

OverAll

OverAll

@Transactional (重点)

Create Database

Add @EnableTransactionManagement

Without @Transactional

With @Transactional

Correct Request

Request body -> fail

Request body -> work

@Transactional in JPA

Web Server(easy, 基本sense)

Running Springboot Apps from the Command Line

Deploy SpringBoot WAR file to External Tomcat Server

Logging(重点, easy)

Level

Example

Logging Architecture

Spring AOP(easy, 基本sense)

Usage

Without AOP

AOP Concepts

Examples

Debug to show how AOP Work

Spring Batch(基本sense)

Spring MVC(基本sense, 保证能回答问答题)

Tomcat

Run the application

@Transactional (重点)

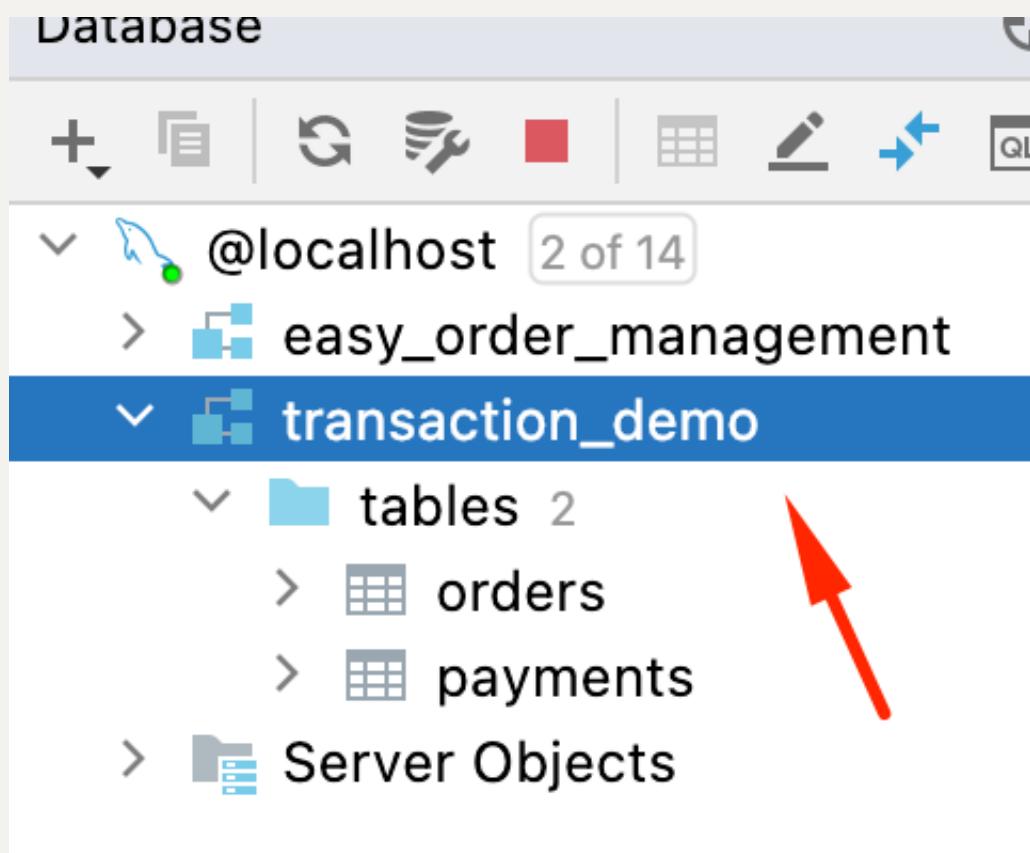
if the transaction failed, then it rollback.

Repo: <https://github.com/TAIsRich/tutorial-transaction.git>

Branch: main

Create Database

```
1 create database transaction_demo
```



Add @EnableTransactionManagement

```
import ...  
  
@SpringBootApplication  
@EnableTransactionManagement  
public class TutorialTransactionApplication {  
  
    public static void main(String[] args) { SpringApplication.run(TutorialTransactio  
}  

```

Without @Transactional

we only support DEBIT card, so if we provide other cards, we throw exception.

tutorial-transaction > src > main > java > com > chuwa > tutorialtransaction > service > impl > OrderServiceImpl.java > placeOrder

```
orders x OrderServiceImpl.java x  
27  
28     @Override  
29     //... @Transactional  
30     public OrderResponse placeOrder(OrderRequest orderRequest) {  
31  
32         Order order = orderRequest.getOrder();  
33         order.setStatus("INPROGRESS");  
34         order.setOrderTackingNumber(UUID.randomUUID().toString());  
35         orderRepository.save(order);  
36  
37         Payment payment = orderRequest.getPayment();  
38  
39         if(!payment.getType().equals("DEBIT")){  
40             throw new PaymentException("Payment card type do not support");  
41         }  
42  
43         payment.setOrderId(order.getId());  
44         paymentRepository.save(payment);  
45  
46         OrderResponse orderResponse = new OrderResponse();  
47         orderResponse.setOrderTackingNumber(order.getOrderTackingNumber());  
48         orderResponse.setStatus(order.getStatus());  
49         orderResponse.setMessage("SUCCESS");  
50         return orderResponse;  
51     }  
52 }
```

transaction doesn't created successfully.

The screenshot shows a Java IDE interface with two panes. The left pane displays a POST request in Postman to the endpoint `(host)/api/v1/orders`. The JSON body of the request is:

```
1 {
2     "order": {
3         "totalQuantity": 3,
4         "totalPrice": "2000",
5         "shoppingCartId": 2
6     },
7     "payment": {
8         "type": "Credit",
9         "cardName": "Chuwa America",
10        "cardNumber": "666 6666 6666 6666",
11        "expiryYear": 2027,
12        "expiryMonth": 7,
13        "cvc": 123
14    }
15 }
```

A red arrow points to the `type: "Credit"` line in the JSON body. The right pane shows two database tables: `orders` and `payments`. The `orders` table has one record with `last_updated` at `2022-07-05 00:37:29.070000` and `order_tacking_number` at `05c622cb-2324-4484-8bed-3cff983`, with a status message `order record created`. The `payments` table has no records, indicated by the message `payment record NOT created since not DEBIT card, throw exception`.

We need to rollback when this transaction is executed unsuccessfully.

both tables have records
or
both tables have no records

The bottom part of the screenshot shows the Java application's log in the console:

```
1 [2022-07-04 17:37:29.107] ERROR 83266 --- [nio-8080-exec-2] o.a.c.c.C.[.].dispatcherservlet : Servlet.service() for servlet [dispatcher] threw exception [Request processing failed; nested exception is com.chuwa.tutorialtransaction.exception.PaymentException: Payment card type do not support]
2 com.chuwa.tutorialtransaction.exception.PaymentException Create breakpoint : Payment card type do not support
3     at com.chuwa.tutorialtransaction.service.impl.OrderServiceImpl.placeOrder(OrderServiceImpl.java:40) ~[classes/:na]
4     at com.chuwa.tutorialtransaction.controller.OrderController.placeOrder(OrderController.java:24) ~[classes/:na] <14 internal lines>
5     at javax.servlet.http.HttpServlet.service(HttpServlet.java:681) ~[tomcat-embed-core-9.0.64.jar:4.0.FR] <1 internal line>
6     at javax.servlet.http.HttpServlet.service(HttpServlet.java:764) ~[tomcat-embed-core-9.0.64.jar:4.0.FR] <33 internal lines>
```

A red arrow points to the `PaymentException` class name in the stack trace.

With `@Transactional`

The screenshot shows a developer's workspace with several open windows:

- Postman:** A browser-based API testing tool showing a POST request to `(host)/api/v1/orders`. The request body is a JSON object with fields like `order`, `totalQuantity`, `totalPrice`, `shoppingCartId`, and `payment`. The `payment` field contains a card details object with `type: "Credit"` and a placeholder card number.
- Database:** Two database tables are shown: `orders` and `payments`. The `orders` table has a row with `last_updated: 2022-07-05 00:37:29.070000` and `order_tacking_number: 05c622cb-2324-4484-8bed-3c1`. The `payments` table also has a row with the same timestamp and tracking number.
- Terminal:** A Java application named `TutorialTransactionApplication` is running. The log output shows an error message: `2022-07-04 17:48:07.416 ERROR 85371 --- [nio-8080-exec-1] o.a.c.c.C.[.].[dispatcherServlet] : Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed; nested exception is com.chuwa.tutorialtransaction.exception.PaymentException: Payment card type do not support]`.

Annotations in red highlight specific parts of the code and logs, such as the old record in the database and the payment exception message.

Correct Request

transaction executed successfully

Request body -> fail

```

1  {
2      "order": {
3          "totalQuantity": 3,
4          "totalPrice": "2000",
5          "shoppingCartId": 2
6      },
7      "payment": {
8          "type": "CREDIT",
9          "cardName": "Chuwa America",
10         "cardNumber": "666 6666 6666 6666",
11         "expiryYear": 2027,
12         "expiryMonth": 7,
13         "cvc": 123
14     }
15 }
```

```

14 }
15 }
```

Request boyd -> work

```

1 {
2     "order": {
3         "totalQuantity": 3,
4         "totalPrice": "2000",
5         "shoppingCartId": 2
6     },
7     "payment": {
8         "type": "DEBIT",
9         "cardName": "Chuwa America",
10        "cardNumber": "666 6666 6666 6666",
11        "expiryYear": 2027,
12        "expiryMonth": 7,
13        "cvc": 123
14    }
15 }
```

Chuwa / transaction / New Request

POST `(Host)/api/v1/orders`

Params Auth Headers (10) Body **JSON** Pre-req Tests Settings

`1 {
2 "order": {
3 "totalQuantity": 3,
4 "totalPrice": "2000",
5 "shoppingCartId": 2
6 },
7 "payment": {
8 "type": "DEBIT",
9 "cardName": "Chuwa America",
10 "cardNumber": "666 6666 6666 6666",
11 "expiryYear": 2027,
12 "expiryMonth": 7,
13 "cvc": 123
14 }
15 }`

Body **Pretty** `200 OK` `31ms` `287 B` Save Response

last_updated	order_tacking	id	card_name	card_number	cvc	expiry_month
1.069800	2022-07-05 08:37:29.070000	05622cb-2324-441	2 Chuwa America	666 6666 6666 6666	123	7
2.544800	2022-07-05 08:51:26.544000	779179c8-9c8a-43c9-9c51-b107a0aea2ac				

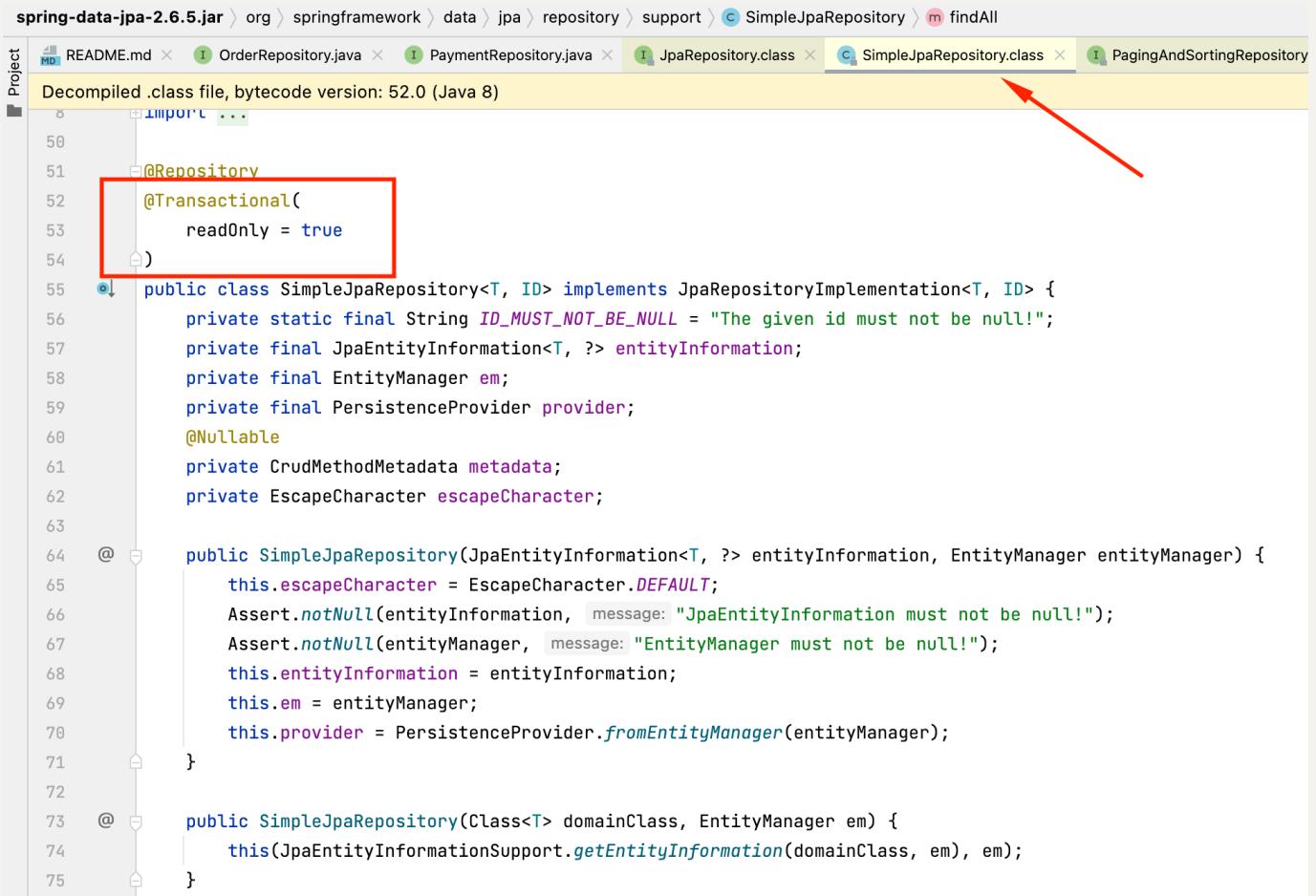
Run: TutorialTransactionApplication

Console Actuator

```
into
orders
(date_created, last_updated, order_tacking_number, shopping_cart_id, status, total_price, total_quantity)
values
```

