

Creating microservices for account and loan

```
package com.example.account_service.controller;
```

```
import com.example.account_service.entity.Account;
```

```
import com.example.account_service.service.AccountService;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.web.bind.annotation.*;
```

```
import java.util.List;
```

```
@RestController
```

```
@RequestMapping("/api/accounts")
```

```
public class AccountController {
```

```
    @Autowired
```

```
    private AccountService service;
```

```
    @PostMapping
```

```
    public Account createAccount(@RequestBody Account account) {
```

```
        return service.createAccount(account);
```

```
    }
```

```
    @GetMapping
```

```
    public List<Account> getAllAccounts() {
```

```
        return service.getAllAccounts();
```

```
    }
```

```
    @GetMapping("/{id}")
```

```
    public Account getAccountById(@PathVariable Long id) {  
        return service.getAccountById(id).orElse(null);  
    }  
}
```

```
package com.example.account_service.entity;
```

```
import jakarta.persistence.*;
```

```
import lombok.*;
```

```
@Entity
```

```
@Data
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
public class Account {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Long accountId;
```

```
    private String customerName;
```

```
    private String accountType;
```

```
    private Double balance;
```

```
    public Long getAccountId() {
```

```
        return accountId;
```

```
    }
```

```
    public void setAccountId(Long accountId) {
```

```
        this.accountId = accountId;
```

```
    }
```

```
    public String getCustomerName() {
```

```

        return customerName;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    public String getAccountType() {
        return accountType;
    }

    public void setAccountType(String accountType) {
        this.accountType = accountType;
    }

    public Double getBalance() {
        return balance;
    }

    public void setBalance(Double balance) {
        this.balance = balance;
    }
}

package com.example.account_service.repository;

import com.example.account_service.entity.Account;
import org.springframework.data.jpa.repository.JpaRepository;

public interface AccountRepository extends JpaRepository<Account, Long> {}

package com.example.account_service.service;

import com.example.account_service.entity.Account;
import com.example.account_service.repository.AccountRepository;

```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import java.util.List;
import java.util.Optional;
```

```
@Service
```

```
public class AccountService {
```

```
    @Autowired
```

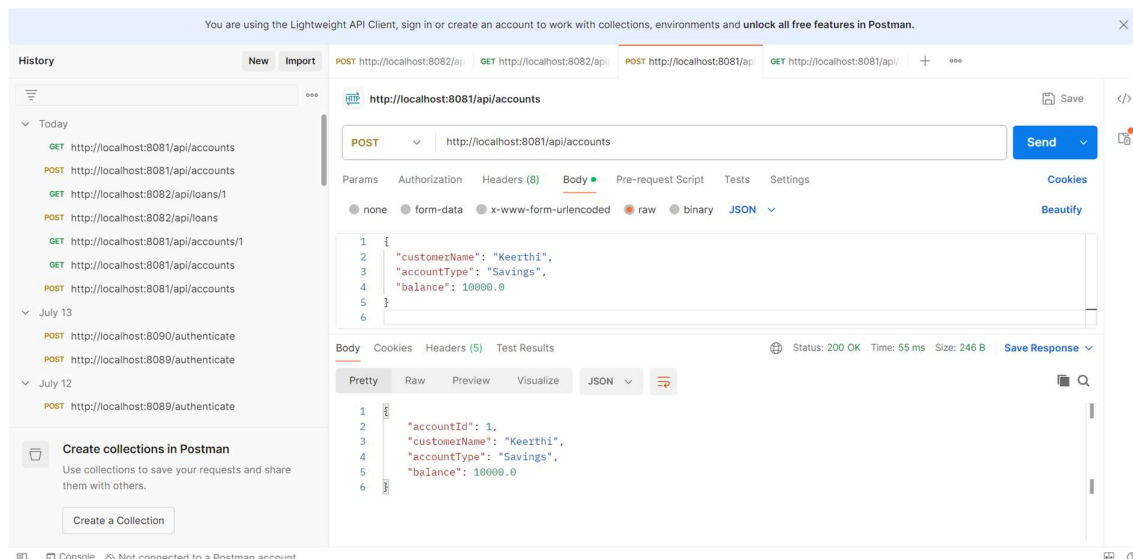
```
    private AccountRepository repository;
```

```
    public Account createAccount(Account account) {
        return repository.save(account);
    }
```

```
    public List<Account> getAllAccounts() {
        return repository.findAll();
    }
```

```
    public Optional<Account> getAccountById(Long id) {
        return repository.findById(id);
    }
}
```

OUTPUT:



Loan service:

```
package com.example.loan_service.controller;
```

```
import com.example.loan_service.entity.Loan;
```

```
import com.example.loan_service.service.LoanService;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.web.bind.annotation.*;
```

```
import java.util.List;
```

```
@RestController
```

```
@RequestMapping("/api/loans")
```

```
public class LoanController {
```

```
    @Autowired
```

```
    private LoanService service;
```

```
    @PostMapping
```

```

    public Loan createLoan(@RequestBody Loan loan) {
        return service.createLoan(loan);
    }

    @GetMapping
    public List<Loan> getAllLoans() {
        return service.getAllLoans();
    }

    @GetMapping("/{id}")
    public Loan getLoanById(@PathVariable Long id) {
        return service.getLoanById(id).orElse(null);
    }
}

package com.example.loan_service.entity;

import jakarta.persistence.*;
import lombok.*;

@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Loan {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long loanId;

```

```
private String customerName;

private String loanType;

private Double loanAmount;

    public Long getLoanId() {

        return loanId;

    }

    public void setLoanId(Long loanId) {

        this.loanId = loanId;

    }

    public String getCustomerName() {

        return customerName;

    }

    public void setCustomerName(String customerName) {

        this.customerName = customerName;

    }

    public String getLoanType() {

        return loanType;

    }

    public void setLoanType(String loanType) {

        this.loanType = loanType;

    }

    public Double getLoanAmount() {

        return loanAmount;

    }

    public void setLoanAmount(Double loanAmount) {

        this.loanAmount = loanAmount;

    }

}
```

```
package com.example.loan_service.repository;
```

```
import com.example.loan_service.entity.Loan;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface LoanRepository extends JpaRepository<Loan, Long> {}
```

```
package com.example.loan_service.service;
```

```
import com.example.loan_service.entity.Loan;
```

```
import com.example.loan_service.repository.LoanRepository;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Service;
```

```
import java.util.List;
```

```
import java.util.Optional;
```

```
@Service
```

```
public class LoanService {
```

```
    @Autowired
```

```
    private LoanRepository repository;
```

```
    public Loan createLoan(Loan loan) {
```

```
        return repository.save(loan);
```

```
    }
```

```
    public List<Loan> getAllLoans() {
```

```
        return repository.findAll();
```



```
}
```

```
public Optional<Loan> getLoanById(Long id) {
```

```
    return repository.findById(id);
```

```
}
```

```
}
```

OUTPUT:

The screenshot shows the Postman interface with a GET request to `http://localhost:8082/api/loans/1` successfully executed. The response is a JSON object containing loan information.

Request: GET `http://localhost:8082/api/loans/1`

Response: Status: 200 OK, Time: 79 ms, Size: 246 B

JSON Response:

```
{
  "loanId": 1,
  "customerName": "Keeirthi",
  "loanType": "Home Loan",
  "loanAmount": 500000.0
}
```