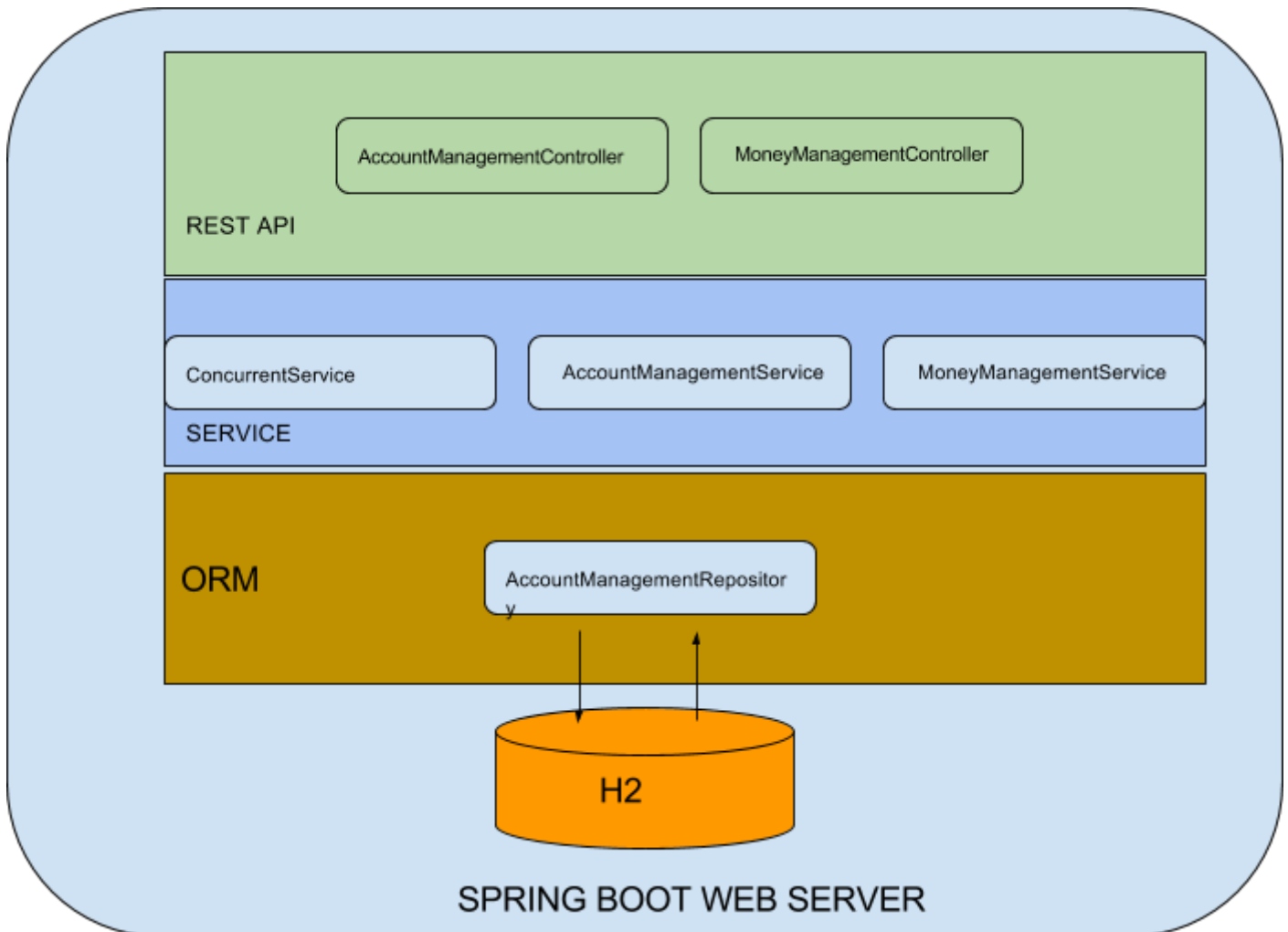


## Project Guide.

1. Technology.
  - Java 8, Spring boot, Hibernate, h2 in-memory database.
  - Junit4, Mockito.
2. Solution Design overview.



**AccountManagementController** provide api to create new account.

**MoneyManagementController** provide transferring api.

**AccountManagementService** provide method to create and get account.

**MoneyManagementService** contain transferring implementation.

**ConcurrentService** handle concurrent request , provide locking mechanism and deadlock preventing.

Brief of implementation:

- Create Account table with field (id,name,balance). Name is not duplicated and balance can not be negative.
- Deduce balance from sender account and credit to receiver account directly when transferring.
- create a cache to store accountid and lock object,each accountid has a lock object, before start transaction I lock 2 account (from and to) and release them when transaction is finished to avoid conflict data between many transactions.

3. Limitation

- when locking 2 account at the same time, need to check carefully when using 2 those locks object in other codes to avoid deadlock.
- when updating directly balance to Account table, that make difficult to validate data in future.
- when a transaction update wrong balance we don't know how to verify balance is correct or not.
- can improve by creat transaction table store all transaction created for all accounts

#### 4. Deploy.

Step 1 build project using maven "mvn clean install"

Step 2 create /deploy folder

Step 3 go to target folder copy ingenico-project-spring-boot.jar to /deploy folder.

Step 4 go to /deploy folder and run java command "java -jar ingenico-project-spring-boot.jar"

#### 5. Test.

API	Create new account.
URL	<a href="http://localhost:8080/ingenico/account">http://localhost:8080/ingenico/account</a>
HTTP METHOD	POST
Header	[{"key": "Content-Type", "value": "application/json", "description": ""}]
Body	{"id": null, "balance": 100.0, "name": "test"}
Response OK	{ "status": 1, "error": "Request successful, no exception" }
Response Error	{ "status": 0, "error": "<Detail>" }
Error messages	Account name does exist. Balance must be positive number.

API	Transfer money
URL	<a href="http://localhost:8080/ingenico/transfer">http://localhost:8080/ingenico/transfer</a>
HTTP METHOD	POST
Header	[{"key": "Content-Type", "value": "application/json", "description": ""}]
Body	{"fromAccountId": 4, "toAccountId": 5, "amount": 100.0}
Response OK	{       "status": 1,       "error": "Request successful, no exception"     }
Response Error	{       "status": 0,       "error": "<Detail>"     }
Error messages	Account id <id> does not exist Account balance is insufficient for this transaction. Can not transfer for itself.

6. Database verify

Go to <http://localhost:8080/console/>

Login with this details

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

User Name: sa

Password:

Connect Test Connection