@Transactional in JPA

JPA has also used @Transactional



The screenshot shows a decompiled .class file for the SimpleJpaRepository class from the spring-data-jpa-2.6.5.jar library. The code is annotated with various Java annotations like @Repository, @Transactional, and @NotNullable. A red box highlights the @Transactional annotation on line 52, which is part of a constructor parameter. A red arrow points from the top right towards this highlighted code. The IDE interface includes tabs for README.md, OrderRepository.java, PaymentRepository.java, JpaRepository.class, SimpleJpaRepository.class (which is the active tab), and PagingAndSortingRepository.

```
spring-data-jpa-2.6.5.jar > org > springframework > data > jpa > repository > support > C SimpleJpaRepository > m findAll
Project README.md × OrderRepository.java × PaymentRepository.java × JpaRepository.class × C SimpleJpaRepository.class × I PagingAndSortingRepository
Decompiled .class file, bytecode version: 52.0 (Java 8)
...
50
51     @Repository
52     @Transactional(
53         readOnly = true
54     )
55     public class SimpleJpaRepository<T, ID> implements JpaRepositoryImplementation<T, ID> {
56         private static final String ID_MUST_NOT_BE_NULL = "The given id must not be null!";
57         private final JpaEntityInformation<T, ?> entityInformation;
58         private final EntityManager em;
59         private final PersistenceProvider provider;
60         @Nullable
61         private CrudMethodMetadata metadata;
62         private EscapeCharacter escapeCharacter;
63
64         @
65         public SimpleJpaRepository(JpaEntityInformation<T, ?> entityInformation, EntityManager entityManager) {
66             this.escapeCharacter = EscapeCharacter.DEFAULT;
67             Assert.notNull(entityInformation, message: "JpaEntityInformation must not be null!");
68             Assert.notNull(entityManager, message: "EntityManager must not be null!");
69             this.entityInformation = entityInformation;
70             this.em = entityManager;
71             this.provider = PersistenceProvider.fromEntityManager(entityManager);
72         }
73
74         @
75         public SimpleJpaRepository(Class<T> domainClass, EntityManager em) {
76             this(JpaEntityInformationSupport.getEntityInformation(domainClass, em), em);
77         }
}
```

Web Server(easy, 基本sense)

Repo: <https://github.com/TAIsRich/tutorial-transaction.git>

Branch: WebServer



The terminal window shows the command 'git:(WebServer)' indicating the current git branch. The window title is 'Terminal: Local'.

```
Terminal: Local × + ▾
→ tutorial-transaction git:(WebServer) ×
```

Tomcat, JBoss, Jetty

1. Download and install Tomcat server
2. Package Java application as WAR File
3. Deploy WAR file in tomcat server
4. Start Tomcat server

```

pom.xml (tutorial-transaction) × spring-boot-starter-web-2.6.9.pom ×
16 <properties>
17   <java.version>1.8</java.version>
18 </properties>
19 <dependencies>
20   <dependency>
21     <groupId>org.springframework.boot</groupId>
22     <artifactId>spring-boot-starter-data-jpa</artifactId>
23   </dependency>
24
25   <dependency>
26     <groupId>org.springframework.boot</groupId>
27     <artifactId>spring-boot-starter-web</artifactId>
28   </dependency>
29
30   <dependency>
31     <groupId>mysql</groupId>
32     <artifactId>mysql-connector-java</artifactId>
33     <scope>runtime</scope>
34   </dependency>
35
36   <dependency>
37     <groupId>org.projectlombok</groupId>
38     <artifactId>lombok</artifactId>
39     <optional>true</optional>
40   </dependency>
41
42   <dependency>
43     <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-test</artifactId>
     <scope>test</scope>

```

```

1 <dependency>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-starter-web</artifactId>
4
5   <exclusions>
6     <exclusion>
7       <groupId>org.springframework.boot</groupId>
8       <artifactId>spring-boot-starter-tomcat</artifactId>
9     </exclusion>
10
11   </exclusions>
12
13 </dependency>
14 <dependency>
15   <groupId>org.springframework.boot</groupId>
16   <artifactId>spring-boot-starter-jetty</artifactId>
17 </dependency>

```

```

1 <dependency>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-starter-web</artifactId>
4
5   <exclusions>
6     <exclusion>
7       <groupId>org.springframework.boot</groupId>
8       <artifactId>spring-boot-starter-tomcat</artifactId>
9     </exclusion>
10
11   </exclusions>
12
13 </dependency>
14 <dependency>
15   <groupId>org.springframework.boot</groupId>
16   <artifactId>spring-boot-starter-jetty</artifactId>
17 </dependency>

```

pom.xml (tutorial-transaction) ×

```

20 <dependency>
21   <groupId>org.springframework.boot</groupId>
22   <artifactId>spring-boot-starter-data-jpa</artifactId>
23 </dependency>
24
25 <dependency>
26   <groupId>org.springframework.boot</groupId>
27   <artifactId>spring-boot-starter-web</artifactId>
28   <exclusions>
29     <exclusion>
30       <groupId>org.springframework.boot</groupId>
31       <artifactId>spring-boot-starter-tomcat</artifactId>
32     </exclusion>
33   </exclusions>
34 </dependency>
35 <dependency>
36   <groupId>org.springframework.boot</groupId>
37   <artifactId>spring-boot-starter-jetty</artifactId>
38 </dependency>
39

```

project > dependencies > dependency > optional

run with Jetty server instead of tomcat

```

main] org.eclipse.jetty.server.session      : node0 Scavenging every 60000ms
main] o.e.jetty.server.handler.ContextHandler : Started o.s.b.w.e.j.JettyEmbeddedWebApplicationContext@6daf2337{application,/, /,AVAILABLE}
main] org.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
main] org.hibernate.Version                 : HHH000412: Hibernate ORM core version 5.6.9.Final
main] o.hibernat.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
main] com.zaxxer.hikari.HikariDataSource   : HikariPool-1 - Starting...
main] com.zaxxer.hikari.HikariDataSource   : HikariPool-1 - Start completed.
main] org.hibernate.dialect.Dialect        : HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
main] o.h.e.t.j.p.i.JtaPlatformInitiator   : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformImpl]
main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed even if they are not explicitly requested.
main] o.e.j.s.h.ContextHandler.application   : Initializing Spring DispatcherServlet 'dispatcherServlet'
main] o.s.web.servlet.DispatcherServlet      : Initializing Servlet 'dispatcherServlet'
main] o.s.web.servlet.DispatcherServlet      : Completed initialization in 0 ms
main] o.e.jetty.server.AbstractConnector     : Started ServerConnector@779de014{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
main] o.s.b.web.embedded.jetty.JettyWebServer : Jetty started on port(s) 8080 (http/1.1) with context path '/'
main] c.c.t.TutorialTransactionApplication  : Started TutorialTransactionApplication in 4.051 seconds (JVM running for 4.712)

```

Running Springboot Apps from the Command Line

1. java -jar

```

1 $> ./mvnw package
2 $> cd target
3 $> java -jar tutorial-transaction-0.0.1-SNAPSHOT.jar

