API DOC for ATBANKApplication with request/response and samples

LOGIN APIs

1) POST /login

Description :- This API is used to login.

API Fields :-

API	To login
Method	POST
Request	/login
Status	200

Body Fields :-

Field	Type	Mandatory	Description
username	String	Y	Username for the account
password	String	Y	Password for the account

Success Response Fields :-

Field	Type	Mandatory	Description
userId	Long	Y	UserId
user	User object	Y	User details
accountType	String	Y	Account Type i.e. SAVINGS, CURRENT etc
accountStatu	String	Y	Account Status i.e.

s			ACTIVE, CLOSED etc
createdOn	Date	Y	Date on which account got created
balance	BigDecimal	Y	Balance in that account

Failure Response Fields :-

Field	**Description**
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :- http://localhost:8080/login
Body :- {"username":"kakati","password":"jakati"}

Success Response :-

```
{
   "id": 55,
   "user": {
      "id": 69,
      "username": "kakati"
   },
   "accountType": "SAVINGS",
   "accountStatus": "ACTIVE",
   "createdOn": "2018-07-05T08:26:17.026+0000",
   "balance": 200
}
```

Failure Response :-

Body :- "username":"kakati","password":"jakati"}

```
{
    "message": "Invalid username or password.",
    "code": "PRECONDITION_FAILED"
}
```

USER APIs

2) POST /api/users

Description :- This API is used to create an user.

API Fields :-

API	To create an user
Method	POST
Request	/api/users
Status	200

Body Fields :-

Field	**Type**	**Mandatory**	**Description**
username	String	Y	Username for the account
password	String	Y	Password for the account

Success Response Fields :-

Field	**Type**	**Description**

id	Long	Id for the user
username	String	Username of the user

Failure Response Fields :-

Field	**Description**
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :- http://localhost:8080/api/users
Body :- {"username":"kakati","password":"jakati"}

Success Response :-

```
{
   "id": 91,
   "username": "kakati"
}
```

Failure Response :-
Body :- "username":"kakati","password":"jakati"}

```
{
   "message": "The username kakati is already exists, Please select a different username.",
   "code": "PRECONDITION_FAILED"
}
```

3) PUT /api/users

Description :- This API is used to update an user.

API Fields :-

API	To update an user
Method	PUT
Request	/api/users
Status	200

Body Fields :-

Field	**Type**	**Mandatory**	**Description**
username	String	Y	Username for the account
password	String	Y	Password for the account

Success Response Fields :-

Field	**Type**	**Description**
id	Long	Id for the user
username	String	Username of the user

Failure Response Fields :-

Field	**Description**
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :- http://localhost:8080/api/users
Body :- {"username":"kakati","password":"qwertyakati"}

Success Response :-

```
{
   "id": 91,
   "username": "kakati"
}
```

Failure Response :-
Body :- "username":"kakatqi","password":"jakati"}

```
{
   "message": "The username kakatqi does not exists.",
   "code": "PRECONDITION_FAILED"
}
```

4) DELETE /api/users

Description :- This API is used to delete an user.

API Fields :-

API	To delete an user
Method	DELETE
Request	/api/users
Status	200

Body Fields :-

Field	**Type**	**Mandatory**	**Description**
username	String	Y	Username for the account

password	String	Y	Password for the account

Success Response Fields :-

Field	Type	Description
id	Long	Id for the user
username	String	Username of the user

Failure Response Fields :-

Field	Description
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :- http://localhost:8080/api/users
Body :- {"username":"kakati","password":"qwertyakati"}

Success Response :-

```
{
   "id": 91,
   "username": "kakati"
}
```

Failure Response :-

Body :- "username":"kakatqi","password":"jakati"}

```
{
   "message": "The username kakatqi does not exists.",
   "code": "PRECONDITION_FAILED"
}
```

ACCOUNT APIs

5) **POST /api/accounts**

Description :- This API is used to create an account.

API Fields :-

API	To create an account
Method	POST
Request	/api/accounts
Status	200

Body Fields :-

Field	**Type**	**Description**
userId	Long	User Id for which the account need to be created
amount	BigDecimal	Initial amount

Success Response Fields :-

Field	**Type**	**Mandatory**	**Description**
userId	Long	Y	UserId for which the account has to be created
user	User object	Y	User details

accountType	String	Y	Account Type i.e. SAVINGS, CURRENT etc
accountStatus	String	Y	Account Status i.e. ACTIVE, CLOSED etc
createdOn	Date	Y	Date on which account got created
balance	BigDecimal	Y	Balance in that account

Failure Response Fields :-

Field	**Description**
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :- http://localhost:8080/api/accounts
Body :- {"userId":"69","amount":"200"}

Success Response :-

```
{
   "id": 55,
   "user": {
      "id": 69,
      "username": "kakati"
   },
   "accountType": "SAVINGS",
   "accountStatus": "ACTIVE",
   "createdOn": "2018-07-05T08:26:17.026+0000",
   "balance": 200
}
```

Failure Response :-

Body :- {"userId":"9","amount":"200"}

```
{
   "message": "The user id 9 does not exists.",
   "code": "PRECONDITION_FAILED"
}
```

6) PUT /api/accounts

Description :- This API is used to update an account.

