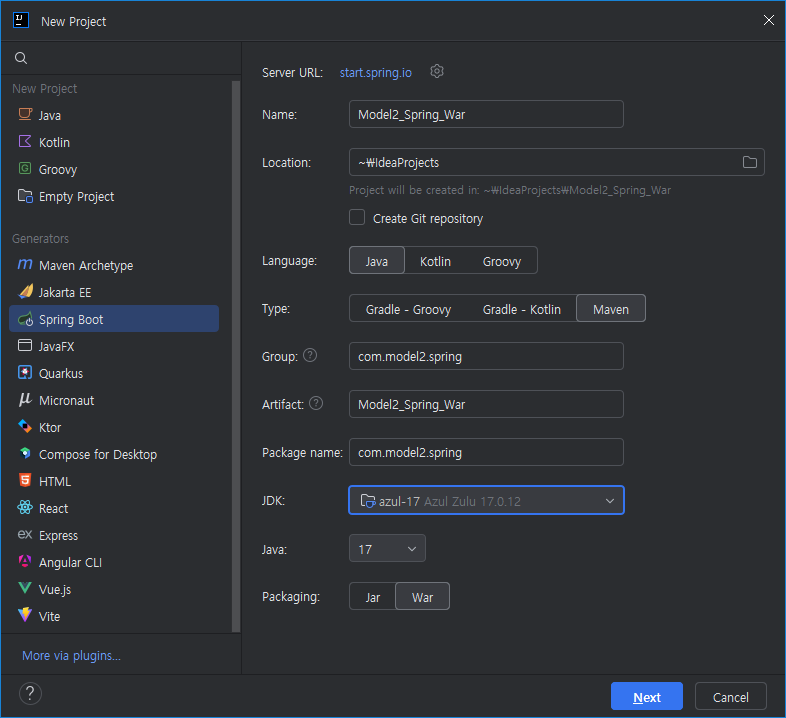
공유한소스코드>https://old-developer.tistory.com/182

https://old-developer.tistory.com/183



SpringBoot:3.3.3

Dependencies

SpringWeb

OracleTutorials(LiveSQL)|길라잡이

IT늦공김부장

https://namji9507.tistory.com/entry/Intellij-GitHub-인텔리제이-깃허브-연동

참고해서

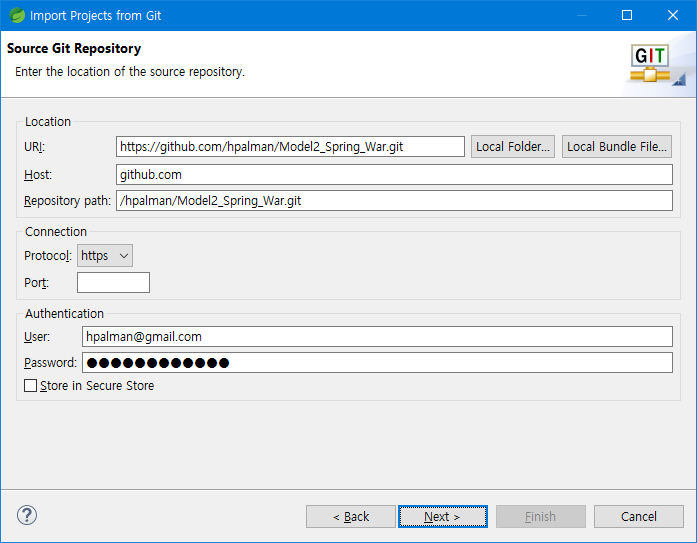
https://github.com/hpalman/Model2\_Spring\_War

