@GetMapping(value = "/hello/{name}", produces = "application/json")  
@Operation(summary = "Say hello to the given name")  
@ResponseStatus(HttpStatus.OK)  
public String sayHelloAlternative(@PathVariable String name) {  
 **return** **JSON\_STRING**.replace("${name}", name);  
}

Spring: Backend Technologie, Core, MVC, Persistance, SpringBoot -> Convention over config  
 Einfach zu verwenden, Auto Config, Nahtlose implementation

Dependency Inj: Arten: Konstruktor, setter, Eigenschaft  
 Trennung von Zuständigkeiten, Losegekoppeltes System, Sehr hilfreich beim testen

REST: Grundprinzipien: Eindeutige Definition von Ressourcen, Statuslose comm., Verwendung von http Standardmethoden  
 Neustart des Servers, Skalierbarkeit, High Performance

ServerSocket server = new ServerSocket(4711);  
**while** (true) {  
 Socket client = server.accept();  
 InputStream **in** = client.getInputStream();  
 OutputStream out = client.getOutputStream();  
 int factor1 = **in**.read();  
 int factor2 = **in**.read();  
 int result = factor1 \* factor2;  
 out.write(result);  
 client.close();  
}