API Fields :-

API	To update an account
Method	PUT
Request	/api/accounts
Status	200

Body Fields :-

Field	**Type**	**Description**
accountId	Long	Account Id for updation
accountStatus	String	Status to be changed for the account

Success Response Fields :-

Field	Type	Mandatory	Description
userId	Long	Y	UserId for which the account has to be created
user	User object	Y	User details
accountType	String	Y	Account Type i.e. SAVINGS, CURRENT etc
accountStatus	String	Y	Account Status i.e. ACTIVE, CLOSED etc
createdOn	Date	Y	Date on which account got created
balance	BigDecimal	Y	Balance in that account

Failure Response Fields :-

Field	Description
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :- http://localhost:8080/api/accounts
Body :-{"accountId":"55","accountStatus":"CLOSED"}

Success Response :-

```
{
   "id": 55,
   "user": {
      "id": 69,
      "username": "kakati"
```

```
    },
    "accountType": "SAVINGS",
    "accountStatus": "CLOSED",
    "createdOn": "2018-07-05T08:26:17.026+0000",
    "balance": 200
}
```

Failure Response :-

Body :- {"accountId":"9","accountStatus":"CLOSED"}

```
{
    "message": "The account id 9 does not exists.",
    "code": "PRECONDITION_FAILED"
}
```

7) GET /api/accounts/<accountId>

Description :- This API is used to update an account.

API Fields :-

API	To get details of an account
Method	GET
Request	/api/accounts
Status	200

Path Fields :-

Field	Type	Description
accountId	Long	Account Id to get details

Success Response Fields :-

Field	Type	Mandatory	Description
userId	Long	Y	UserId for which the account has to be created
user	User object	Y	User details
accountType	String	Y	Account Type i.e. SAVINGS, CURRENT etc
accountStatus	String	Y	Account Status i.e. ACTIVE, CLOSED etc
createdOn	Date	Y	Date on which account got created
balance	BigDecimal	Y	Balance in that account

Failure Response Fields :-

Field	Description
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :- http://localhost:8080/api/accounts/55

Success Response :-

```
[
   {
      "id": 55,
      "user": {
         "id": 69,
         "username": "kakati"
      },
      "accountType": "SAVINGS",
      "accountStatus": "CLOSED",
      "createdOn": "2018-07-05T08:26:17.000+0000",
      "balance": 200
   }
]
```

Failure Response :-

```
{
   "message": "The account id 9 does not exists.",
   "code": "PRECONDITION_FAILED"
}
```

TRANSACTION APIs

8) POST /api/transactions/credit

Description :- This API is used to credit an amount to an account

API Fields :-

API	To credit an amount to an account
Method	POST
Request	/api/transactions/credit

Status	200

Body Fields :-

Field	Type	Description
accountId	Long	Account Id for which credit has to be made
amount	BigDecimal	Amount to be credited

Success Response Fields :-

Field	Type	Mandatory	Description
userId	Long	Y	UserId
user	User object	Y	User details
accountType	String	Y	Account Type i.e. SAVINGS, CURRENT etc
accountStatus	String	Y	Account Status i.e. ACTIVE, CLOSED etc
createdOn	Date	Y	Date on which account got created
balance	BigDecimal	Y	Balance in that account

Failure Response Fields :-

Field	Description
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :-   http://localhost:8080/api/transactions/credit
Body :- {"accountId":"55","amount":"50"}

Success Response :-

```
{
   "id": 55,
   "user": {
      "id": 69,
      "username": "kakati"
   },
   "accountType": "SAVINGS",
   "accountStatus": "ACTIVE",
   "createdOn": "2018-07-05T08:26:17.000+0000",
   "balance": 250
}
```

Failure Response :-

Body :- {"accountId":"25","amount":"200"}

```
{
   "message": "The Account with id 25 does not exists.",
   "code": "PRECONDITION_FAILED"
}

{
   "message": "Negative amount cannot be added.",
   "code": "PRECONDITION_FAILED"
}
```

9) POST /api/transactions/debit

Description :- This API is used to debit an amount from an account

API Fields :-

API	To debit an amount from an account
Method	POST
Request	/api/transactions/debit
Status	200

Body Fields :-

Field	**Type**	**Description**
accountId	Long	Account Id for which debit has to be made
amount	BigDecimal	Amount to be debited