```

2. mvn spring-boot:run

```
1 $> ./mvnw spring-boot:run
```

```
1 → tutorial-transaction git:(WebServer) ✘ ./mvnw package

2 [INFO] Scanning for projects...
3 [INFO]
4 [INFO] -----< com.chuwa:tutorial-transaction >-----
5 [INFO] Building tutorial-transaction 0.0.1-SNAPSHOT
6 [INFO] -----[ jar ]-----
7 [INFO]
8 [INFO] --- maven-resources-plugin:3.2.0:resources (default-
resources) @ tutorial-transaction ---
9 [INFO] Using 'UTF-8' encoding to copy filtered resources.
10 [INFO] Using 'UTF-8' encoding to copy filtered properties files.
11 [INFO] Copying 1 resource
12 [INFO] Copying 0 resource
13 [INFO]
14 [INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile)
@ tutorial-transaction ---
15 [INFO] Changes detected - recompiling the module!
16 [INFO] Compiling 11 source files to
/Users/bigo/Documents/githubProjects/chuwa/tutorial-
transaction/target/classes
17 [INFO]
18 [INFO] --- maven-resources-plugin:3.2.0:testResources (default-
testResources) @ tutorial-transaction ---
19 [INFO] Using 'UTF-8' encoding to copy filtered resources.
20 [INFO] Using 'UTF-8' encoding to copy filtered properties files.
21 [INFO] skip non existing resourceDirectory
/Users/bigo/Documents/githubProjects/chuwa/tutorial-
transaction/src/test/resources
22 [INFO]
```

```
23 [INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-
testCompile) @ tutorial-transaction ---
24 [INFO] Changes detected - recompiling the module!
25 [INFO] Compiling 1 source file to
/Users/bigo/Documents/githubProjects/chuwa/tutorial-
transaction/target/test-classes
26 [INFO]
27 [INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @
tutorial-transaction ---
28 [INFO]
29 [INFO] -----
30 [INFO] T E S T S
31 [INFO] -----
32 [INFO] Running
com.chuwa.tutorialtransaction.TutorialTransactionApplicationTests
33 21:47:16.277 [main] DEBUG
org.springframework.test.context.BootstrapUtils - Instantiating
CacheAwareContextLoaderDelegate from class
[org.springframework.test.context.cache.DefaultCacheAwareContextL
oaderDelegate]
34 21:47:16.290 [main] DEBUG
org.springframework.test.context.BootstrapUtils - Instantiating
BootstrapContext using constructor [public
org.springframework.test.context.support.DefaultBootstrapContext(
java.lang.Class,org.springframework.test.context.CacheAwareContex
tLoaderDelegate)]
35 21:47:16.330 [main] DEBUG
org.springframework.test.context.BootstrapUtils - Instantiating
TestContextBootstrapper for test class
[com.chuwa.tutorialtransaction.TutorialTransactionApplicationTest
s] from class
[org.springframework.boot.test.context.SpringBootTestBoots
trapper]
```

```
36 21:47:16.347 [main] INFO  
org.springframework.boot.test.context.SpringBootTestBootstrapper - Neither @ContextConfiguration nor @ContextHierarchy found for test class  
[com.chuwa.tutorialtransaction.TutorialTransactionApplicationTests], using SpringBootTestContextLoader  
37 21:47:16.352 [main] DEBUG  
org.springframework.test.context.support.AbstractContextLoader -  
Did not detect default resource location for test class  
[com.chuwa.tutorialtransaction.TutorialTransactionApplicationTests]: class path resource  
[com/chuwa/tutorialtransaction/TutorialTransactionApplicationTests-context.xml] does not exist  
38 21:47:16.353 [main] DEBUG  
org.springframework.test.context.support.AbstractContextLoader -  
Did not detect default resource location for test class  
[com.chuwa.tutorialtransaction.TutorialTransactionApplicationTests]: class path resource  
[com/chuwa/tutorialtransaction/TutorialTransactionApplicationTestsContext.groovy] does not exist  
39 21:47:16.353 [main] INFO  
org.springframework.test.context.support.AbstractContextLoader -  
Could not detect default resource locations for test class  
[com.chuwa.tutorialtransaction.TutorialTransactionApplicationTests]: no resource found for suffixes {-context.xml,  
Context.groovy}.  
40 21:47:16.354 [main] INFO  
org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuration classes for test class  
[com.chuwa.tutorialtransaction.TutorialTransactionApplicationTests]: TutorialTransactionApplicationTests does not declare any static, non-private, non-final, nested classes annotated with @Configuration.
```

```
41 21:47:16.425 [main] DEBUG  
org.springframework.test.context.support.ActiveProfilesUtils -  
Could not find an 'annotation declaring class' for annotation  
type [org.springframework.test.context.ActiveProfiles] and class  
[com.chuwa.tutorialtransaction.TutorialTransactionApplicationTest  
s]  
42 21:47:16.494 [main] DEBUG  
org.springframework.context.annotation.ClassPathScanningCandidate  
ComponentProvider - Identified candidate component class: file  
[/Users/bigo/Documents/githubProjects/chuwa/tutorial-  
transaction/target/classes/com/chuwa/tutorialtransaction/Tutorial  
TransactionApplication.class]  
43 21:47:16.495 [main] INFO  
org.springframework.boot.test.context.SpringBootTestContextBootst  
rapper - Found @SpringBootConfiguration  
com.chuwa.tutorialtransaction.TutorialTransactionApplication for  
test class  
com.chuwa.tutorialtransaction.TutorialTransactionApplicationTests  
44 21:47:16.601 [main] DEBUG  
org.springframework.boot.test.context.SpringBootTestContextBootst  
rapper - @TestExecutionListeners is not present for class  
[com.chuwa.tutorialtransaction.TutorialTransactionApplicationTest  
s]: using defaults.
```

```
45 21:47:16.601 [main] INFO  
org.springframework.boot.test.context.SpringBootTestBootstrapper - Loaded default TestExecutionListener class names from location [META-INF/spring.factories]:  
[org.springframework.boot.test.mock.mockito.MockitoTestExecutionListener,  
org.springframework.boot.test.mock.mockito.ResetMocksTestExecutionListener,  
org.springframework.boot.test.autoconfigure.restdocs.RestDocsTestExecutionListener,  
org.springframework.boot.test.autoconfigure.web.client.MockRestServiceServerResetTestExecutionListener,  
org.springframework.boot.test.autoconfigure.web.servlet.MockMvcPrintOnlyOnFailureTestExecutionListener,  
org.springframework.boot.test.autoconfigure.web.servlet.WebDriverTestExecutionListener,  
org.springframework.boot.test.autoconfigure.webservices.client.MockWebServiceServerTestExecutionListener,  
org.springframework.test.context.web.ServletTestExecutionListener,  
,  
org.springframework.test.context.support.DirtiesContextBeforeModesTestExecutionListener,  
org.springframework.test.context.event.ApplicationEventsTestExecutionListener,  
org.springframework.test.context.support.DependencyInjectionTestExecutionListener,  
org.springframework.test.context.support.DirtiesContextTestExecutionListener,  
org.springframework.test.context.transaction.TransactionalTestExecutionListener,  
org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener,  
org.springframework.test.context.event.EventPublishingTestExecutionListener]
```

```
46 21:47:16.622 [main] INFO  
org.springframework.boot.test.context.SpringBootTestBootstrapper - Using TestExecutionListeners:  
[org.springframework.test.context.web.ServletTestExecutionListener@20c0a64d,  
org.springframework.test.context.support.DirtiesContextBeforeModeTestExecutionListener@455b6df1,  
org.springframework.test.context.event.ApplicationEventsTestExecutionListener@4ddb9df8,  
org.springframework.boot.test.mock.mockito.MockitoTestExecutionListener@3f67593e,  
org.springframework.boot.test.autoconfigure.SpringBootDependencyInjectionTestExecutionListener@1ab06251,  
org.springframework.test.context.support.DirtiesContextTestExecutionListener@41ab013,  
org.springframework.test.context.transaction.TransactionalTestExecutionListener@14bee915,  
org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener@1115ec15,  
org.springframework.test.context.event.EventPublishingTestExecutionListener@82ea68c,  
org.springframework.boot.test.mock.mockito.ResetMocksTestExecutionListener@59e505b2,  
org.springframework.boot.test.autoconfigure.restdocs.RestDocsTestExecutionListener@3af0a9da,  
org.springframework.boot.test.autoconfigure.web.client.MockRestServiceServerResetTestExecutionListener@43b9fd5,  
org.springframework.boot.test.autoconfigure.web.servlet.MockMvcPrintOnlyOnFailureTestExecutionListener@79dc5318,  
org.springframework.boot.test.autoconfigure.web.servlet.WebDriverTestExecutionListener@8e50104,  
org.springframework.boot.test.autoconfigure.webservices.client.MockWebServiceServerTestExecutionListener@37e4d7bb]  
47 21:47:16.626 [main] DEBUG  
org.springframework.test.context.support.AbstractDirtiesContextTestExecutionListener - Before test class: context
```

```
[DefaultTestContext@600b9d27 testClass =
TutorialTransactionApplicationTests, testInstance = [null],
testMethod = [null], testException = [null],
mergedContextConfiguration =
[WebMergedContextConfiguration@3688eb5b testClass =
TutorialTransactionApplicationTests, locations = '{}', classes =
'{class
com.chuwa.tutorialtransaction.TutorialTransactionApplication}',
contextInitializerClasses = '[]', activeProfiles = '{}',
propertySourceLocations = '{}', propertySourceProperties =
'{org.springframework.boot.test.context.SpringBootTestBoot
strapper=true}', contextCustomizers =
set[org.springframework.boot.test.context.filter.ExcludeFilterCon
textCustomizer@67080771,
org.springframework.boot.test.json.DuplicateJsonObjectContextCust
omizerFactory$DuplicateJsonObjectContextCustomizer@696da30b,
org.springframework.boot.test.mock.mockito.MockitoContextCustomiz
er@0,
org.springframework.boot.test.web.client.TestRestTemplateContextC
ustomizer@693fe6c9,
org.springframework.boot.test.autoconfigure.actuate.metrics.Metri
csExportContextCustomizerFactory$DisableMetricExportContextCustom
izer@10163d6,
org.springframework.boot.test.autoconfigure.properties.PropertyMa
ppingContextCustomizer@0,
org.springframework.boot.test.autoconfigure.web.servlet.WebDriver
ContextCustomizerFactory$Customizer@3fb1549b,
org.springframework.boot.test.context.SpringBootTestArgs@1,
org.springframework.boot.test.context.SpringBootTestWebEnvironmen
t@48503868], resourcebasePath = 'src/main/webapp', contextLoader
=
'org.springframework.boot.test.context.SpringBootTestLoader',
parent = [null]], attributes =
map['org.springframework.test.context.web.ServletTestExecutionLis
tener.activateListener' -> true]], class annotated with
@DirtiesContext [false] with mode [null].
```

```
48
49
50
51
52
53
54
55
56
57 2022-07-04 21:47:16.992  INFO 8646 --- [           main]
      .c.t.TutorialTransactionApplicationTests : Starting
      TutorialTransactionApplicationTests using Java 15.0.1 on Yanans-
      MBP with PID 8646 (started by bigo in
      /Users/bigo/Documents/githubProjects/chuwa/tutorial-transaction)
58 2022-07-04 21:47:16.993  INFO 8646 --- [           main]
      .c.t.TutorialTransactionApplicationTests : No active profile set,
      falling back to 1 default profile: "default"
59 2022-07-04 21:47:17.793  INFO 8646 --- [           main]
      .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring
      Data JPA repositories in DEFAULT mode.
60 2022-07-04 21:47:17.844  INFO 8646 --- [           main]
      .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data
      repository scanning in 41 ms. Found 2 JPA repository interfaces.
61 2022-07-04 21:47:18.385  INFO 8646 --- [           main]
      o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing
      PersistenceUnitInfo [name: default]
62 2022-07-04 21:47:18.470  INFO 8646 --- [           main]
      org.hibernate.Version                  : HHH000412: Hibernate
      ORM core version 5.6.9.Final
63 2022-07-04 21:47:18.698  INFO 8646 --- [           main]
      o.hibernate.annotations.common.Version  : HCANN000001: Hibernate
      Commons Annotations {5.1.2.Final}
64 2022-07-04 21:47:18.841  INFO 8646 --- [           main]
      com.zaxxer.hikari.HikariDataSource   : HikariPool-1 -
      Starting...
```

```
65 2022-07-04 21:47:19.088 INFO 8646 --- [           main]
    com.zaxxer.hikari.HikariDataSource      : HikariPool-1 - Start
    completed.

66 2022-07-04 21:47:19.110 INFO 8646 --- [           main]
    org.hibernate.dialect.Dialect          : HHH000400: Using
    dialect: org.hibernate.dialect.MySQL8Dialect

67 2022-07-04 21:47:19.790 INFO 8646 --- [           main]
    o.h.e.t.j.p.i.JtaPlatformInitiator     : HHH000490: Using
    JtaPlatform implementation:
    [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlat
    form]

68 2022-07-04 21:47:19.801 INFO 8646 --- [           main]
    j.LocalContainerEntityManagerFactoryBean : Initialized JPA
    EntityManagerFactory for persistence unit 'default'

69 2022-07-04 21:47:20.162 INFO 8646 --- [           main]
    org.eclipse.jetty.util.log              : Logging initialized
    @4518ms to org.eclipse.jetty.util.log.Slf4jLog

70 2022-07-04 21:47:20.256 WARN 8646 --- [           main]
    JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-
    view is enabled by default. Therefore, database queries may be
    performed during view rendering. Explicitly configure
    spring.jpa.open-in-view to disable this warning

71 2022-07-04 21:47:20.742 INFO 8646 --- [           main]
    .c.t.TutorialTransactionApplicationTests : Started
    TutorialTransactionApplicationTests in 4.057 seconds (JVM running
    for 5.098)

72 [INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time
    elapsed: 4.752 s - in
    com.chuwa.tutorialtransaction.TutorialTransactionApplicationTests

73 2022-07-04 21:47:20.954 INFO 8646 --- [ionShutdownHook]
    j.LocalContainerEntityManagerFactoryBean : Closing JPA
    EntityManagerFactory for persistence unit 'default'

74 2022-07-04 21:47:20.956 INFO 8646 --- [ionShutdownHook]
    com.zaxxer.hikari.HikariDataSource      : HikariPool-1 -
    Shutdown initiated...
```

```
75 2022-07-04 21:47:20.968 INFO 8646 --- [ionShutdownHook]
    com.zaxxer.hikari.HikariDataSource      : HikariPool-1 -
    Shutdown completed.

76 [INFO]
77 [INFO] Results:
78 [INFO]
79 [INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
80 [INFO]
81 [INFO]
82 [INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ tutorial-
transaction ---
83 [INFO] Building jar:
    /Users/bigo/Documents/githubProjects/chuwa/tutorial-
transaction/target/tutorial-transaction-0.0.1-SNAPSHOT.jar
84 [INFO]
85 [INFO] --- spring-boot-maven-plugin:2.6.9:repackage (repackage) @
tutorial-transaction ---
86 [INFO] Replacing main artifact with repackaged archive
87 [INFO] -----
-----
88 [INFO] BUILD SUCCESS
89 [INFO] -----
-----
90 [INFO] Total time: 10.461 s
91 [INFO] Finished at: 2022-07-04T21:47:22-07:00
92 [INFO] -----
-----
93 → tutorial-transaction git:(WebServer) ✘ cd target
94 → target git:(WebServer) ✘ ls
95 classes                                maven-archiver
                                         test-classes
96 generated-sources                      maven-status
                                         tutorial-transaction-0.0.1-
                                         SNAPSHOT.jar
```

```
97 generated-test-sources                               surefire-reports
                                                     tutorial-transaction-0.0.1-
                                                     SNAPSHOT.jar.original
98 → target git:(WebServer) ✘
99 → target git:(WebServer) ✘ java -jar tutorial-transaction-0.0.1-
SNAPSHOT.jar
100
101   .
102   / \ \ / \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
103   ( ( ) \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
104   \ \ \ / \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
105   ' \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
106   ======|_|=====|_|=/_/_/_/
107 :: Spring Boot ::          (v2.6.9)
108
109 2022-07-04 21:50:14.721  INFO 9342 --- [           main]
  c.c.t.TutorialTransactionApplication      : Starting
  TutorialTransactionApplication v0.0.1-SNAPSHOT using Java
  1.8.0_271 on Yanans-MBP with PID 9342
  (/Users/bigo/Documents/githubProjects/chuwa/tutorial-
  transaction/target/tutorial-transaction-0.0.1-SNAPSHOT.jar
  started by bigo in
  /Users/bigo/Documents/githubProjects/chuwa/tutorial-
  transaction/target)
110 2022-07-04 21:50:14.724  INFO 9342 --- [           main]
  c.c.t.TutorialTransactionApplication      : No active profile set,
  falling back to 1 default profile: "default"
111 2022-07-04 21:50:15.723  INFO 9342 --- [           main]
  .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring
  Data JPA repositories in DEFAULT mode.
112 2022-07-04 21:50:15.806  INFO 9342 --- [           main]
  .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data
  repository scanning in 63 ms. Found 2 JPA repository interfaces.
113 2022-07-04 21:50:16.319  INFO 9342 --- [           main]
  org.eclipse.jetty.util.log                : Logging initialized
  @2849ms to org.eclipse.jetty.util.log.Slf4jLog
```

```
114 2022-07-04 21:50:16.547 INFO 9342 --- [           main]
     o.s.b.w.e.j.JettyServletWebServerFactory : Server initialized
     with port: 8080
115 2022-07-04 21:50:16.551 INFO 9342 --- [           main]
     org.eclipse.jetty.server.Server          : jetty-
     9.4.48.v20220622; built: 2022-06-21T20:42:25.880Z; git:
     6b67c5719d1f4371b33655ff2d047d24e171e49a; jvm 1.8.0_271-b09
116 2022-07-04 21:50:16.601 INFO 9342 --- [           main]
     o.e.j.s.h.ContextHandler.application    : Initializing Spring
     embedded WebApplicationContext
117 2022-07-04 21:50:16.602 INFO 9342 --- [           main]
     w.s.c.ServletWebServerApplicationContext : Root
     WebApplicationContext: initialization completed in 1794 ms
118 2022-07-04 21:50:16.776 INFO 9342 --- [           main]
     org.eclipse.jetty.server.session         :
     DefaultSessionIdManager workerName=node0
119 2022-07-04 21:50:16.776 INFO 9342 --- [           main]
     org.eclipse.jetty.server.session         : No SessionScavenger
     set, using defaults
120 2022-07-04 21:50:16.779 INFO 9342 --- [           main]
     org.eclipse.jetty.server.session         : node0 Scavenging every
     660000ms
121 2022-07-04 21:50:16.793 INFO 9342 --- [           main]
     o.e.jetty.server.handler.ContextHandler : Started
     o.s.b.w.e.j.JettyEmbeddedWebAppContext@6aeb35e6{application,,,
     [file:///private/var/folders/1/_7pf__s890tz1vfb8cfktr55h0000gn/T/
     jetty-docbase.8080.1479607307061441142/],AVAILABLE}
122 2022-07-04 21:50:16.793 INFO 9342 --- [           main]
     org.eclipse.jetty.server.Server          : Started @3324ms
123 2022-07-04 21:50:17.034 INFO 9342 --- [           main]
     o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing
     PersistenceUnitInfo [name: default]
124 2022-07-04 21:50:17.138 INFO 9342 --- [           main]
     org.hibernate.Version                  : HHH000412: Hibernate
     ORM core version 5.6.9.Final
```

```
125 2022-07-04 21:50:17.494 INFO 9342 --- [           main]
    o.hibernate.annotations.common.Version      : HCANN000001: Hibernate
    Commons Annotations {5.1.2.Final}
126 2022-07-04 21:50:17.662 INFO 9342 --- [           main]
    com.zaxxer.hikari.HikariDataSource       : HikariPool-1 -
    Starting...
127 2022-07-04 21:50:17.944 INFO 9342 --- [           main]
    com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start
    completed.
128 2022-07-04 21:50:17.971 INFO 9342 --- [           main]
    org.hibernate.dialect.Dialect          : HHH000400: Using
    dialect: org.hibernate.dialect.MySQL8Dialect
129 2022-07-04 21:50:18.928 INFO 9342 --- [           main]
    o.h.e.t.j.p.i.JtaPlatformInitiator     : HHH000490: Using
    JtaPlatform implementation:
    [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlat
    form]
130 2022-07-04 21:50:18.939 INFO 9342 --- [           main]
    j.LocalContainerEntityManagerFactoryBean : Initialized JPA
    EntityManagerFactory for persistence unit 'default'
131 2022-07-04 21:50:19.475 WARN 9342 --- [           main]
    JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-
    view is enabled by default. Therefore, database queries may be
    performed during view rendering. Explicitly configure
    spring.jpa.open-in-view to disable this warning
132 2022-07-04 21:50:19.899 INFO 9342 --- [           main]
    o.e.j.s.h.ContextHandler.application    : Initializing Spring
    DispatcherServlet 'dispatcherServlet'
133 2022-07-04 21:50:19.899 INFO 9342 --- [           main]
    o.s.web.servlet.DispatcherServlet      : Initializing Servlet
    'dispatcherServlet'
134 2022-07-04 21:50:19.900 INFO 9342 --- [           main]
    o.s.web.servlet.DispatcherServlet      : Completed
    initialization in 1 ms
```

```
135 2022-07-04 21:50:19.925 INFO 9342 --- [           main]
     o.e.jetty.server.AbstractConnector       : Started
     ServerConnector@28eaa59a{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
136 2022-07-04 21:50:19.926 INFO 9342 --- [           main]
     o.s.b.web.embedded.jetty.JettyWebServer : Jetty started on
     port(s) 8080 (http/1.1) with context path '/'
137 2022-07-04 21:50:19.936 INFO 9342 --- [           main]
     c.c.t.TutorialTransactionApplication   : Started
     TutorialTransactionApplication in 5.78 seconds (JVM running for
     6.467)
138
```

