例子准备

1. maven
2. jdk1.8
3. eclipse

用spring boot 进行mybatis-generator和tk-mybatis整合和测试。

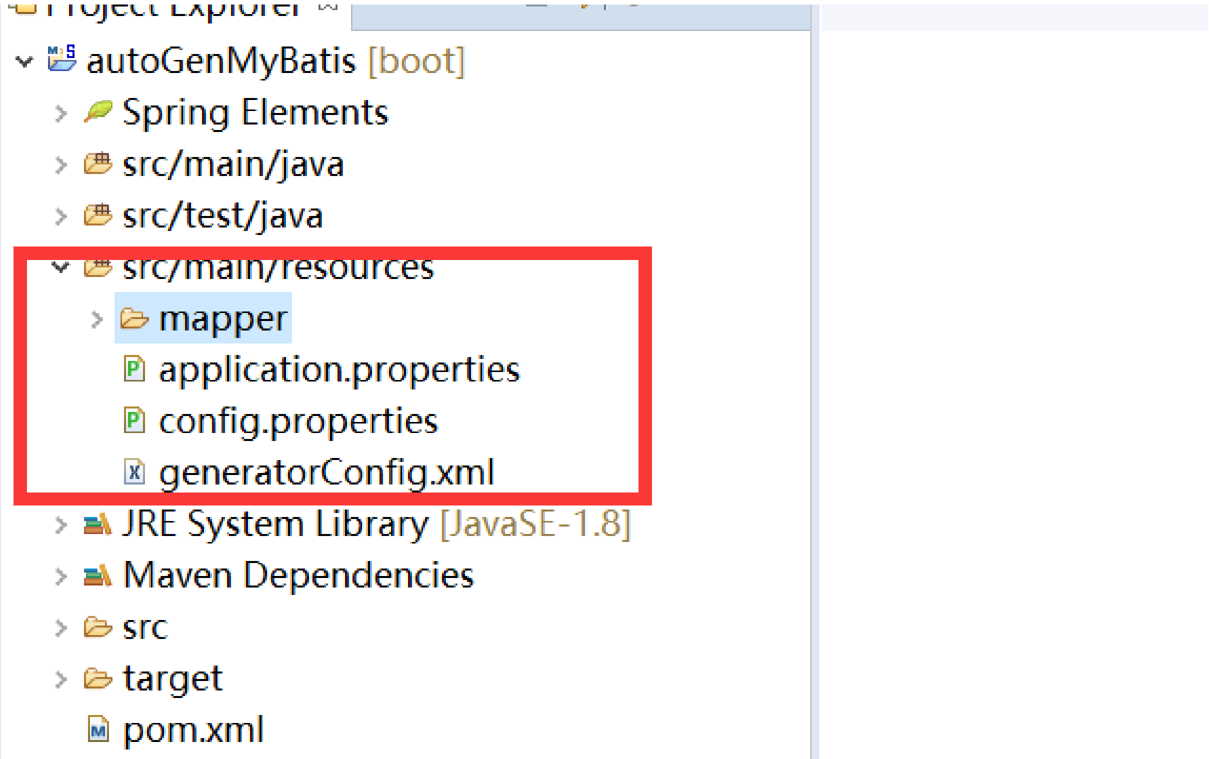
1. 创建一个maven 项目 pom 如下

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>autoGenMyBatis</groupId>        <!-- 自定义 -->
    <artifactId>autoGenMyBatis</artifactId>    <!-- 自定义 -->
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>
    <name>autoGenMyBatis</name>                <!-- 自定义 -->
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.5.6.RELEASE</version>
        <relativePath />
    </parent>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.mybatis.spring.boot</groupId>
            <artifactId>mybatis-spring-boot-starter</artifactId>
            <!-- 用spring boot 为了测试 -->
            <version>1.3.1</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>tk.mybatis</groupId>
            <artifactId>mapper-spring-boot-starter</artifactId>
            <version>1.1.1</version>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
            <plugin>
                <groupId>org.mybatis.generator</groupId>
                <artifactId>mybatis-generator-maven-plugin</artifactId>
                <version>1.3.2</version>
                <configuration>
                    <configurationFile>${basedir}/src/main/resources/generatorConfig.xml</configurationFile>
                    <overwrite>true</overwrite>
                    <verbose>true</verbose>
                </configuration>
                <dependencies>
                    <dependency>
```

```
                        <groupId>mysql</groupId>
                        <artifactId>mysql-connector-java</artifactId>
                        <version>5.1.43</version>
                    </dependency>
                    <dependency>
                        <groupId>tk.mybatis</groupId>
                        <artifactId>mapper</artifactId>
                        <version>3.4.0</version>
                    </dependency>
                </dependencies>
            </plugin>
        </plugins>
    </build>
</project>
```

2. 在resources添加如下文件

- mapper 文件夹 存放mybatis xml 配置文件
- application.propertis 配置spring boot 的一些参数
- config.propertis 配置tk.mybatis 自动生成代码的一些参数
- generatorConfig.xml 配置 mybatis generator 生成代码的参数



```
Project Explorer
autoGenMyBatis [boot]
  Spring Elements
  src/main/java
  src/test/java
  src/main/resources
    mapper
    application.properties
    config.properties
    generatorConfig.xml
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
  target
  pom.xml
```

application.propertis 测试需要

```
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/tabler
spring.datasource.username=root
spring.datasource.password=000052
mybatis.mapper-locations=classpath:mapper/*.xml
```

config.propertis