Success Response Fields :-

Field	**Type**	**Mandatory**	**Description**
userId	Long	Y	UserId
user	User object	Y	User details
accountType	String	Y	Account Type i.e. SAVINGS, CURRENT etc
accountStatus	String	Y	Account Status i.e. ACTIVE, CLOSED etc
createdOn	Date	Y	Date on which account got created
balance	BigDecimal	Y	Balance in that account

Failure Response Fields :-

Field	Description
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :- http://localhost:8080/api/transactions/debit
Body :- {"accountId":"55","amount":"50"}

Success Response :-

```
{
   "id": 55,
   "user": {
      "id": 69,
      "username": "kakati"
   },
   "accountType": "SAVINGS",
   "accountStatus": "ACTIVE",
   "createdOn": "2018-07-05T08:26:17.000+0000",
   "balance": 200
}
```

Failure Response :-

Body :- {"accountId":"25","amount":"200"}

```
{
   "message": "The Account with id 25 does not exists.",
   "code": "PRECONDITION_FAILED"
}

{
   "message": "Negative amount cannot be added.",
   "code": "PRECONDITION_FAILED"
}
```

10) POST /api/transactions/transfer

Description :- This API is used to transfer an amount from account to another account

API Fields :-

API	To debit an amount from an account
Method	POST
Request	/api/transactions/transfer
Status	200

Body Fields :-

Field	Type	Description
accountId	Long	Account Id for which debit has to be made
toAccountId	Long	Account Id for which credit has to be made
amount	BigDecimal	Amount to be debited

Success Response Fields :-

Field	Type	Mandatory	Description
userId	Long	Y	UserId
user	User object	Y	User details
accountType	String	Y	Account Type i.e. SAVINGS, CURRENT etc
accountStatus	String	Y	Account Status i.e. ACTIVE, CLOSED etc
createdOn	Date	Y	Date on which account got created

balance	BigDecimal	Y	Balance in that account

Failure Response Fields :-

Field	Description
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :- http://localhost:8080/api/transactions/transfer
Body :- {"accountId":"55","toAccountId":"54","amount":"50"}

Success Response :-

```
{
   "id": 55,
   "user": {
      "id": 69,
      "username": "kakati"
   },
   "accountType": "SAVINGS",
   "accountStatus": "ACTIVE",
   "createdOn": "2018-07-05T08:26:17.000+0000",
   "balance": 150
}
```

Failure Response :-

Body :- {"accountId":"55","toAccountId":"53","amount":"50"}

```
{
   "message": "The Account with id 53 does not exists.",
   "code": "PRECONDITION_FAILED"
}
```

```
{
    "message": "Negative amount cannot be added.",
    "code": "PRECONDITION_FAILED"
}
```

11) GET /api/transactions/accounts/<accountId>

Description :- This API is used to get all the transaction for the account

API Fields :-

API	To get all the transaction for the account
Method	GET
Request	/api/transactions/accounts/<accountId>
Status	200

Path Fields :-

Field	**Type**	**Description**
accountId	Long	Account Id to get details

Request Parameters :-

Field	**Type**	**Mandatory**	**Description**
from	Long (in millies)	N	Filter list on submitted date from
to	Long (in millies)	N	Filter list on submitted date to
size	Integer **(Default 10)**	N	Size of the list

Success Response Fields :-

Field	**Type**	**Mandatory**	**Description**
id	Long	Y	Transaction id
accountId	Long	Y	Account id
transactionType	String	Y	Transaction type i.e. CREDIT, DEBIT
createdOn	Date	Y	Date on which transaction got created

Failure Response Fields :-

Field	**Description**
code	Code for the exception occured
message	Error message

Sample Request and Response :-

Request :- http://localhost:8080/api/transactions/accounts/55

Success Response :-

```
[
   {
      "id": 5,
      "accountId": 55,
      "body": "The amount 50 was debited  from the account id 55",
      "transactionType": "DEBIT",
      "transactionOn": "2018-07-05T08:59:22.000+0000"
   },
   {
      "id": 4,
      "accountId": 55,
      "body": "The amount 50 was debited  from the account id 55",
      "transactionType": "DEBIT",
      "transactionOn": "2018-07-05T08:56:12.000+0000"
```

```
    },
    {
        "id": 3,
        "accountId": 55,
        "body": "The amount 50 was added to the account id 55",
        "transactionType": "CREDIT",
        "transactionOn": "2018-07-05T08:47:02.000+0000"
    },
    {
        "id": 2,
        "accountId": 55,
        "body": "A new Account has been created with id 55 and amount with 200",
        "transactionType": "CREDIT",
        "transactionOn": "2018-07-05T08:26:17.000+0000"
    }
]
```

Failure Response :-

Body :- {"accountId":"55","toAccountId":"53","amount":"50"}

```
[
    {
        "message": "The Account with id 53 does not exists.",
        "code": "PRECONDITION_FAILED"
    }
]
```