

Creating Microservices for account and loan:

Code:

AccountApplication.java:

```
package com.cognizant.account.controller;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
```

```
public class AccountApplication {  
    public static void main(String[] args) {  
        SpringApplication.run(AccountApplication.class, args);  
    }  
}
```

AccountController.java:

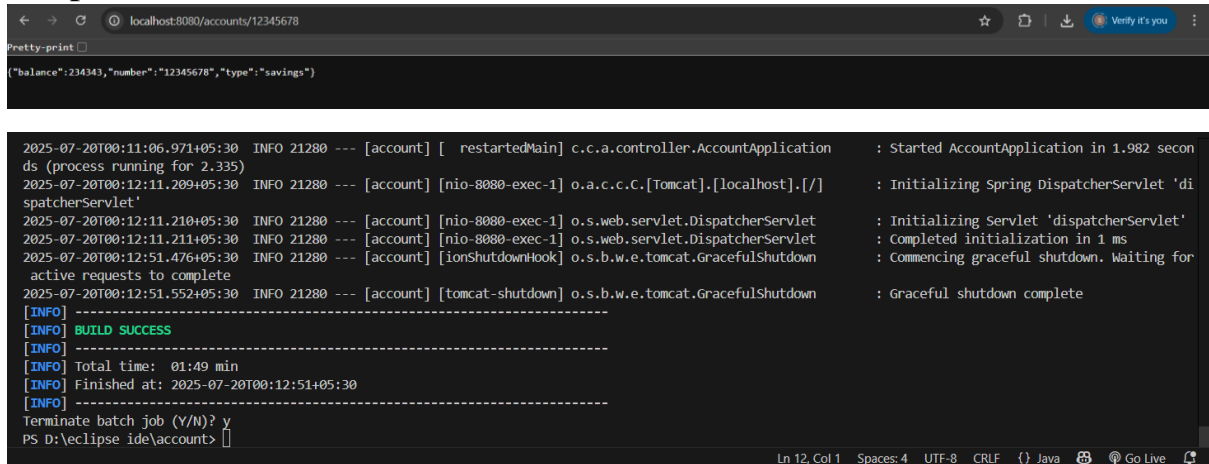
```
package com.cognizant.account.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RestController;  
import java.util.Map;
```

```
@RestController
```

```
public class AccountController {  
    @GetMapping("/accounts/{number}")  
    public Map<String, Object> getAccount(@PathVariable String number) {  
        return Map.of(  
            "number", number,  
            "type", "savings",  
            "balance", 234343  
        );  
    }  
}
```

```
}  
  
}
```

Output:



The screenshot shows a web browser at localhost:8080/accounts/12345678 displaying a JSON response: {"balance":234343,"number":"12345678","type":"savings"}. Below the browser, a terminal window shows the application's startup logs, including INFO messages for restarting the main application, initializing Spring DispatcherServlet, and completing initialization. The terminal also shows the graceful shutdown process and a successful build message.

```
2025-07-20T00:11:06.971+05:30 INFO 21280 --- [account] [ restartedMain] c.c.a.controller.AccountApplication : Started AccountApplication in 1.982 seconds (process running for 2.335)
2025-07-20T00:12:11.209+05:30 INFO 21280 --- [account] [nio-8080-exec-1] o.a.c.c.c.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-07-20T00:12:11.210+05:30 INFO 21280 --- [account] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-07-20T00:12:11.211+05:30 INFO 21280 --- [account] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
2025-07-20T00:12:51.476+05:30 INFO 21280 --- [account] [ionShutdownHook] o.s.b.w.e.tomcat.GracefulShutdown : Commencing graceful shutdown. Waiting for active requests to complete
2025-07-20T00:12:51.552+05:30 INFO 21280 --- [account] [tomcat-shutdown] o.s.b.w.e.tomcat.GracefulShutdown : Graceful shutdown complete
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:49 min
[INFO] Finished at: 2025-07-20T00:12:51+05:30
[INFO] -----
Terminate batch job (Y/N)? y
PS D:\eclipse_ide\account>
```

CODE:

EurekaDiscoveryServerApplication.java:

```
package com.cognizant.eureka_discovery_server;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication
@EnableEurekaServer

public class EurekaDiscoveryServerApplication {

    public static void main(String[] args) {

        SpringApplication.run(EurekaDiscoveryServerApplication.class, args);

    }

}
```

AccountController.java:

```
package com.cognizant.account.controller;

import org.springframework.web.bind.annotation.*;
import java.util.*;

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

```

public class AccountController {

    @GetMapping("/{number}")

    public Map<String, Object> getAccount(@PathVariable String number) {

        Map<String, Object> response = new HashMap<>();

        response.put("number", number);

        response.put("type", "savings");

        response.put("balance", 234343);

        return response;

    }

}

```

LoanController.java:

```

package com.cognizant.loan.controller;

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

import java.util.*;

@RestController

@RequestMapping("/loans")

public class LoanController {

    @GetMapping("/{number}")

    public Map<String, Object> getLoan(@PathVariable String number) {

        Map<String, Object> response = new HashMap<>();

        response.put("number", number);

        response.put("type", "car");

        response.put("loan", 400000);

        response.put("emi", 3258);

        response.put("tenure", 18);

        return response;

    }

}

```

GreetController.java:

```

package com.cognizant.greet_service.controller;

```

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
```

```
public class GreetController {
```

```
    @GetMapping("/greet")
```

```
    public String greet() {
```

```
        return "Hello World";
```

```
    }
```

```
}
```

LogFilter.java:

```
package com.cognizant.api_gateway.filter;
```

```
import org.springframework.cloud.gateway.filter.GlobalFilter;
```

```
import org.springframework.cloud.gateway.filter.GatewayFilterChain;
```

```
import org.springframework.core.Ordered;
```

```
import org.springframework.stereotype.Component;
```

```
import org.springframework.web.server.ServerWebExchange;
```

```
import reactor.core.publisher.Mono;
```

```
@Component
```

```
public class LogFilter implements GlobalFilter, Ordered {
```

```
    @Override
```

```
    public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {
```

```
        System.out.println("Incoming request: " + exchange.getRequest().getURI());
```

```
        return chain.filter(exchange);
```

```
    }
```

```
    @Override
```

```
    public int getOrder() {
```

```
        return -1; // highest priority
```

```
}  
  
}
```

Output:

System Status			
Environment	test	Current time	2025-07-19T21:57:03 +0530
Data center	default	Uptime	00:13
		Lease expiration enabled	true
		Renews threshold	8
		Renews (last min)	12
DS Replicas			
localhost			
Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
ACCOUNT-SERVICE	n/a (1)	(1)	UP (1) - sid-3010:account-service:8080
API-GATEWAY	n/a (1)	(1)	UP (1) - sid-3010:api-gateway:9090
GREET-SERVICE	n/a (1)	(1)	UP (1) - sid-3010:greet-service:8082
LOAN-SERVICE	n/a (1)	(1)	UP (1) - sid-3010:loan-service:8081
General Info			
Name	Value		