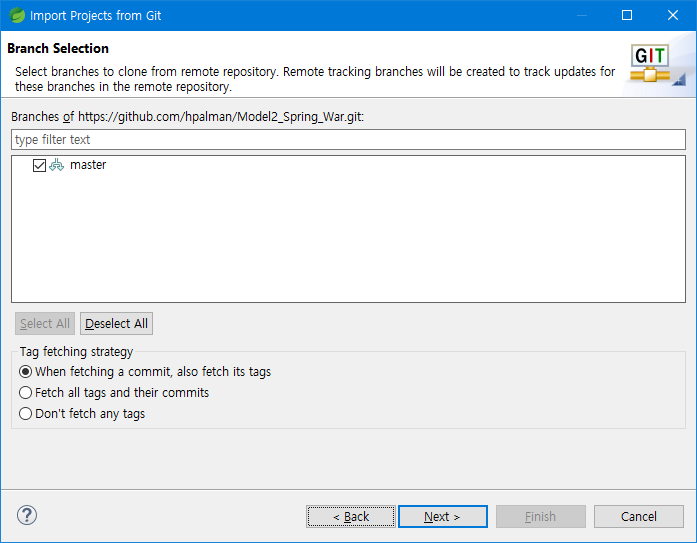
에

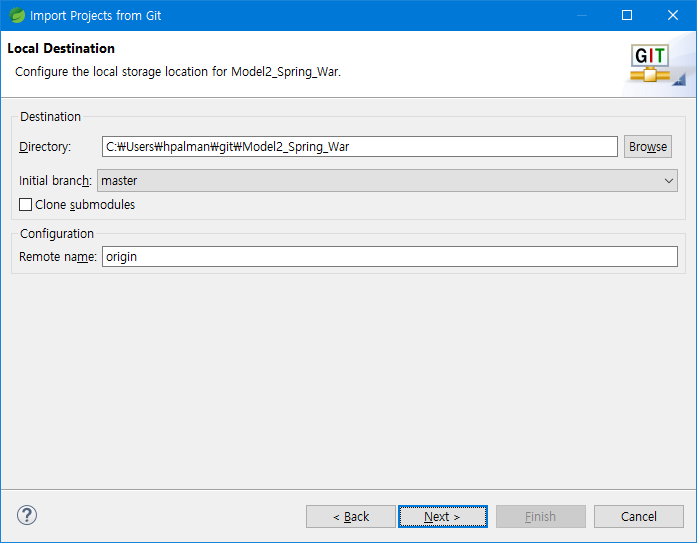
커밋과

푸쉬하였음

STS

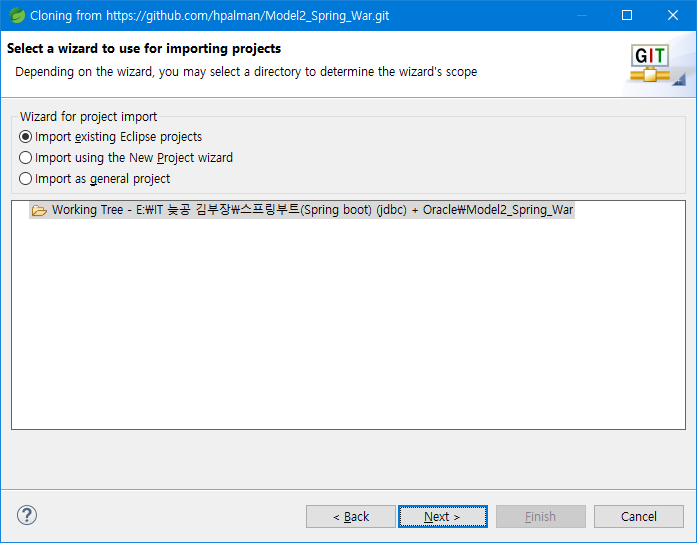






E:\IT늦공김부장\스프링부트(Springboot)(jdbc)+Oracle

==>E:\IT늦공김부장\스프링부트(Springboot)(jdbc)+Oracle\Model2\_Spring\_War

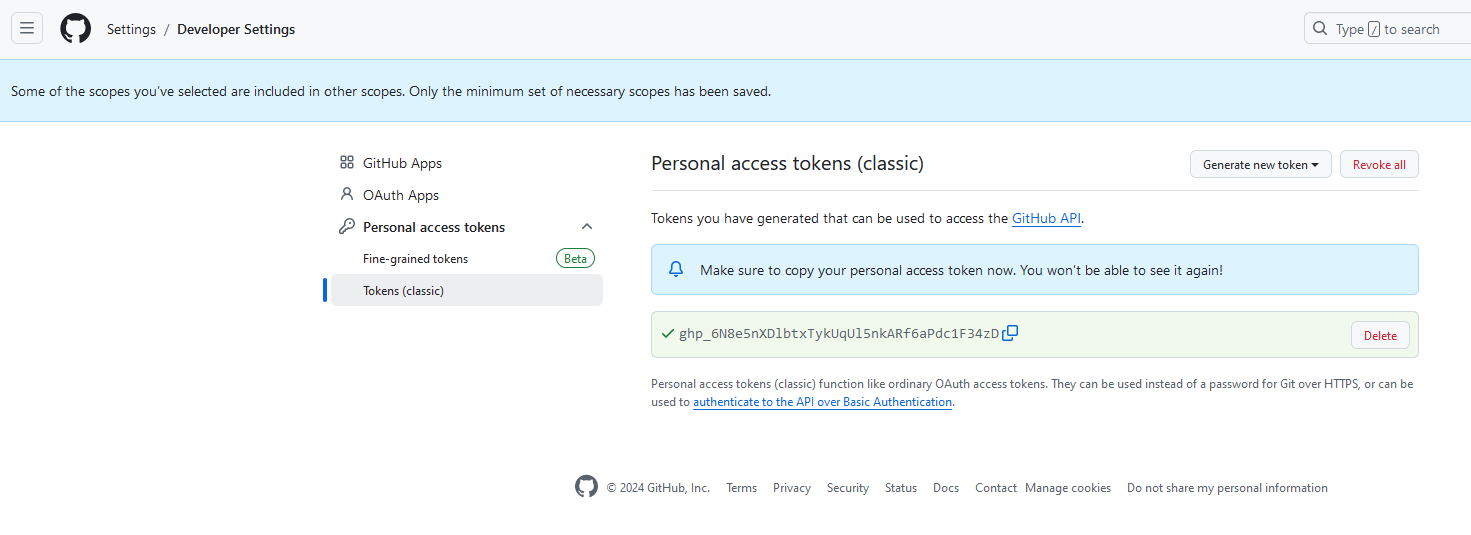


2021년8월13일부터git작업을인증할때계정암호를더이상승인하지않는다고한다...

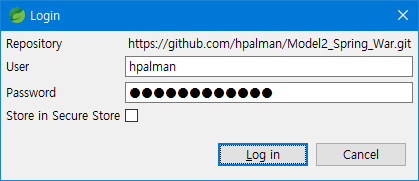
github에서토큰생성함

~~ghp\_nvn5kcSigEyNbQiWujFt5aLcvJo61P2YR4Rl~~

ghp\_6N8e5nXDlbtxTykUqUl5nkARf6aPdc1F34zD



참고)https://velog.io/@godls9638/eclipse-github-token



# STS에lombok적용하여outline확인하기

https://hello-world.tistory.com/8

# [Spring]SpringBoot시작하기(7)-MyBatis에서의DataBase다중연결

출처:https://dev-overload.tistory.com/30[E:overload:티스토리]

firstsecondthirdfourthfifthsixthseventheighthninthtentheleventhtwelfththirteenth

Master,Slave1,Slave2,Slave3

1st,2nd

3rd,4th,5th,6th,

Fist,Second,Third,Fourth,Fifth,Sixth

Master

Primary

Secondary

영어차수단계/PrimarySecondarytertiary