Use maven plugin

```
1 tutorial-transaction git:(WebServer) ✘ ./mvnw spring-boot:run
2 [INFO] Scanning for projects...
3 [INFO]
4 [INFO] -----< com.chuwa:tutorial-transaction >-----
5 [INFO] Building tutorial-transaction 0.0.1-SNAPSHOT
6 [INFO] -----[ jar ]-----
7 [INFO]
8 [INFO] >>> spring-boot-maven-plugin:2.6.9:run (default-cli) >
     test-compile @ tutorial-transaction >>>
9 [INFO]
10 [INFO] --- maven-resources-plugin:3.2.0:resources (default-
     resources) @ tutorial-transaction ---
11 [INFO] Using 'UTF-8' encoding to copy filtered resources.
12 [INFO] Using 'UTF-8' encoding to copy filtered properties files.
13 [INFO] Copying 1 resource
14 [INFO] Copying 0 resource
15 [INFO]
16 [INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @
     tutorial-transaction ---
```

```
17 [INFO] Nothing to compile - all classes are up to date
18 [INFO]
19 [INFO] --- maven-resources-plugin:3.2.0:testResources (default-
testResources) @ tutorial-transaction ---
20 [INFO] Using 'UTF-8' encoding to copy filtered resources.
21 [INFO] Using 'UTF-8' encoding to copy filtered properties files.
22 [INFO] skip non existing resourceDirectory
   /Users/bigo/Documents/githubProjects/chuwa/tutorial-
transaction/src/test/resources
23 [INFO]
24 [INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-
testCompile) @ tutorial-transaction ---
25 [INFO] Changes detected - recompiling the module!
26 [INFO] Compiling 1 source file to
   /Users/bigo/Documents/githubProjects/chuwa/tutorial-
transaction/target/test-classes
27 [INFO]
28 [INFO] <<< spring-boot-maven-plugin:2.6.9:run (default-cli) <
test-compile @ tutorial-transaction <<<
29 [INFO]
30 [INFO]
31 [INFO] --- spring-boot-maven-plugin:2.6.9:run (default-cli) @
tutorial-transaction ---
32 [INFO] Attaching agents: []
33
34 .
35 / \ \ / \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
36 ( ( ) \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
37 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
38 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
39 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
40 :: Spring Boot ::                               (v2.6.9)
41
```

```
42 2022-07-04 21:52:35.644 INFO 9912 --- [           main]
    c.c.t.TutorialTransactionApplication : Starting
    TutorialTransactionApplication using Java 15.0.1 on Yanans-MBP
    with PID 9912
    (/Users/bigo/Documents/githubProjects/chuwa/tutorial-
    transaction/target/classes started by bigo in
    /Users/bigo/Documents/githubProjects/chuwa/tutorial-transaction)
43 2022-07-04 21:52:35.647 INFO 9912 --- [           main]
    c.c.t.TutorialTransactionApplication : No active profile set,
    falling back to 1 default profile: "default"
44 2022-07-04 21:52:36.430 INFO 9912 --- [           main]
    .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring
    Data JPA repositories in DEFAULT mode.
45 2022-07-04 21:52:36.484 INFO 9912 --- [           main]
    .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data
    repository scanning in 42 ms. Found 2 JPA repository interfaces.
46 2022-07-04 21:52:36.890 INFO 9912 --- [           main]
    org.eclipse.jetty.util.log          : Logging initialized
    @2160ms to org.eclipse.jetty.util.log.Slf4jLog
47 2022-07-04 21:52:37.039 INFO 9912 --- [           main]
    o.s.b.w.e.j.JettyServletWebServerFactory : Server initialized with
    port: 8080
48 2022-07-04 21:52:37.042 INFO 9912 --- [           main]
    org.eclipse.jetty.server.Server      : jetty-9.4.48.v20220622;
    built: 2022-06-21T20:42:25.880Z; git:
    6b67c5719d1f4371b33655ff2d047d24e171e49a; jvm 15.0.1+9-18
49 2022-07-04 21:52:37.072 INFO 9912 --- [           main]
    o.e.j.s.h.ContextHandler.application : Initializing Spring
    embedded WebApplicationContext
50 2022-07-04 21:52:37.072 INFO 9912 --- [           main]
    w.s.c.ServletWebServerApplicationContext : Root
    WebApplicationContext: initialization completed in 1347 ms
51 2022-07-04 21:52:37.157 INFO 9912 --- [           main]
    org.eclipse.jetty.server.session     : DefaultSessionIdManager
    workerName=node0
```

```
52 2022-07-04 21:52:37.157 INFO 9912 --- [           main]
    org.eclipse.jetty.server.session          : No SessionScavenger
    set, using defaults
53 2022-07-04 21:52:37.159 INFO 9912 --- [           main]
    org.eclipse.jetty.server.session          : node0 Scavenging every
    660000ms
54 2022-07-04 21:52:37.168 INFO 9912 --- [           main]
    o.e.jetty.server.handler.ContextHandler : Started
    o.s.b.w.e.j.JettyEmbeddedWebAppContext@59072e9d{application,/,
    [file:///private/var/folders/1/_7pf__s890tz1vfb8cfktr55h0000gn/T/j
    etty-docbase.8080.17719546151893811632/],AVAILABLE}
55 2022-07-04 21:52:37.168 INFO 9912 --- [           main]
    org.eclipse.jetty.server.Server          : Started @2439ms
56 2022-07-04 21:52:37.373 INFO 9912 --- [           main]
    o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing
    PersistenceUnitInfo [name: default]
57 2022-07-04 21:52:37.440 INFO 9912 --- [           main]
    org.hibernate.Version                 : HHH000412: Hibernate
    ORM core version 5.6.9.Final
58 2022-07-04 21:52:37.725 INFO 9912 --- [           main]
    o.hibernate.annotations.common.Version : HCANN000001: Hibernate
    Commons Annotations {5.1.2.Final}
59 2022-07-04 21:52:37.868 INFO 9912 --- [           main]
    com.zaxxer.hikari.HikariDataSource   : HikariPool-1 -
    Starting...
60 2022-07-04 21:52:38.221 INFO 9912 --- [           main]
    com.zaxxer.hikari.HikariDataSource   : HikariPool-1 - Start
    completed.
61 2022-07-04 21:52:38.246 INFO 9912 --- [           main]
    org.hibernate.dialect.Dialect       : HHH000400: Using
    dialect: org.hibernate.dialect.MySQL8Dialect
62 2022-07-04 21:52:38.994 INFO 9912 --- [           main]
    o.h.e.t.j.p.i.JtaPlatformInitiator  : HHH000490: Using
    JtaPlatform implementation:
    [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatf
    orm]
```

```
63 2022-07-04 21:52:39.003 INFO 9912 --- [           main]
    j.LocalContainerEntityManagerFactoryBean : Initialized JPA
    EntityManagerFactory for persistence unit 'default'
64 2022-07-04 21:52:39.322 WARN 9912 --- [           main]
    JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view
    is enabled by default. Therefore, database queries may be
    performed during view rendering. Explicitly configure
    spring.jpa.open-in-view to disable this warning
65 2022-07-04 21:52:39.709 INFO 9912 --- [           main]
    o.e.j.s.h.ContextHandler.application     : Initializing Spring
    DispatcherServlet 'dispatcherServlet'
66 2022-07-04 21:52:39.709 INFO 9912 --- [           main]
    o.s.web.servlet.DispatcherServlet       : Initializing Servlet
    'dispatcherServlet'
67 2022-07-04 21:52:39.710 INFO 9912 --- [           main]
    o.s.web.servlet.DispatcherServlet       : Completed
    initialization in 1 ms
68 2022-07-04 21:52:39.728 INFO 9912 --- [           main]
    o.e.jetty.server.AbstractConnector      : Started
    ServerConnector@548d5ed3{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
69 2022-07-04 21:52:39.729 INFO 9912 --- [           main]
    o.s.b.web.embedded.jetty.JettyWebServer : Jetty started on
    port(s) 8080 (http/1.1) with context path '/'
70 2022-07-04 21:52:39.739 INFO 9912 --- [           main]
    c.c.t.TutorialTransactionApplication   : Started
    TutorialTransactionApplication in 4.589 seconds (JVM running for
    5.01)
71
```

Deploy SpringBoot WAR file to External Tomcat Server

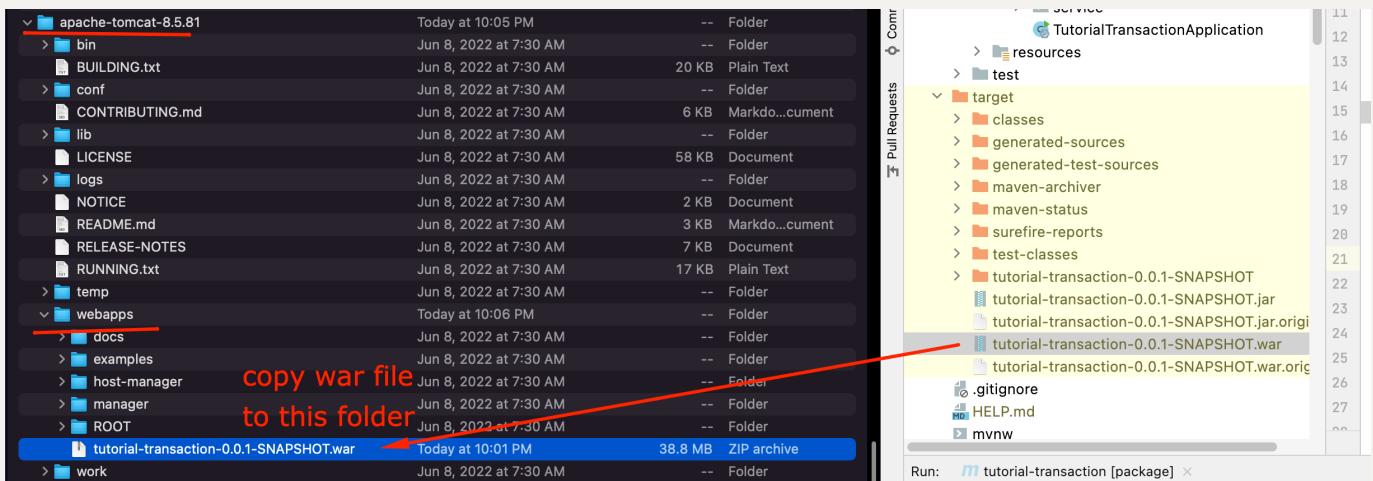
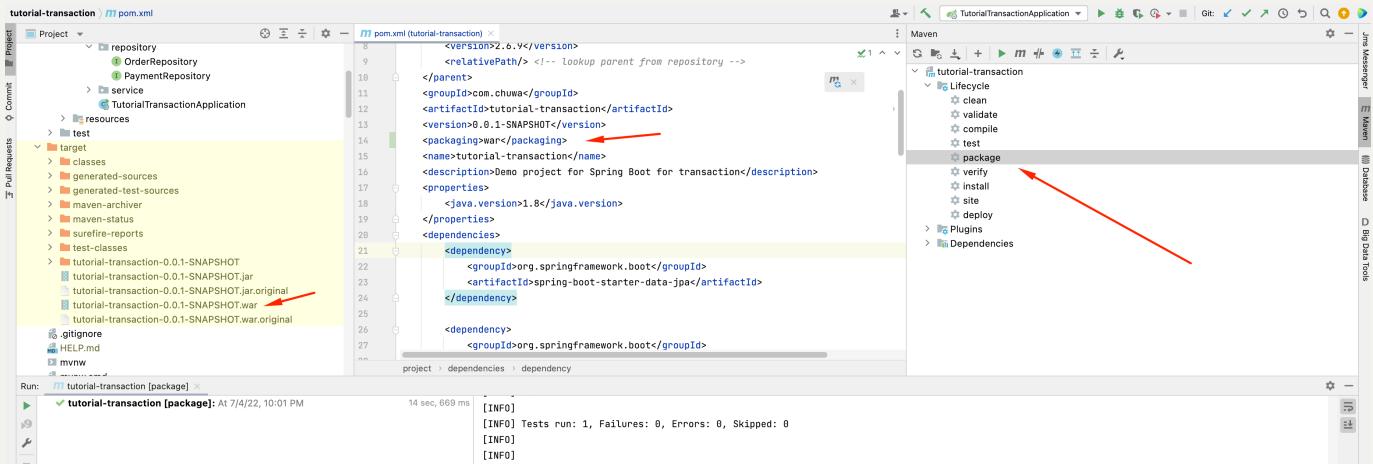
Repo: <https://github.com/TAIsRich/tutorial-transaction.git>

Branch: war-deploy

```

1 <groupId>com.chuwa</groupId>
2 <artifactId>tutorial-transaction</artifactId>
3 <version>0.0.1-SNAPSHOT</version>
4 <packaging>war</packaging>
5 <name>tutorial-transaction</name>
6 <description>Demo project for Spring Boot for transaction</description>

```



```

1 $> cd bin
2 $> chmod +x catalina.sh
3 $>

```

```
1 → bin sudo bash startup.sh
2 Using CATALINA_BASE:      /Users/bigo/Documents/software/apache-
tomcat-8.5.81
3 Using CATALINA_HOME:      /Users/bigo/Documents/software/apache-
tomcat-8.5.81
4 Using CATALINA_TMPDIR:   /Users/bigo/Documents/software/apache-
tomcat-8.5.81/temp
5 Using JRE_HOME:           /Library/Java/JavaVirtualMachines/jdk-
11.0.15.jdk/Contents/Home
6 Using CLASSPATH:          /Users/bigo/Documents/software/apache-
tomcat-
8.5.81/bin/bootstrap.jar:/Users/bigo/Documents/software/apache-
tomcat-8.5.81/bin/tomcat-juli.jar
7 Using CATALINA_OPTS:
8 Tomcat started.
```

Check local host 8080, it is running. if your application has GET API, try it. **add war file name in the path**

```
localhost:8080/tutorial-transaction-0.0.1-SNAPSHOT/posts
```

localhost:8080

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/8.5.81



If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:

- [Security Considerations How-To](#)
- [Manager Application How-To](#)
- [Clustering/Session Replication How-To](#)

Server Status

Manager App

Host Manager

Developer Quick Start

Tomcat Setup	Realms & AAA	Examples	Servlet Specifications
First Web Application	JDBC DataSources		Tomcat Versions

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in:
`$CATALINA_HOME/conf/tomcat-users.xml`

In Tomcat 8.5 access to the manager application is split between different users.
[Read more...](#)

Release Notes

Changelog

Migration Guide

Security Notices

Documentation

[Tomcat 8.5 Documentation](#)

[Tomcat 8.5 Configuration](#)

[Tomcat Wiki](#)

Find additional important configuration information in:
`$CATALINA_HOME/RUNNING.txt`

Developers may be interested in:

[Tomcat 8.5 Bug Database](#)

[Tomcat 8.5 JavaDocs](#)

[Tomcat 8.5 Git Repository at GitHub](#)

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume.)

tomcat-users
User support and discussion

taglibs-user
User support and discussion for [Apache Taglibs](#)

tomcat-dev
Development mailing list, including commit messages

Logging(重点， easy)

Repo: <https://github.com/TAIsRich/Blog-Securing-APIs.git>

Branch: log

What is the logging?

