

Spring boot: Project Setup and Layered Architecture (Concept & Coding)

"Concept & Coding" YT Video Notes

Setting up the Project

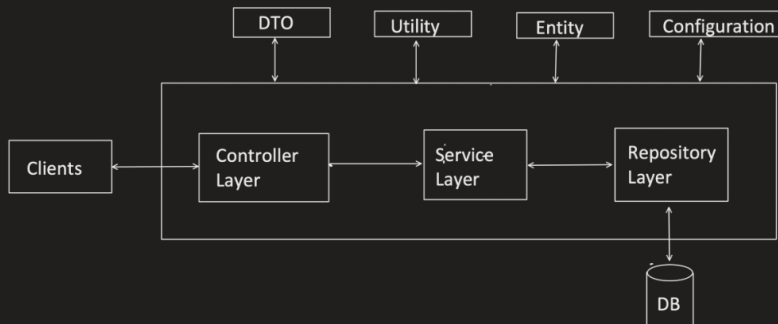
1. Go to Spring Initializr i.e. "start.spring.io"

Project <input checked="" type="radio"/> Gradle - Groovy <input type="radio"/> Gradle - Kotlin <input checked="" type="radio"/> Maven	Language <input checked="" type="radio"/> Java <input type="radio"/> Kotlin <input type="radio"/> Groovy	Dependencies ADD DEPENDENCIES... X + B
Spring Boot <input type="radio"/> 3.3.0 (SNAPSHOT) <input type="radio"/> 3.3.0 (M1) <input type="radio"/> 3.2.4 (SNAPSHOT) <input checked="" type="radio"/> 3.2.3 <input type="radio"/> 3.1.10 (SNAPSHOT) <input type="radio"/> 3.1.9		
Project Metadata		
Group: <input type="text" value="com.conceptandcoding"/>		
Artifact: <input type="text" value="learningspringboot"/>		
Name: <input type="text" value="springboot application"/>		
Description: <input type="text" value="project for learning spring boot"/>		
Package name: <input type="text" value="com.conceptandcoding.learningspringboot"/>		
Packaging: <input checked="" type="radio"/> Jar <input type="radio"/> War		
Java: <input type="radio"/> 21 <input checked="" type="radio"/> 17		

Spring Web [web](#)
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Report Abuse

LAYERED ARCHITECTURE



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

    @Autowired
    PaymentService paymentService;

    @PostMapping("/add")
    public ResponseEntity<PaymentResponse> getPaymentById(@PathVariable Long id) {
        //Now passing data to internal request obj
        PaymentRequest internalRequest = new PaymentRequest();
        internalRequest.setId(id);
        internalRequest.setPaymentId(id);

        //Now this internalRequest is further logic for processing
        PaymentResponse payment = paymentService.getPaymentById(internalRequest);

        //Return the response obj
        return ResponseEntity.ok(payment);
    }
}
```

```
//Service
@Service
public class PaymentService {

    @Autowired
    PaymentRepository paymentRepository;

    public PaymentResponse getPaymentById(PaymentRequest internalRequest) {
        PaymentEntity paymentEntity = paymentRepository.getPaymentById(internalRequest.getId());
        return paymentEntity;
    }

    //Now if we request DB
    public PaymentResponse getPaymentById(PaymentRequest internalRequest) {
        PaymentEntity paymentEntity = paymentRepository.getPaymentById(internalRequest.getId());
        return paymentEntity;
    }

    //Now if we request DB
    public PaymentResponse getPaymentById(PaymentRequest internalRequest) {
        PaymentEntity paymentEntity = paymentRepository.getPaymentById(internalRequest.getId());
        return paymentEntity;
    }
}
```

```
//Repository
public class PaymentRepository {

    public PaymentEntity getPaymentById(PaymentRequest request) {
        PaymentEntity paymentEntity = entityManager(request);
        return paymentEntity;
    }

    private PaymentEntity entityManager(PaymentRequest request) {
        //Connect with DB and fetch the data
        PaymentEntity payment = new PaymentEntity();
        payment.setPaymentId(request.getId());
        payment.setPaymentId(request.getId());
        payment.setPaymentId(request.getId());
        return payment;
    }
}
```