,
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "completed": true,
  "actualAmount": 1200.00,
  "canBeCompleted": true,
  "date": "2025-05-05",
  "estimatedCompletionDate": "2025-05-05"
}
```

Ustawienie innej daty

Aby ustawić inną datę realizacji marzenia niż bieżący dzień, można opcjonalnie przekazać parametr `completionDate` w zapytaniu:

```
POST /subplan/complete/201?completionDate=2025-04-04
```

POST /subplan/uncomplete/{id}

Opis

Cofa oznaczenie pod-marzenia jako zakończone.

Przykładowe zapytanie

```
POST /subplan/uncomplete/501
```

Przykładowa odpowiedź (Response Body)

```
{
  "planId": 201,
  "subplanId": 501,
  "name": "Zakupy sprzętu",
  "description": "Kupno nowego laptopa",
  "amount": 1200.00,
  "currency": {
```

```
    "currencyId": 1,
    "isoCode": "PLN"
  },
  "completed": true,
  "actualAmount": 1200.00,
  "canBeCompleted": true,
  "date": "2025-05-05",
  "estimatedCompletionDate": "2025-05-05"
}
```

GET /subplan/all/{planId}

Opis

Zwraca wszystkie pod-marzenia w ramach danego marzenia.

Przykładowe zapytanie

```
GET /subplan/all/201
```

Przykładowa odpowiedź (Response Body)

```
[
  {
    "planId": 201,
    "subplanId": 501,
    "name": "Zakupy sprzętu",
    "description": "Kupno nowego laptopa",
    "amount": 1200.00,
    "currency": {
      "currencyId": 1,
      "isoCode": "PLN"
    },
    "completed": false,
    "actualAmount": 0.00,
    "canBeCompleted": false,
    "date": "2025-05-05",
    "estimatedCompletionDate": "2025-05-05"
  }
]
```

DELETE /subplan/{id}

Opis

Usuwa pod-marzenie.

Przykładowe zapytanie

```
DELETE /subplan/501
```

Przykładowa odpowiedź

```
Status: 204 No Content
```

Usuwanie powiązanej transakcji

Parametr `deleteTransaction` jest opcjonalny i określa, czy wraz ze składową marzenia ma zostać usunięta również powiązana z nią transakcja — jeśli zostanie ustawiony na `true`, transakcja zostanie usunięta.

```
DELETE /subplan/501?deleteTransaction=true
```

User

GET /user/account

Opis

Zwraca dane aktualnie zalogowanego użytkownika.

Przykładowe zapytanie

```
GET /user/account
```

Przykładowa odpowiedź (Response Body)

```
{
  "userId": 5,
  "firstName": "Jan",
  "lastName": "Kowalski",
  "email": "jankowalski@example.com",
  "phoneNumber": "+48123456789",
  "dateOfJoining": "2023-03-15",
  "currentBudget": 2500.00,
  "currency": {
    "currencyId": 1,
    "isoCode": "PLN"
  },
}
```



```
"darkMode": true,  
"notificationsByEmail": true,  
"notificationsByPhone": false  
}
```

POST /user/currency

Opis

Dodaje dostępne waluty do konta użytkownika.

Przykładowe zapytanie

```
POST /user/currency
```

Przykładowa odpowiedź

```
Status: 204 No Content
```