The screenshot shows an IDE interface with multiple tabs open. The top tab is 'application.properties' containing configuration for a MySQL database and Hibernate dialect.

```

server.port = 8088
# datasource
spring.datasource.url = jdbc:mysql://localhost:3306/myblog?useSSL=false&serverTimezone=UTC
spring.datasource.username= springstudent
spring.datasource.password= springstudent
# hibernate properties
spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.MySQL5InnoDBDialect

```

The bottom section is a 'Console' tab showing a log of application startup and configuration. Several log entries are highlighted with red boxes:

- [restartedMain] INFO 47921 --- [restartedMain] com.chuwa.blog.BlogRestApiApplication : Starting BlogRestApiApplication using Java 1.8.0_271 on Yanans-MBP with PID 47921 (C:/Users/bigo/Documents/githubProjects/chuwa/blog-Securing-APIs/target/classes started by bigo in C:/Users/bigo/Documents/githubProjects/chuwa/blog-Securing-APIs)
- [restartedMain] INFO 47921 --- [restartedMain] .DevToolsPropertyDefaultsPostProcessor : No active profile set, falling back to 1 default profile: "default"
- [restartedMain] .DevToolsPropertyDefaultsPostProcessor : DevTools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
- [restartedMain] .DevToolsPropertyPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'
- [restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
- [restartedMain] INFO 47921 --- [restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 51 ms. Found 4 JPA repository interfaces.
- [restartedMain] INFO 47921 --- [restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8088 (http)
- [restartedMain] INFO 47921 --- [restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
- [restartedMain] INFO 47921 --- [restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.63]
- [restartedMain] o.a.c.c.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
- [restartedMain] INFO 47921 --- [restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1669 ms
- [restartedMain] WARN 47921 --- [restartedMain] JpaBaseConfiguration\$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
- [restartedMain] INFO 47921 --- [restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
- [restartedMain] INFO 47921 --- [restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.9.Final
- [restartedMain] INFO 47921 --- [restartedMain] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations (5.1.2.Final)
- [restartedMain] INFO 47921 --- [restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
- [restartedMain] INFO 47921 --- [restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
- [restartedMain] INFO 47921 --- [restartedMain] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQL5InnoDBDialect
- [restartedMain] INFO 47921 --- [restartedMain] o.h.e.t.j.p.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
- [restartedMain] INFO 47921 --- [restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
- [restartedMain] DEBUG 47921 --- [restartedMain] .PrePostAnnotationSecurityMetadataSource : [org.springframework.security.access.prepost.PreAuthorize(value='hasRole('ADMIN'))] found on specific method: public org.springframework.http.ResponseEntity<com.chuwa.blog.controller.PostController.createPost(com.chuwa.blog.payload.PostDto)
- [restartedMain] DEBUG 47921 --- [restartedMain] .controller.PostController.createPost(com.chuwa.blog.controller.PostController) with attributes [[@authorize: 'hasRole('ADMIN')', filter: 'null', filterTarget: 'null']]
- [restartedMain] DEBUG 47921 --- [restartedMain] .PrePostAnnotationSecurityMetadataSource : [org.springframework.security.access.prepost.PreAuthorize(value='hasRole('ADMIN'))] found on specific method: public org.springframework.http.ResponseEntity<com.chuwa.blog.controller.PostController.updatePostById(com.chuwa.blog.payload.PostDto, long)]
- [restartedMain] DEBUG 47921 --- [restartedMain] .DelegatingMethodSecurityMetadataSource : Caching method [CacheKey@com.chuwa.blog.controller.PostController; public org.springframework.http.ResponseEntity<com.chuwa.blog.controller.PostController.updatePostById(com.chuwa.blog.payload.PostDto, long)] with attributes [[@authorize: 'hasRole('ADMIN')', filter: 'null', filterTarget: 'null']]
- [restartedMain] DEBUG 47921 --- [restartedMain] .PrePostAnnotationSecurityMetadataSource : [org.springframework.security.access.prepost.PreAuthorize(value='hasRole('ADMIN'))] found on specific method: public org.springframework.http.ResponseEntity<com.chuwa.blog.controller.PostController.deletePost(long)]

Date(When) + LEVEL + class name(Who) + log information(What)

什么时候(when)某人(who)干了什么级别(level)的事情what

```

1 System.out.println(new Date() + " " + LEVEL +
AuthJWTController.class.getName() + " : " + "is called")
2 // vs.
3 private static final Logger logger =
LoggerFactory.getLogger(AuthJWTController.class);
4 logger.info("is called");

```

Level

```

1 logger.trace("A TRACE Message");
2 logger.debug("A DEBUG Message");
3 logger.info("An INFO Message");
4 logger.warn("A WARN Message");
5 logger.error("An ERROR Message");

```

Example

```
1  @PostMapping("/signin")
2      public ResponseEntity<JWTAuthResponse>
3          authenticateUser(@RequestBody LoginDto loginDto) {
4              logger.trace("A TRACE Message");
5              logger.debug("A DEBUG Message");
6              logger.info("An INFO Message");
7              logger.warn("A WARN Message");
8              logger.error("An ERROR Message");
9              logger.info(loginDto.getAccountOrEmail() + "is trying to
sign in our application");
10             Authentication authentication =
11                 authenticationManager.authenticate(new
12                     UsernamePasswordAuthenticationToken(
13                         loginDto.getAccountOrEmail(),
14                         loginDto.getPassword()
15                     ));
16
17             SecurityContextHolder.getContext().setAuthentication(authentication);
18
19             // get token from tokenProvider
20             String token =
21                 tokenProvider.generateToken(authentication);
22
23             logger.info(loginDto.getAccountOrEmail() + "sign in
successfully");
24
25             return ResponseEntity.ok(new JWTAuthResponse(token));
26         }
27     }
```

```

    @Autowired
    private JwtTokenProvider tokenProvider;

    @PostMapping(RegExp = "/signin")
    public ResponseEntity<JWTAuthResponse> authenticateUser(@RequestBody LoginDto loginDto) {
        logger.trace("A TRACE Message");
        logger.debug("A DEBUG Message");
        logger.info("An INFO Message");
        logger.warn("A WARN Message");
        logger.error("An ERROR Message");
        logger.info(loginDto.getAccountOrEmail() + " is trying to sign in our application");

        Authentication authentication = authenticationManager.authenticate(new UsernamePasswordAuthenticationToken(
            loginDto.getAccountOrEmail(), loginDto.getPassword()
        ));

        SecurityContextHolder.getContext().setAuthentication(authentication);

        // get token from tokenProvider
        String token = tokenProvider.generateToken(authentication);

        logger.info(loginDto.getAccountOrEmail() + " sign in successfully");
        return ResponseEntity.ok(new JWTAuthResponse(token));
    }

```

2022-07-05 23:32:24.269 DEBUG 47921 --- [nio-8088-exec-2] o.s.security.web.FilterChainProxy : Securing POST /api/auth/jwt/signin
2022-07-05 23:32:24.286 DEBUG 47921 --- [nio-8088-exec-2] s.s.w.c.SecurityContextPersistenceFilter : Set SecurityContextHolder to empty SecurityContext
2022-07-05 23:32:24.319 DEBUG 47921 --- [nio-8088-exec-2] o.s.s.w.a.AnonymousAuthenticationFilter : Set SecurityContextHolder to anonymous SecurityContext
2022-07-05 23:32:24.319 DEBUG 47921 --- [nio-8088-exec-2] o.s.s.w.session.SessionManagementFilter : Request requested invalid session id 20EC4C83D9E9E91303B84C3B406987D03C
2022-07-05 23:32:24.350 DEBUG 47921 --- [nio-8088-exec-2] o.s.s.w.a.i.FilterSecurityInterceptor : Authorized filter invocation [POST /api/auth/jwt/signin] with attributes [permitAll]
2022-07-05 23:32:24.351 DEBUG 47921 --- [nio-8088-exec-2] o.s.security.web.FilterChainProxy : Secured POST /api/auth/jwt/signin
2022-07-05 23:32:24.551 INFO 47921 --- [nio-8088-exec-2] c.c.blog.controller.AuthJWTController : An INFO Message
2022-07-05 23:32:24.551 WARN 47921 --- [nio-8088-exec-2] c.c.blog.controller.AuthJWTController : A WARN Message
2022-07-05 23:32:24.552 ERROR 47921 --- [nio-8088-exec-2] c.c.blog.controller.AuthJWTController : An ERROR Message
2022-07-05 23:32:24.552 INFO 47921 --- [nio-8088-exec-2] c.c.blog.controller.AuthJWTController : chuwaAccis trying to sign in our application
2022-07-05 23:32:25.036 DEBUG 47921 --- [nio-8088-exec-2] o.s.s.a.dao.DaoAuthenticationProvider : Authenticated user
2022-07-05 23:32:25.225 INFO 47921 --- [nio-8088-exec-2] c.c.blog.controller.AuthJWTController : chuwaAccsign in successfully
2022-07-05 23:32:25.256 DEBUG 47921 --- [nio-8088-exec-2] s.s.w.c.SecurityContextPersistenceFilter : Cleared SecurityContextHolder to complete request

```

    logger.info("service getAllPost with pageable are called");
    logger.info("creating a sort object");
    Sort sort = sortDir.equalsIgnoreCase(Sort.Direction.ASC.name()) ? Sort.by(sortBy).ascending()
        : Sort.by(sortBy).descending();

    // create pageable instance
    logger.info("creating a PageRequest object");
    PageRequest pageRequest = PageRequest.of(pageNo, pageSize, sort);
    logger.info("calling postRepository to get the data from database");
    Page<Post> pagePosts = postRepository.findAll(pageRequest);

    // get content for page object
    logger.info("Fetching data successfully and converting data to Dtos");
    List<Post> posts = pagePosts.getContent();
    List<PostDto> postDtos = posts.stream().map(post -> mapToDTO(post)).collect(Collectors.toList());

    logger.info("adding meta data to the response");
    PostResponse postResponse = new PostResponse();
    postResponse.setContent(postDtos);
    postResponse.setPageNo(pagePosts.getNumber());
    postResponse.setPageSize(pagePosts.getTotalSize());

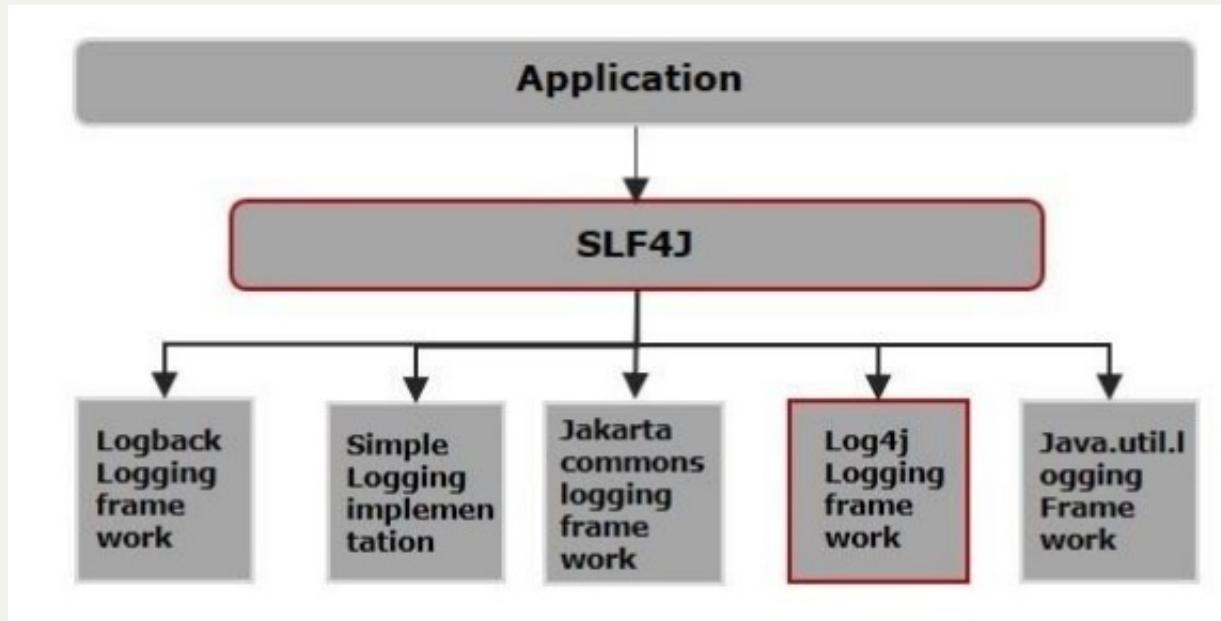
```

2022-07-05 23:54:03.878 DEBUG 58020 --- [nio-8088-exec-1] o.s.s.w.a.i.FilterSecurityInterceptor : Authorized filter invocation [POST /api/auth/jwt/signin] with attributes [permitAll]
2022-07-05 23:54:03.878 DEBUG 58020 --- [nio-8088-exec-1] o.s.security.web.FilterChainProxy : Secured POST /api/auth/jwt/signin
2022-07-05 23:54:03.169 INFO 58020 --- [nio-8088-exec-1] c.c.blog.controller.AuthJWTController : An INFO Message
2022-07-05 23:54:03.169 WARN 58020 --- [nio-8088-exec-1] c.c.blog.controller.AuthJWTController : A WARN Message
2022-07-05 23:54:03.169 ERROR 58020 --- [nio-8088-exec-1] c.c.blog.controller.AuthJWTController : An ERROR Message
2022-07-05 23:54:03.169 INFO 58020 --- [nio-8088-exec-1] c.c.blog.controller.AuthJWTController : chuwaAccis trying to sign in our application
2022-07-05 23:54:03.516 DEBUG 58020 --- [nio-8088-exec-1] o.s.s.a.dao.DaoAuthenticationProvider : Authenticated user
2022-07-05 23:54:03.698 INFO 58020 --- [nio-8088-exec-1] c.c.blog.controller.AuthJWTController : chuwaAccsign in successfully
2022-07-05 23:54:03.731 DEBUG 58020 --- [nio-8088-exec-1] s.s.w.c.SecurityContextPersistenceFilter : Cleared SecurityContextHolder to complete request
2022-07-05 23:54:48.525 DEBUG 58020 --- [nio-8088-exec-3] o.s.security.web.FilterChainProxy : Securing GET /api/posts
2022-07-05 23:54:48.525 DEBUG 58020 --- [nio-8088-exec-3] s.s.w.c.SecurityContextPersistenceFilter : Set SecurityContextHolder to empty SecurityContext
2022-07-05 23:54:48.567 DEBUG 58020 --- [nio-8088-exec-3] o.s.s.w.a.i.FilterSecurityInterceptor : Authorized filter invocation [GET /api/posts] with attributes [permitAll]
2022-07-05 23:54:48.567 DEBUG 58020 --- [nio-8088-exec-3] o.s.security.web.FilterChainProxy : Secured GET /api/posts
2022-07-05 23:54:48.580 INFO 58020 --- [nio-8088-exec-3] c.c.blog.service.impl.PostServiceImpl : service getAllPost with pageable are called
2022-07-05 23:54:48.580 INFO 58020 --- [nio-8088-exec-3] c.c.blog.service.impl.PostServiceImpl : creating a sort object
2022-07-05 23:54:48.581 INFO 58020 --- [nio-8088-exec-3] c.c.blog.service.impl.PostServiceImpl : creating a PageRequest object
2022-07-05 23:54:48.581 INFO 58020 --- [nio-8088-exec-3] c.c.blog.service.impl.PostServiceImpl : calling postRepository to get the data from database
2022-07-05 23:54:48.617 INFO 58020 --- [nio-8088-exec-3] c.c.blog.service.impl.PostServiceImpl : Fetching data successfully and converting data to Dtos
2022-07-05 23:54:48.668 INFO 58020 --- [nio-8088-exec-3] c.c.blog.service.impl.PostServiceImpl : adding meta data to the response
2022-07-05 23:54:48.684 DEBUG 58020 --- [nio-8088-exec-3] s.s.w.c.SecurityContextPersistenceFilter : Cleared SecurityContextHolder to complete request

Questions: What can we do with those logging infomations?

How do you do debug?