세로형순서

# 영문

1 primary

2 secondary

3 tertiary

4 quaternary

5 quinary

6 senary

7 septenary

8 octonary

9 nonary

10 denary

11 -

12 duodenary

가로형순서

1 2 3 4 5

primary secondary tertiary quaternary quinary

6 7 8 9 10

senary septenary octonary nonary denary

11 12

- duodenary

출처

primary,secondary,tertiary,quaternary,quinary,senary,septenary,octonary,nonary,denary

그리고11번째에해당되는단어는존재하지않지만12번째는duodenary입니다.

참고로tertiary이후로는거의쓰이는일이없습니다.

# 커스텀구성속성에관한메타데이터를생성

우리의커스텀구성속성에관한메타데이터를생성하려면프로젝트의src/main/resources/META-INF아래에additional-spring-configuration-metadata.json이라는이름의파일을생성해야한다.

그다음에메타데이터를입력하고저장하자.  
이것은JSON형식으로된taco.orders.pageSize속성의메타데이터다.

{

"properties":[

{

"name":"taco.orders.page-size",

"type":"java.lang.String",

"description":"Setsthemaximumnumberoforderstodisplayinalist."

},

{

"name":"taco.discount.codes",

"type":"java.util.Map<String,Integer>",

"description":"Amapofdiscountcodestoadiscountpercentage."

