

Spring boot: Project Setup and Layered Architecture (Concept && Codi...

"Concept && Coding" YT Video Notes

Setting up the Project

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

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

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

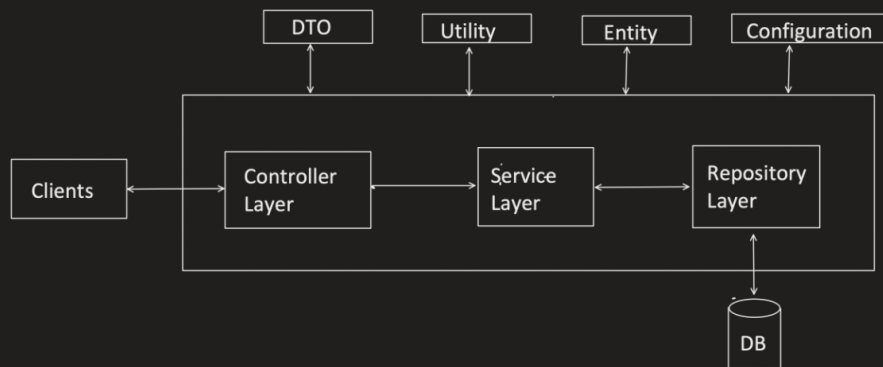
Java: ☐ 21 ☒ **17**

Dependencies ADD DEPENDENCIES... 14 + 8

Spring Web NEW
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("/{id}")
    public ResponseEntity<PaymentResponse> getPaymentById(@PathVariable Long id) {
        //Map incoming data to internal request DTO
        PaymentRequest internalRequest = new PaymentRequest();
        internalRequest.setId(id);

        //Pass this internalRequest to further layer for processing
        PaymentResponse payment = paymentService.getPaymentDetails(id(internalRequest));

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

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

    @Autowired
    PaymentRepository paymentRepository;

    public PaymentResponse getPaymentDetails(id(PaymentRequest internalRequest)) {
        PaymentEntity paymentModel = paymentRepository.getPayment(id(internalRequest));
        return paymentResponse;
    }

    //Map id to response and
    PaymentResponse paymentResponse = new PaymentResponse();
    paymentResponse.setId(id(internalRequest));

    //Map id to response and
    PaymentResponse response = new PaymentResponse();
    response.setId(paymentModel.getId());
    response.setAmount(paymentModel.getAmount());
    response.setCurrency(paymentModel.getCurrency());
    return response;
}
```

```
//Repository
@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.getId());
        payment.setAmount(request.getAmount());
        payment.setCurrency(request.getCurrency());
        return payment;
    }
}
```