Logging Architecture



see more details by reading this article: <https://www.baeldung.com/spring-boot-logging>

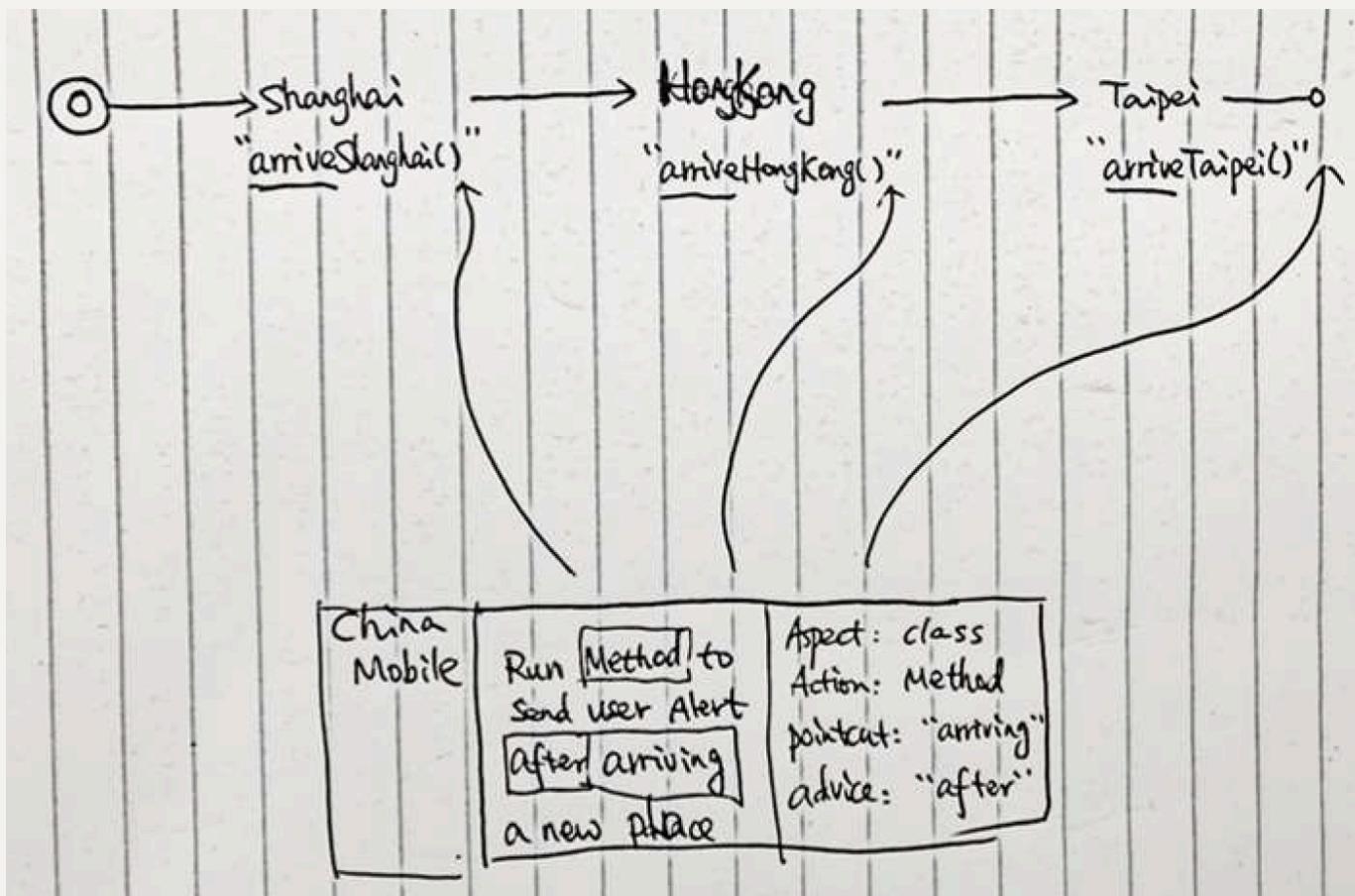
Spring AOP(easy, 基本sense)

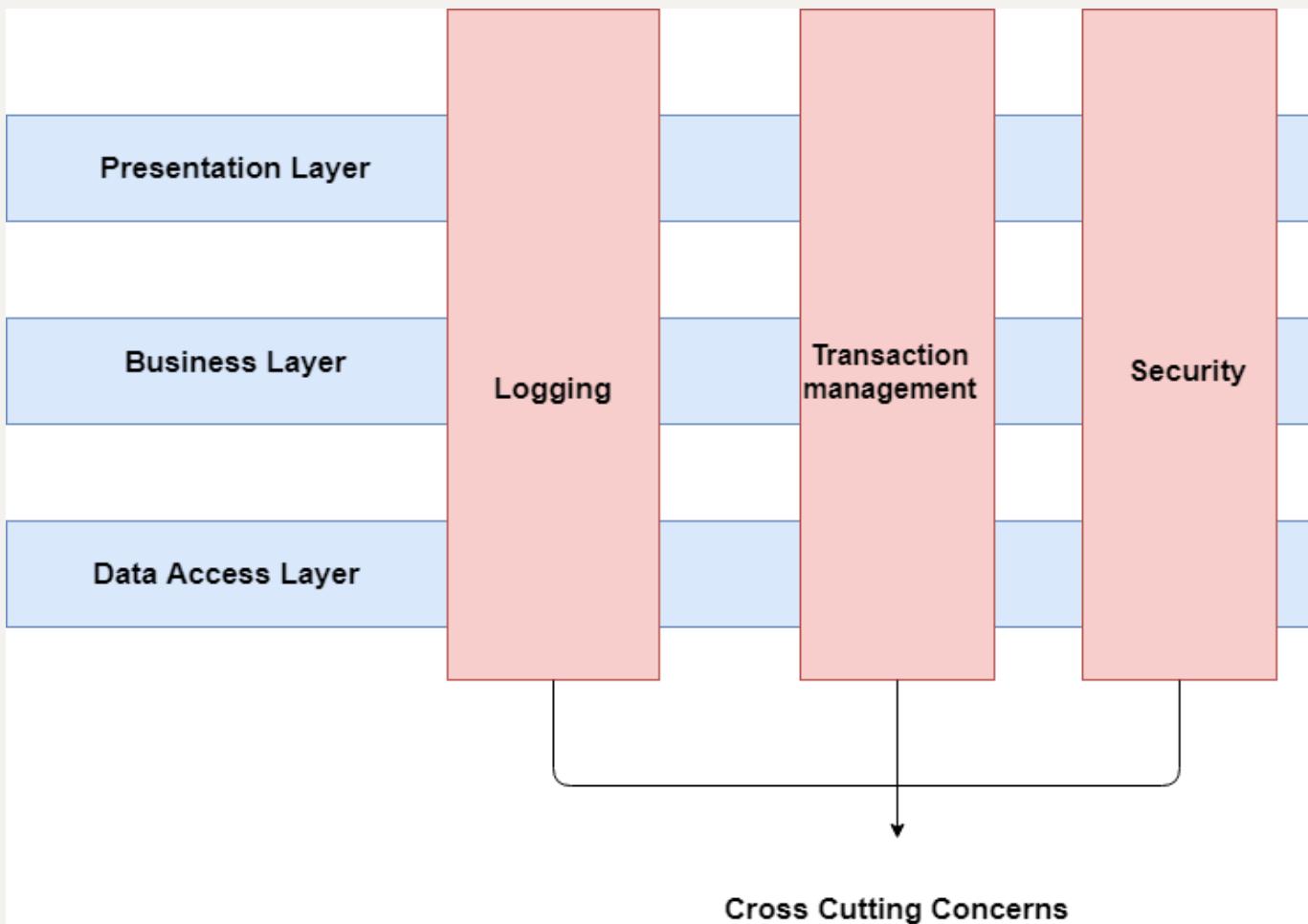
Repo: <https://github.com/TAIsRich/Blog-Securing-APIs.git>

Branch: aop

(run failure, I need to revise it later)

cross-cutting, Aspect Oriented Programming





Usage

- Logging
- Tracing
- Exception Handling
- Security Management

Without AOP

We need to write the log for sign in and sign up.

```
private static final Logger logger = LoggerFactory.getLogger(AuthJWTController.class);

@PostMapping("/signin")
public ResponseEntity<JWTAuthResponse> authenticateUser(@RequestBody LoginDto loginDto) {
    logger.info(loginDto.getAccountOrEmail() + "is trying to sign in our application");

    Authentication authentication = authenticationManager.authenticate(new UsernamePasswordAuthenticationToken(
        loginDto.getAccountOrEmail(), loginDto.getPassword()
    ));

    SecurityContextHolder.getContext().setAuthentication(authentication);

    // get token from tokenProvider
    String token = tokenProvider.generateToken(authentication);

    logger.info(loginDto.getAccountOrEmail() + "sign in successfully");
    return ResponseEntity.ok(new JWTAuthResponse(token));
}
```

```

    @PostMapping("/signup")
    public ResponseEntity<?> registerUser(@RequestBody SignUpDto signUpDto) {
        logger.info("New User is trying to sign up our application");

        // check if username is in a DB
        if (userRepository.existsByAccount(signUpDto.getAccount())) {
            return new ResponseEntity<?>( body: "Username is already taken!", HttpStatus.BAD_REQUEST);
        }

        // check if email exists in DB
        if (userRepository.existsByEmail(signUpDto.getEmail())) {
            return new ResponseEntity<?>( body: "Email is already taken!", HttpStatus.BAD_REQUEST);
        }

        // create user object
        User user = new User();
        user.setName(signUpDto.getName());
        user.setAccount(signUpDto.getAccount());
        user.setEmail(signUpDto.getEmail());
        user.setPassword(passwordEncoder.encode(signUpDto.getPassword()));

        Role roles = roleRepository.findByName("ROLE_ADMIN").get();
        user.setRoles(Collections.singleton(roles));
        userRepository.save(user);

        logger.info("User registered successfully");
        return new ResponseEntity<?>( body: "User registered successfully", HttpStatus.CREATED);
    }
}

```

AOP Concepts

Aspect - class to define all aop self methos

PointCut: expression to find all main application methods to insert AOP

Advice: before/after/around/afterThrowing - When to execute aop methods when the application pointcut is found.

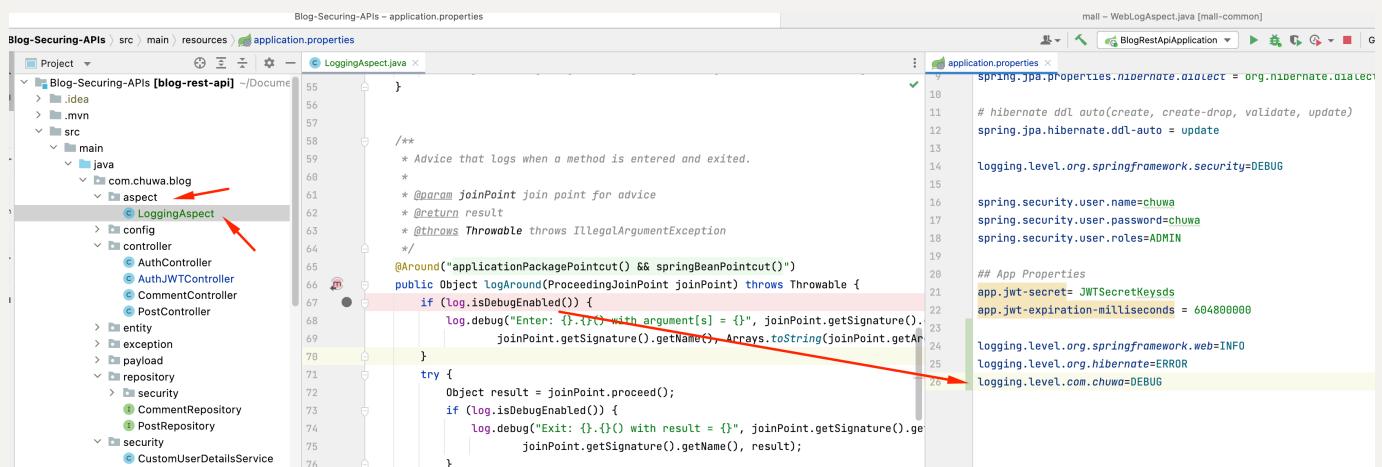
- **@Before** – Run before the method execution
- **@After** – Run after the method returned a result
- **@AfterReturning** – Run after the method returned a result, intercept the returned result as well.
- **@AfterThrowing** – Run after the method throws an exception

- **@Around** – Run around the method execution, combine all three advices above.
 - **@Aspect** - indicate this is a aop class
 - **@Component** - This annotation mark the beans as Spring's managed components
 - **@Before** annotated methods run exactly before all methods matching with pointcut expression

Examples

```
1 <dependency>
2     <groupId>org.springframework.boot</groupId>
3     <artifactId>spring-boot-starter-aop</artifactId>
4 </dependency>
```

```
1 logging.level.org.springframework.web=INFO  
2 logging.level.org.hibernate=ERROR  
3 logginq.level.com.chuwa=DEBUG
```



```

2022-07-06 01:14:55.121 INFO 79459 --- [nio-8088-exec-3] c.c.blog.controller.AuthJwtController : chuwa login successfully
2022-07-06 01:14:55.121 DEBUG 79459 --- [nio-8088-exec-3] com.chuwa.blog.aspect.LoggingAspect : Exit: com.chuwa.blog.controller.AuthJwtController.authenticateUser() with result = <200 OK,com.chuwa.blog.payload.security.JWTAuthResponse@569aba75,[]>
2022-07-06 01:14:55.123 DEBUG 79459 --- [nio-8088-exec-3] s.s.w.c.SecurityContextPersistenceFilter : Cleared SecurityContextHolder to complete request
2022-07-06 01:14:56.792 DEBUG 79459 --- [nio-8088-exec-4] o.s.security.web.FilterChainProxy : Securing GET /api/posts
2022-07-06 01:14:56.792 DEBUG 79459 --- [nio-8088-exec-4] s.s.w.c.SecurityContextPersistenceFilter : Set SecurityContextHolder to empty SecurityContext
2022-07-06 01:15:06.819 DEBUG 79459 --- [nio-8088-exec-4] com.chuwa.blog.aspect.LoggingAspect : Enter: com.chuwa.blog.security.CustomUserDetailsService.loadUserByUsername() with argument[s] = [chuwa@gmail.com]
2022-07-06 01:15:06.842 DEBUG 79459 --- [nio-8088-exec-4] com.chuwa.blog.aspect.LoggingAspect : Exit: com.chuwa.blog.security.CustomUserDetailsService.loadUserByUsername() with result = org.springframework.security.core.userdetails.User [Username=chuwa@gmail.com, Password=[PROTECTED], Enabled=true, AccountNonExpired=true, AccountNonLocked=true, Granted Authorities=[ROLE_ADMIN]]
2022-07-06 01:15:06.843 DEBUG 79459 --- [nio-8088-exec-4] o.s.w.a.i.FilterSecurityInterceptor : Authorized filter invocation [GET /api/posts] with attributes [permitAll]
2022-07-06 01:15:06.843 DEBUG 79459 --- [nio-8088-exec-4] o.s.w.c.SecurityContextPersistenceFilter : Secured GET /api/posts
2022-07-06 01:15:06.852 DEBUG 79459 --- [nio-8088-exec-4] com.chuwa.blog.aspect.LoggingAspect : Enter: com.chuwa.blog.controller.PostController.getAllPosts() with argument[s] = [0, 10, id, asc]
2022-07-06 01:15:06.852 DEBUG 79459 --- [nio-8088-exec-4] com.chuwa.blog.service.impl.PostServiceImpl.getAllPost() with argument[s] = [0, 10, id, asc]
2022-07-06 01:15:06.852 DEBUG 79459 --- [nio-8088-exec-4] com.chuwa.blog.aspect.LoggingAspect : Exit: com.chuwa.blog.service.impl.PostServiceImpl.getAllPost() with result = PostResponse{content=[com.chuwa.blog.payload.PostDto@32a597b6, com.chuwa.blog.payload.PostDto@6868fe8df, com.chuwa.blog.payload.PostDto@79d7dfbb, com.chuwa.blog.payload.PostDto@7d02f108, com.chuwa.blog.payload.PostDto@6384c131, com.chuwa.blog.payload.PostDto@7d2255ae, com.chuwa.blog.payload.PostDto@32a597b6, com.chuwa.blog.payload.PostDto@6868fe8df, com.chuwa.blog.payload.PostDto@79d7dfbb, com.chuwa.blog.payload.PostDto@7d02f108, com.chuwa.blog.payload.PostDto@6384c131, com.chuwa.blog.payload.PostDto@7d2255ae, com.chuwa.blog.payload.PostDto@32a597b6], pageNo=0, pageSize=10, totalElements=7, totalPages=1, last=true}
2022-07-06 01:15:06.948 DEBUG 79459 --- [nio-8088-exec-4] com.chuwa.blog.aspect.LoggingAspect : Enter: com.chuwa.blog.controller.PostController.getAllPosts() with result = PostResponse{content=[com.chuwa.blog.payload.PostDto@32a597b6, com.chuwa.blog.payload.PostDto@6868fe8df, com.chuwa.blog.payload.PostDto@79d7dfbb, com.chuwa.blog.payload.PostDto@7d02f108, com.chuwa.blog.payload.PostDto@6384c131, com.chuwa.blog.payload.PostDto@7d2255ae, com.chuwa.blog.payload.PostDto@32a597b6], pageNo=0, pageSize=10, totalElements=7, totalPages=1, last=true}
2022-07-06 01:15:06.948 DEBUG 79459 --- [nio-8088-exec-4] s.s.w.c.SecurityContextPersistenceFilter : Cleared SecurityContextHolder to complete request