}

]

}

## @Qualifier어노테이션사용법

https://n1tjrgns.tistory.com/163

사용할의존객체를선택할수있도록해준다.

설정에서bean의한정자값을설정한다.

@Autowired어노테이션이적용된주입대상에@Qualifier어노테이션을설정한다.

이때@Qualifier의값으로앞서설정한한정자를사용한다.

## DataSourceConfig.java

|  |
| --- |
| **package**com.example.mybatisdemo.config;  **import**javax.sql.DataSource;  **import**org.apache.ibatis.session.SqlSessionFactory;  **import**org.mybatis.spring.SqlSessionFactoryBean;  **import**org.mybatis.spring.SqlSessionTemplate;  **import**org.springframework.beans.factory.annotation.Qualifier;  **import**org.springframework.boot.context.properties.ConfigurationProperties;  **import**org.springframework.boot.jdbc.DataSourceBuilder;  **import**org.springframework.context.annotation.Bean;  **import**org.springframework.context.annotation.Configuration;  **import**org.springframework.context.annotation.Primary;  **import**org.springframework.core.io.support.PathMatchingResourcePatternResolver;  **import**org.springframework.jdbc.datasource.DataSourceTransactionManager;  @Configuration  **publicclass**DataSourceConfig{  //MySQLDataSourceConfiguration  @Bean(name="mysqlDataSource")  @ConfigurationProperties(prefix="spring.datasource.mysql")  @Primary  DataSourcemysqlDataSource(){  **return**DataSourceBuilder.*create*().build();  }  @Bean(name="mysqlSqlSessionFactory")  @Primary  SqlSessionFactorymysqlSqlSessionFactory(@Qualifier("mysqlDataSource")DataSourcedataSource)  **throws**Exception{  SqlSessionFactoryBeansessionFactory=**new**SqlSessionFactoryBean();  sessionFactory.setDataSource(dataSource);  sessionFactory.setMapperLocations(  **new**PathMatchingResourcePatternResolver().getResources("classpath:mapper/mysql/\*.xml"));  **return**sessionFactory.getObject();  }  @Bean(name="mysqlTransactionManager")  @Primary  DataSourceTransactionManagermysqlTransactionManager(@Qualifier("mysqlDataSource")DataSourcedataSource){  **returnnew**DataSourceTransactionManager(dataSource);  }  @Bean(name="mysqlSqlSessionTemplate")  @Primary  SqlSessionTemplatemysqlSqlSessionTemplate(@Qualifier("mysqlSqlSessionFactory")SqlSessionFactorysqlSessionFactory){  **returnnew**SqlSessionTemplate(sqlSessionFactory);  }  //PostgreSQLDataSourceConfiguration  @Bean(name="postgresDataSource")  @ConfigurationProperties(prefix="spring.datasource.postgres")  DataSourcepostgresDataSource(){  **return**DataSourceBuilder.*create*().build();  }  @Bean(name="postgresSqlSessionFactory")  SqlSessionFactorypostgresSqlSessionFactory(@Qualifier("postgresDataSource")DataSourcedataSource)  **throws**Exception{  SqlSessionFactoryBeansessionFactory=**new**SqlSessionFactoryBean();  sessionFactory.setDataSource(dataSource);  sessionFactory.setMapperLocations(  **new**PathMatchingResourcePatternResolver().getResources("classpath:mapper/postgres/\*.xml"));  **return**sessionFactory.getObject();  }  @Bean(name="postgresTransactionManager")  DataSourceTransactionManagerpostgresTransactionManager(@Qualifier("postgresDataSource")DataSourcedataSource){  **returnnew**DataSourceTransactionManager(dataSource);  }  @Bean(name="postgresSqlSessionTemplate")  SqlSessionTemplatepostgresSqlSessionTemplate(@Qualifier("postgresSqlSessionFactory")SqlSessionFactorysqlSessionFactory){  **returnnew**SqlSessionTemplate(sqlSessionFactory);  }  } |

# Springboot기동시javax.management.InstanceNotFoundException오류조치

출처:https://oingdaddy.tistory.com/358[SISupplyDepot:티스토리]

[logback-spring.xml파일을application.yml파일에옮기는작업](https://oingdaddy.tistory.com/317)을일전에진행했었다.다전환을하고springbootapp을기동하니다음과같은오류가발생하였다.

