

## Setting up the Project

### 1. Go to Spring Initializr i.e. "start.spring.io"

**Project**  
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ Java ☐ Kotlin ☐ Groovy  
☒ Maven

**Language**  
☒ Java ☐ Kotlin ☐ Groovy

**Spring Boot**  
☐ 3.3.0 (SNAPSHOT) ☐ 3.3.0 (M1) ☐ 3.2.4 (SNAPSHOT) ☒ 3.2.3  
☐ 3.1.10 (SNAPSHOT) ☐ 3.1.9

**Project Metadata**  

Group

Artifact

Name

Description

Package name

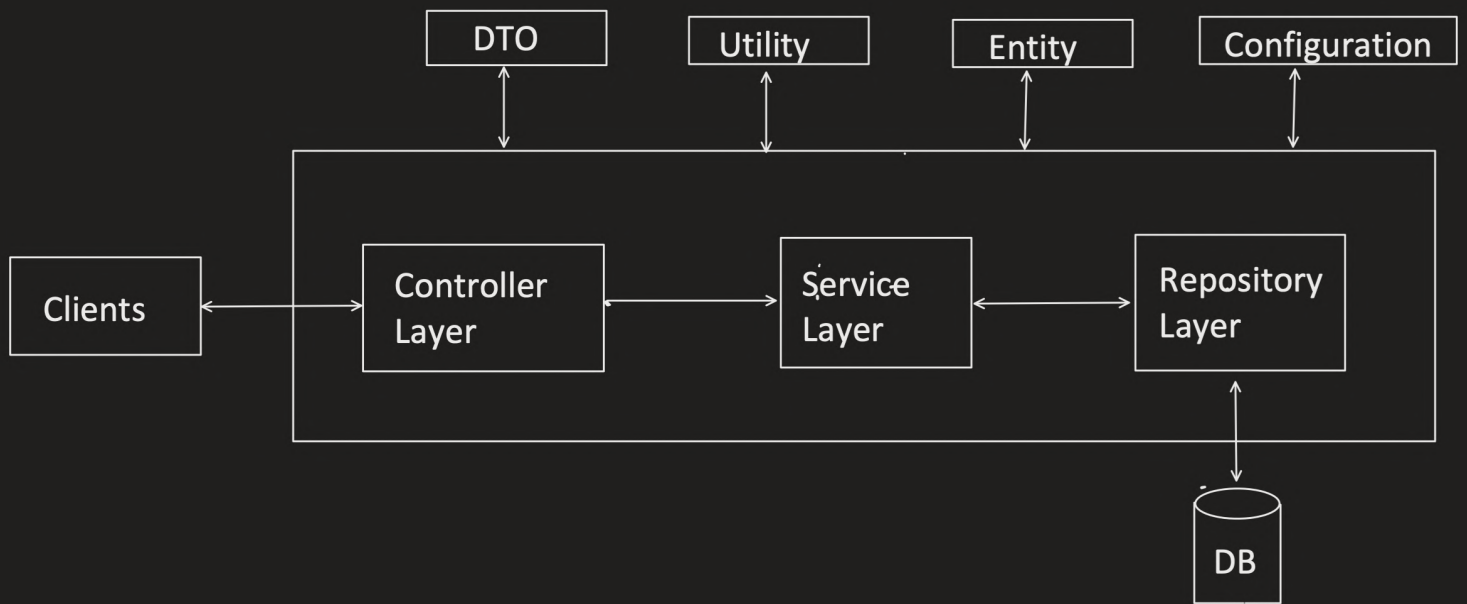
Packaging ☒ Jar ☐ War

Java ☐ 21 ☒ 17

**Dependencies** ADD DEPENDENCIES... ⌵ + B

**Spring Web** WEB  
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

## LAYERED ARCHITECTURE



```
@RestController
@RequestMapping("/payments")
public class PaymentController {

    @Autowired
    PaymentService paymentService;

    @GetMapping("/{id}")
    public ResponseEntity<PaymentResponse> getPaymentById(@PathVariable Long id) {

        //map incoming data to internal request DTO
        PaymentRequest internalRequestObj = new PaymentRequest();
        internalRequestObj.setPaymentId(id);

        //pass this internalRequestObj to further layer for processing
        PaymentResponse payment = paymentService.getPaymentDetailsById(internalRequestObj);

        //return the Response DTO
        return ResponseEntity.ok(payment);
    }
}
```

```
@Service
public class PaymentService {

    @Autowired
    PaymentRepository paymentRepository;

    public PaymentResponse getPaymentDetailsById(PaymentRequest internalRequestObj) {

        PaymentEntity paymentModel = paymentRepository.getPaymentById(internalRequestObj);

        //map it to response obj
        PaymentResponse paymentResponse = mapModelToResponseDTO(paymentModel);
        return paymentResponse;
    }

    private PaymentResponse mapModelToResponseDTO(PaymentEntity paymentEntity){

        PaymentResponse response = new PaymentResponse();
        response.setPaymentId(paymentEntity.getId());
        response.setAmount(paymentEntity.getPaymentAmount());
        response.setCurrency(paymentEntity.getPaymentCurrency());
        return response;
    }
}
```

```
@Repository
public class PaymentRepository {

    public PaymentEntity getPaymentById(PaymentRequest request){
        PaymentEntity paymentModel = executeQuery(request);
        return paymentModel;
    }

    private PaymentEntity executeQuery(PaymentRequest request){
        //connect with DB and fetch the data
        PaymentEntity payment = new PaymentEntity();
        payment.setId(request.getPaymentId());
        payment.setPaymentCurrency("INR");
        payment.setPaymentAmount(100.00);
        return payment;
    }
}
```