```

Debug to show how AOP Work

Object result = joinPoint.proceed(); call the real method

AOP is a Wrapper, wrap out method with some extra info, for example, logging.

```

1  @Around("applicationPackagePointcut() && springBeanPointcut()")
2  public Object logAround(ProceedingJoinPoint joinPoint) throws
3      Throwable {
4          if (log.isDebugEnabled()) {
5              log.debug("Enter: {}.{}() with argument[s] = {}", 
6                  joinPoint.getSignature().getDeclaringTypeName(),
7                  joinPoint.getSignature().getName(),
8                  Arrays.toString(joinPoint.getArgs()));
9          }
10         try {
11             Object result = joinPoint.proceed();
12             if (log.isDebugEnabled()) {
13                 log.debug("Exit: {}.{}() with result = {}", 
14                     joinPoint.getSignature().getDeclaringTypeName(),
15                     joinPoint.getSignature().getName(), result);
16             }
17             return result;
18         } catch (IllegalArgumentException e) {
19             log.error("Illegal argument: {} in {}.{}()", 
20                 Arrays.toString(joinPoint.getArgs()),
21                 joinPoint.getSignature().getDeclaringTypeName(),
22                 joinPoint.getSignature().getName());
23     }

```

```

17     throw e;
18 }
19 }
```

The screenshot shows an IDE interface with two code editors. The left editor contains `LoggingAspect.java` and the right editor contains `WebLogAspect.java`. Both files are annotated with `@Aspect` and `@Component`. The `LoggingAspect` file includes several pointcuts and advice methods. The `WebLogAspect` file includes annotations for `@Before`, `@AfterReturning`, and `@Around` aspects.

```

Blog-Securing-APIs - LoggingAspect.java
Blog-Securing-APIs [blog-rest-api] - Document
Project ~ main.java.com.chuwa.blog.aspect.LoggingAspect
Blog-Securing-APIs [blog-rest-api] - Document
main.java.com.chuwa.blog.aspect.LoggingAspect
LoggingAspect
18 * @date 7/6/22 12:44 AM
19 */
20 @Aspect
21 @Component
22 @Order(2)
23 public class LoggingAspect {
24     private final Logger log = LoggerFactory.getLogger(this.getClass());
25
26     /**
27      * Pointcut that matches all repositories, services and Web REST endpoints.
28      */
29     @Pointcut("within(@org.springframework.stereotype.Repository *) & " +
30             " || within(@org.springframework.stereotype.Service *) & " +
31             " || within(@org.springframework.web.bind.annotation.RestController *)")
32     public void springBeanPointcut() {
33         // Method is empty as this is just a Pointcut, the implementations are in the advice.
34     }
35
36     /**
37      * Pointcut that matches all Spring beans in the application's main packages.
38      */
39     @Pointcut("within(com.chuwa.blog.security..*) & " +
40             " || within(com.chuwa.blog.service..*) & " +
41             " || within(com.chuwa.blog.controller..*) & " +
42             " || within(com.chuwa.blog.repository..*)")
43     public void applicationPackagePointcut() {
44         // Method is empty as this is just a Pointcut, the implementations are in the advice.
45     }
46
47     /**
48      * Advice that logs methods throwing exceptions.
49      */
50     @Around("applicationPackagePointcut()")
51     public void logAfterThrowing(JoinPoint joinPoint, Throwable e) {
52         log.error("Exception in {}.{}() with cause = {}", joinPoint.getSignature().getName(), e.getCause() != null ? e.getCause() :
53             e);
54     }
55
56     /**
57      * @param joinPoint join point for advice
58      * @param e exception
59      */
60     @AfterThrowing(pointcut = "applicationPackagePointcut() & springBeanPointcut()", th
61     public void logAfterThrowing(JoinPoint joinPoint, Throwable e) {
62         log.error("Exception in {}.{}() with cause = {}", joinPoint.getSignature().getName(), e.getCause() != null ? e.getCause() :
63             e);
64     }
65
66     /**
67      * @param joinPoint join point for advice
68      * @param e exception
69      */
70     @AfterReturning(value = "webLog()", returning = "ret")
71     public void doAfterReturning(Object ret) throws Throwable {
72     }
73
74     /**
75      * @param joinPoint join point for advice
76      */
77     @Before("webLog()")
78     public void doBefore(JoinPoint joinPoint) throws Throwable {
79     }
80
81     /**
82      * @param joinPoint join point for advice
83      */
84     @AfterReturning(value = "webLog()", returning = "ret")
85     public void doAfterReturning(Object ret) throws Throwable {
86     }
87
88     /**
89      * @param joinPoint join point for advice
90      */
91     @Around("webLog()")
92     public Object doAround(ProceedingJoinPoint joinPoint) throws Throwable {
93         long startTime = System.currentTimeMillis();
94         // 获取当前请求对象
95         ServletRequestAttributes attributes = (ServletRequestAttributes) RequestContextHolder.getRequestAttributes();
96         HttpServletRequest request = attributes.getRequest();
97         // 记录请求信息(通过Logstash写入Elasticsearch)
98         WebLog weblog = new WebLog();
99         Object result = joinPoint.proceed();
100        Signature signature = joinPoint.getSignature();
101        MethodSignature methodSignature = (MethodSignature) signature;
102        Method method = methodSignature.getMethod();
103        if (method.isAnnotationPresent(ApiOperation.class)) {
104           ApiOperation log = method.getAnnotation(ApiOperation.class);
105            weblog.setDescription(log.value());
106        }
107        long endTime = System.currentTimeMillis();
108        String urlStr = request.getRequestURL().toString();
109        weblog.setBasePath(StrUtil.removeSuffix(urlStr, URLUtil.url(urlStr).getPath()));
110        weblog.setUsername(request.getRemoteUser());
111        weblog.setIp(request.getRemoteAddr());
112    }
113}
```

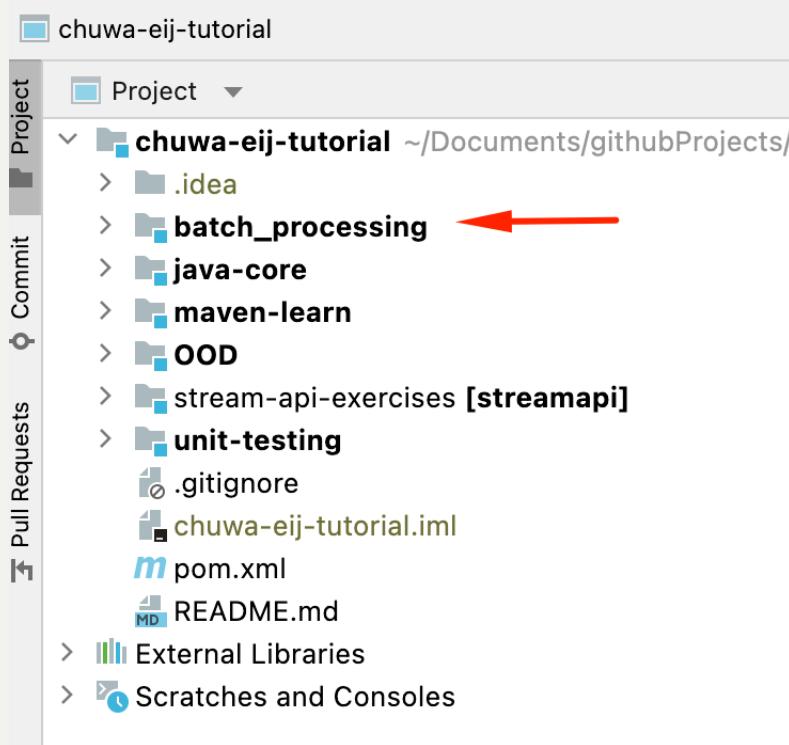
See more details: <https://www.baeldung.com/spring-aop>

Spring Batch(基本sense)

Repo: <https://github.com/TAIsRich/chuwa-eij-tutorial.git>

Branch: main

Module: batch_processing



学习要点：

1. 理解它是什么，在什么场景下要用，给出例子。
2. 实现它的粗略步骤(job, steps)

<https://spring.io/guides/gs/batch-processing/>

定时批处理数据

<https://springboot.io/t/topic/3957>

Spring MVC(基本sense，保证能回答问答题)

Repo: <https://github.com/TAIsRich/springmvc5-demo.git>

Branch: main

学习方式：理解重点，并有基本sense，保证听到概念后能去Google就行。

要点:

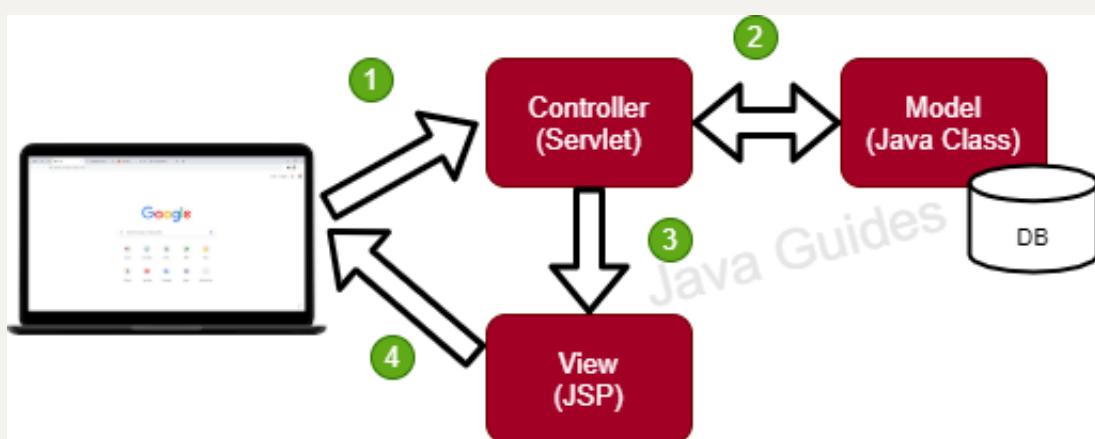
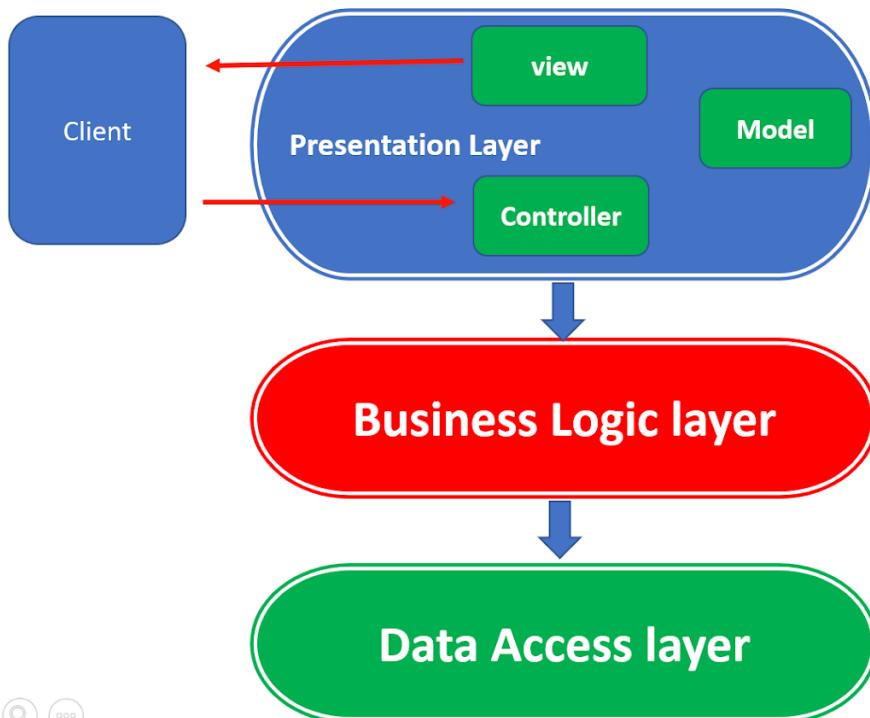
理解下面三幅图，并能口述图三的流程。