[13:23:10.131][DEBUG][sun.rmi.server.call.log:line229]-RMITCPConnection(2)-127.0.0.1:[127.0.0.1]exception:

javax.management.InstanceNotFoundException:org.springframework.boot:type=Admin,name=SpringApplication

atjava.management/com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.getMBean(DefaultMBeanServerInterceptor.java:1083)

atjava.management/com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.getAttribute(DefaultMBeanServerInterceptor.java:637)

atjava.management/com.sun.jmx.mbeanserver.JmxMBeanServer.getAttribute(JmxMBeanServer.java:678)

**InstanceNotFoundException**??

이건또처음보는exception이다.DEBUGlevel이라그런지이렇게오류로그가많이떨어져도application의동작은정상적으로된다.하지만로그에이렇게이해할수없는로그들이막찍히는건아무래도찜찜하다.

logback설정을application.yml파일에넣고부터발생을한거니까분명이설정에이유가있다.

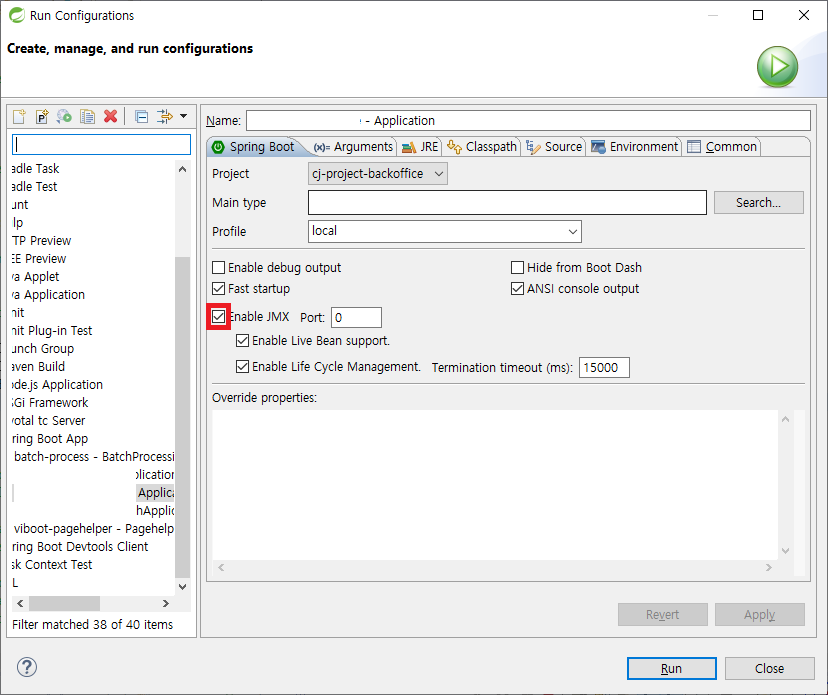
문제가발생했을때의application.yml파일을보자.

|  |  |
| --- | --- |
|  | logging: |
|  | pattern: |
|  | console:"[%d{HH:mm:ss.SSS}][%-5level][%logger.%method:line%line]-%msg%n" |
|  | level: |
|  | org:INFO |
|  | com:INFO |
|  | root:DEBUG |

결론부터말하자면root를DEBUGlevel로설정한것이문제였다.위와같은로그를보지않기위해서는**root의level을INFO이상으로올려주거나root설정을삭제**하면된다.

다른방법으로는애초에이런걸수행하지않게IDE의설정을바꿔주면된다.이클립스를사용하고있다면다음과같이설정을바꿔주자.

**BootDashboard>프로젝트우클릭>OpenConfig**로들어가면



springbootrunconfiguration

springboot의runconfiguration을볼수있고여기에서빨간색네모를친EnableJMX부분의체크를해제해준다.

그리고다시기동을하면오류를보지않을수있다.

끝!

출처:<https://oingdaddy.tistory.com/358>[SISupplyDepot:티스토리]

# 이상증상

## Class not found so assuming code is running on a pre-Java 19 JVM

2024-09-03T17:53:21.540+09:00[0;39m [32mDEBUG[0;39m [35m11476[0;39m [2m---[0;39m [2m[spring-boot-mybatis-multi-schema] [ main][0;39m [2m[0;39m[36mo.apache.tomcat.util.compat.Jre19Compat [0;39m [2m:[0;39m Class not found so assuming code is running on a pre-Java 19 JVM