

```

package com.klu.controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class ArithmeticController {
    @GetMapping("/add/{A}/{B}")
    public String add(@PathVariable("A") int a, @PathVariable("B") int b) {
        return "Addition = " + (a + b);
    }

    @GetMapping("/sub/{A}/{B}")
    public String sub(@PathVariable("A") int a, @PathVariable("B") int b) {
        return "Subtraction = " + (a - b);
    }

    @GetMapping("/mul/{A}/{B}")
    public String mul(@PathVariable("A") int a, @PathVariable("B") int b) {
        return "Multiplication = " + (a * b);
    }

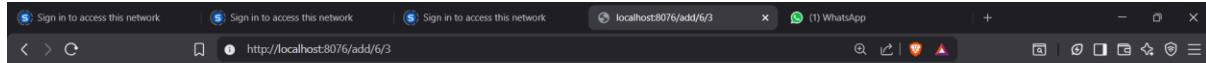
    @GetMapping("/Div/{A}/{B}")
    public String Div(@PathVariable("A") int a, @PathVariable("B") int b) {
        return "Division = " + (a / b);
    }

    @GetMapping("/Mod/{A}/{B}")
    public String Mod(@PathVariable("A") int a, @PathVariable("B") int b) {
        return "Modulus = " + (a % b);
    }
}

```

The screenshot shows the Eclipse IDE interface with the code editor displaying the above Java code. The code implements a RESTful API for arithmetic operations using Spring's annotation-based configuration. The Java code uses `@GetMapping` and `@PostMapping` annotations to map URLs to methods. It also uses `@PathVariable` to extract values from the URL and `int` to define the data type of the parameters.

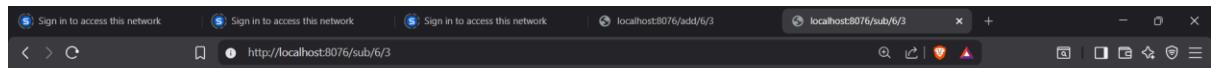
## Addition



Addition = 9



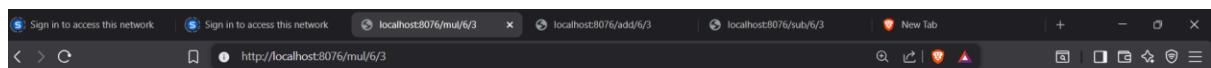
## Subtraction



Subtraction = 3



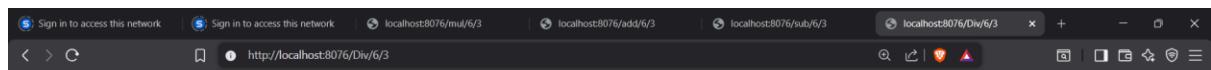
## Multiplication



Multiplication = 18



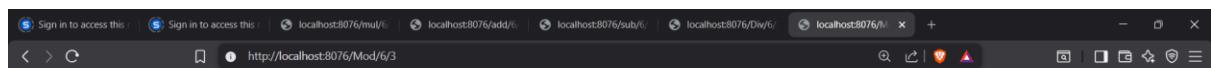
## Division



Division = 2



## Modulus/Remainder



Modulus = 0