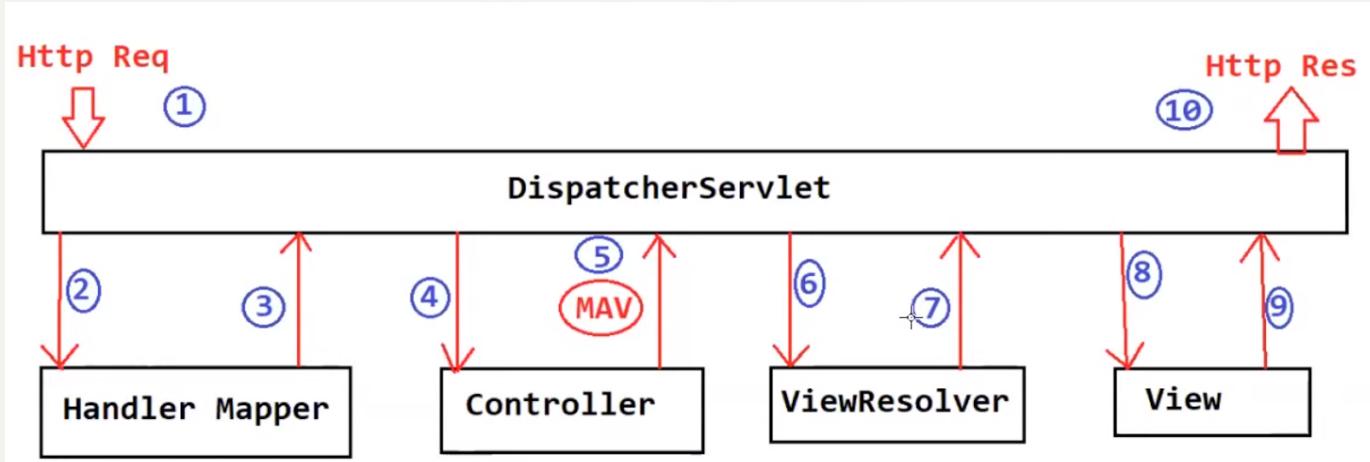
理解Tomcat的配置。

切忌:

去学JSP 细节。

Three-Tier architecture vs MVC pattern





Controller: 第4步调对应的API，该API会去调service, dao, database, 然后把数据返回给controller, controller把数据装载成MAV: Mode And View 返回给dispatcherServlet

View Resolver: 带着controller返回的数据, 以及字符串名字, 然后去找到对应的jsp文件, 并在第7步返回给DispatcherServlet

View Engine: 把JSP文件转换为纯HTML并在第9步返回给前端。

```

springmvc-demo > src > main > java > com > chuwa > training > controller > CustomerController.java
Project main
  main
    java
      com.chuwa.training
        config
          AppContext
          AppInitializer
          WebMvcConfig
        controller
        dao
        entity
        service
      resources
    webapp
      resources
        css
          bootstrap.min.css
        js
          bootstrap.min.js
          jquery-1.11.1.min.js
      WEB-INF
        views
          customer-form.jsp
          hello.jsp
          list-customers.jsp
          signup-success.jsp
        index.jsp
  target
  .gitignore
  pom.xml
  README.md

```

```

private CustomerService customerService;

@GetMapping("/list")
public String listCustomers(Model theModel) {
    List<Customer> theCustomers = customerService.getCustomers();
    theModel.addAttribute("customers", theCustomers);
    return "list-customers";
}

@GetMapping("/showForm")
public String showFormForAdd(Model theModel) {
    Customer theCustomer = new Customer();
    theModel.addAttribute("customer", theCustomer);
    return "customer-form";
}

@PostMapping("/saveCustomer")
public String saveCustomer(@ModelAttribute("customer") Customer theCustomer) {
    customerService.saveCustomer(theCustomer);
    return "redirect:/customer/list";
}

@GetMapping("/updateForm")
public String showFormForUpdate(@RequestParam("customerId") int theId,
                               Model theModel) {
    Customer theCustomer = customerService.getCustomer(theId);
    theModel.addAttribute("customer", theCustomer);
}

```

view Resolver

redirect to another API

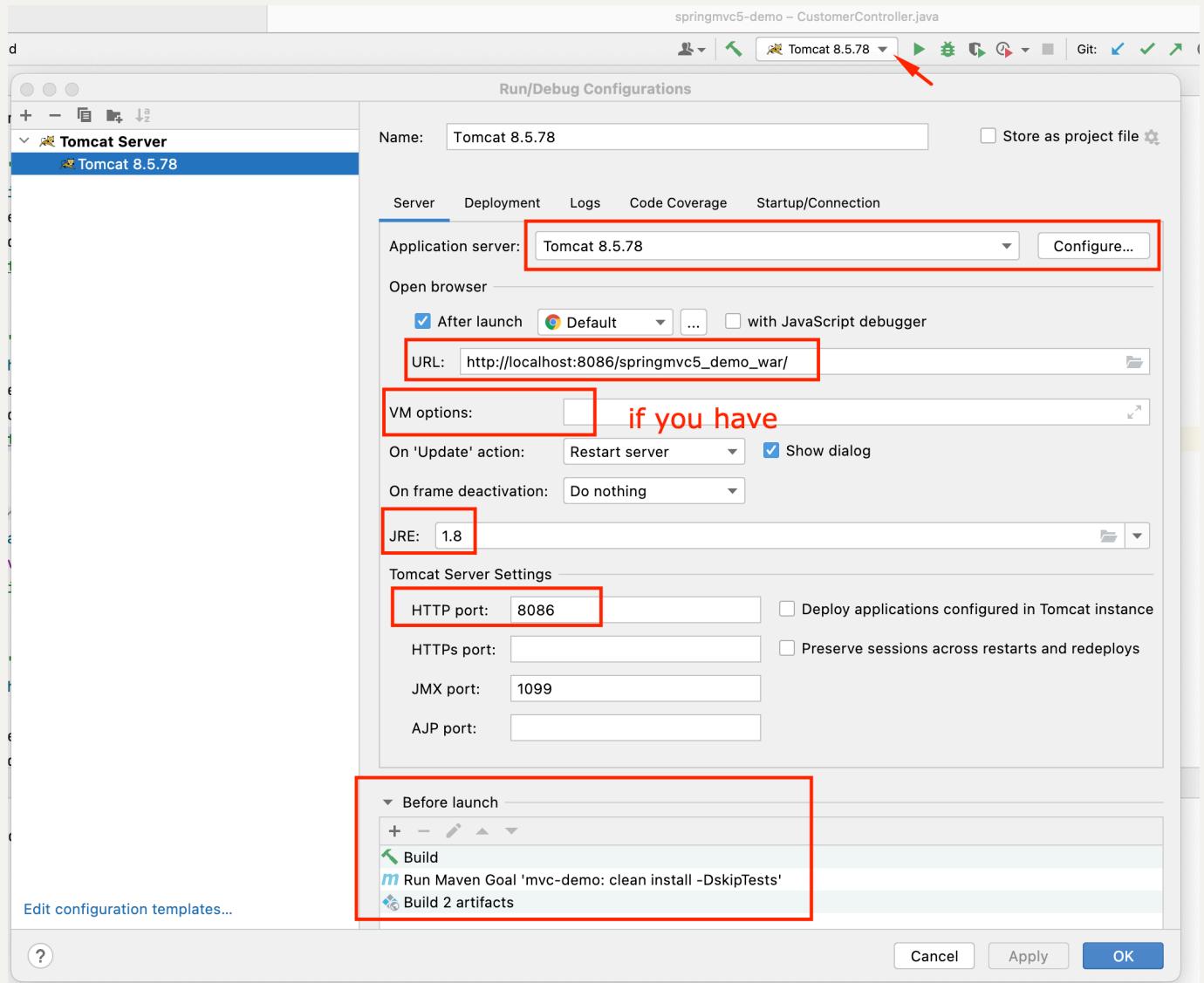
- step 1 - When the client (browser) sends an HTTP request to a specific URL. The DispatcherServlet of Spring MVC receives the request.

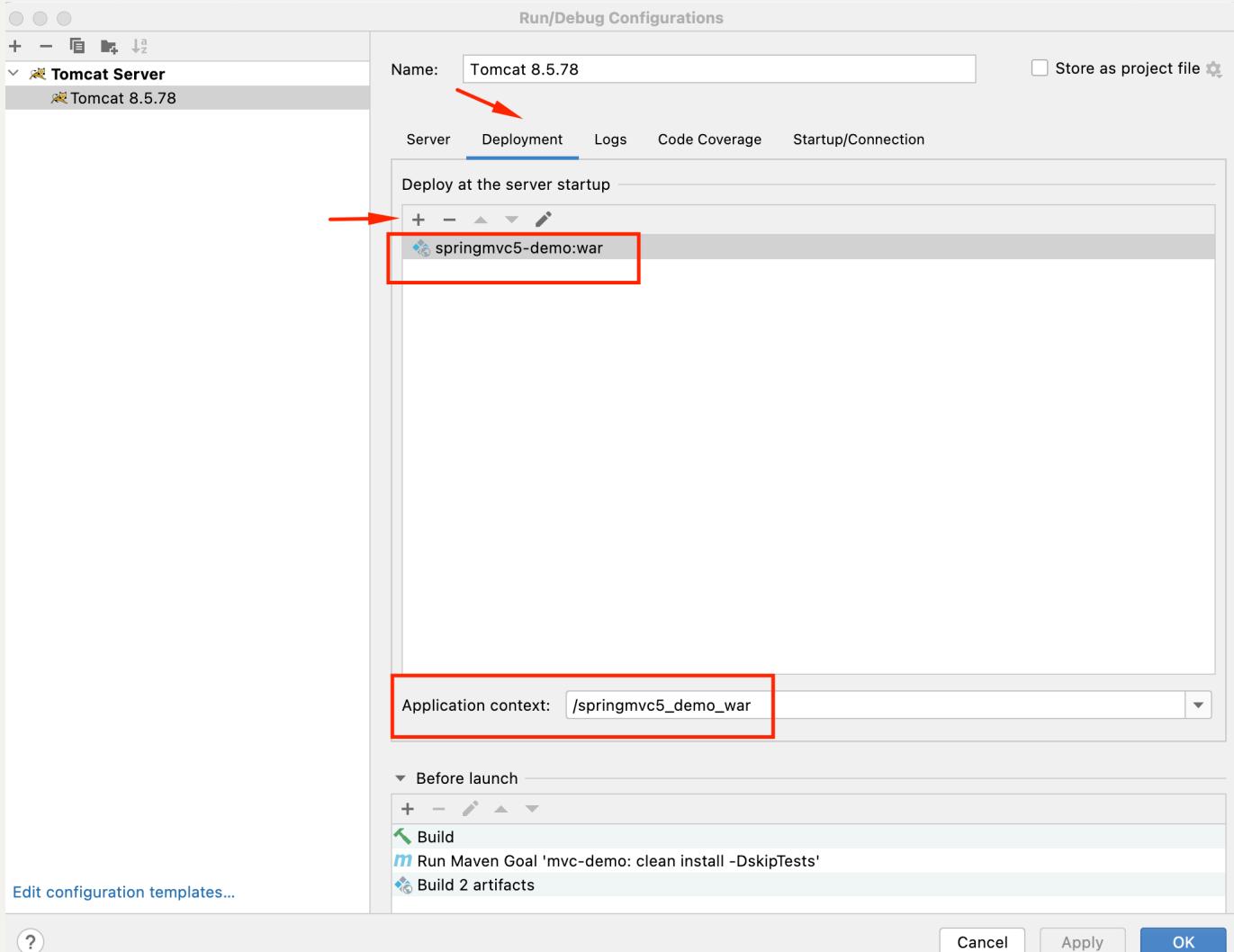
- step 2 - DispatcherServlet consult to HandlerMapper to identify which controller is responsible to handle the HTTP request.
- step 3 - HandlerMapper selects the controller which is mapped to the incoming request URL and returns the (selected Handler) and Controller details to DispatcherServlet.
- step 4 - Now DispatcherServlet knows which controller is responsible to process the request so DispatcherServlet will forward that request to the corresponding controller to process the request.
- step 5 - Now the Controller process the request, validates the request, and creates a model with data. Finally, the Controller returns the logical name of view and model to the DispatcherServlet.
- step 6 - DispatcherServlet consult ViewResolver to resolve a logical view with the physical view that exists in the application.
- step 7 - ViewResolver responsible to map logical view with actual view and return the actual view details back to the DispatcherServlet.
- step 8 - Now DispatcherServlet sends the view and model to the View component.
- step 9 - The View component merge view and model and forms a plain HTML output. Finally, the View component sends HTML output back to the DispatcherServlet.
- step 10 - The DispatcherServlet finally sends HTML output as a response back to the browser for rendering.

JSP and ModelAttribute的更多细节，如果面经里有相关的信息，则自行Google，如果無，千万不要花时间在这上面

Tomcat

try to run the application in your local machine with the downloaded Tomcat





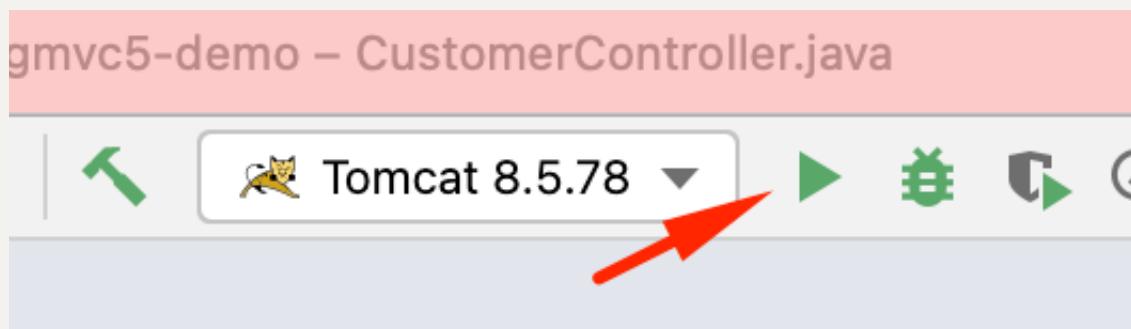
Run the application

```
1  create database mvc_demo;
```

Database

+ @localhost 2 of 15
easy_order_manager
mvc_demo
tables 1
customer

After configuring the Tomcat then click the Run button



Test the APIs based on the controller.

- http://localhost:8080/springmvc5_demo_war/
- http://localhost:8086/springmvc5_demo_war/customer/showForm
- http://localhost:8086/springmvc5_demo_war/customer/list
- POST is called by the form, so we can not call it using browser.

```
1 @Controller
2 @RequestMapping("/customer")
3 public class CustomerController {
4
5     @Autowired
```

```
6  private CustomerService customerService;
7
8  @GetMapping("/list")
9  public String listCustomers(Model theModel) {
10    List<Customer> theCustomers =
11      customerService.getCustomers();
12    theModel.addAttribute("customers", theCustomers);
13    return "list-customers";
14
15  @GetMapping("/showForm")
16  public String showFormForAdd(Model theModel) {
17    Customer theCustomer = new Customer();
18    theModel.addAttribute("customer", theCustomer);
19    return "customer-form";
20  }
21
22  @PostMapping("/saveCustomer")
23  public String saveCustomer(@ModelAttribute("customer")
24    Customer theCustomer) {
25    customerService.saveCustomer(theCustomer);
26    return "redirect:/customer/list";
27  }
28
29  @GetMapping("/updateForm")
30  public String
31    showFormForUpdate(@RequestParam("customerId") int theId,
32                      Model theModel) {
33    Customer theCustomer =
34      customerService.getCustomer(theId);
35    theModel.addAttribute("customer", theCustomer);
36    return "customer-form";
37  }
38
39  @GetMapping("/delete")
```

```
37     public String deleteCustomer(@RequestParam("customerId")  
38         int theId) {  
39         customerService.deleteCustomer(theId);  
40         return "redirect:/customer/list";  
41     }
```