```
#生成代码的数据库链接
jdbc.driverClass = com.mysql.jdbc.Driver
jdbc.url = jdbc:mysql://localhost:3306/tabler
jdbc.user = root
jdbc.password = 000052
#tk引用包
mapper.plugin = tk.mybatis.mapper.generator.MapperPlugin
mapper.mapper = tk.mybatis.mapper.common.Mapper
#java代码路径
generator.targetJavaProject = src/main/java
#资源文件路径
generator.targetXMLProject = src/main/resources
#实体类路径
generator.javaModelGenerator = autoGenMyBatis.domain
#mybatis xml配置路径
generator.sqlMapGenerator = mapper
#mybatis接口类
generator.javaClientGenerator = autoGenMyBatis.mapper
```

generatorConfig.xml

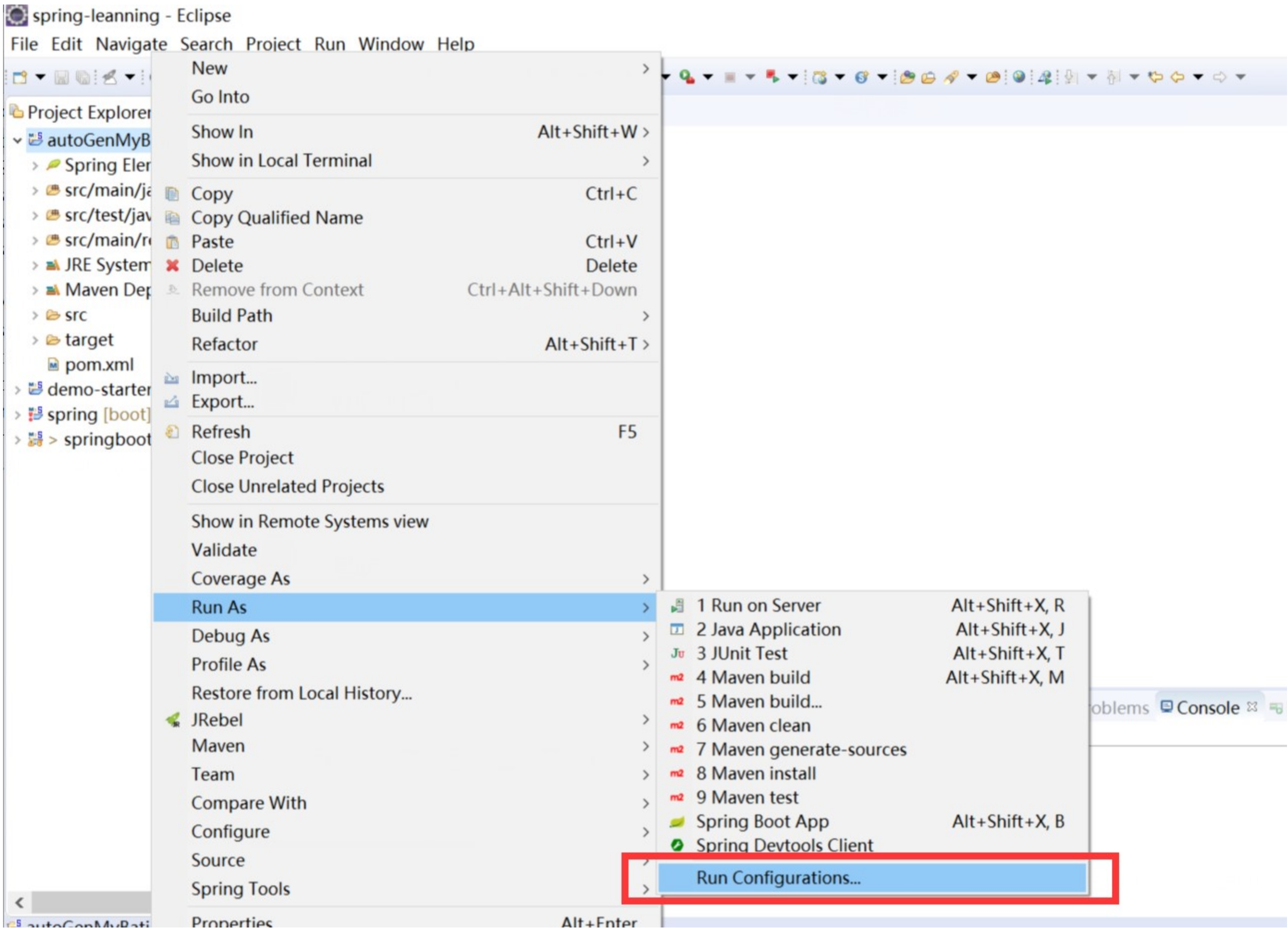
主要配置

的选项，% 生成所有表，假如只需要生成特点的表，%改成对应的表名。如果没用主键可以把generatedKey 去掉。其他详细的配置参去百度。

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE generatorConfiguration PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN" "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
<generatorConfiguration>
    <properties resource="config.properties"/>
    <context id="Mysql" targetRuntime="MyBatis3Simple" defaultModelType="flat">
        <property name="beginningDelimiter" value="`"/>
        <property name="endingDelimiter" value="`"/>
        <plugin type="${mapper.plugin}">
          <property name="mappers" value="${mapper.mapper}"/>
        </plugin>
        <jdbcConnection driverClass="${jdbc.driverClass}"
                        connectionURL="${jdbc.url}"
                        userId="${jdbc.user}"
                        password="${jdbc.password}">
        </jdbcConnection>
        <javaTypeResolver>
            <property name="forceBigDecimals" value="true"/>
        </javaTypeResolver>

        <javaModelGenerator targetPackage="${generator.javaModelGenerator}" targetProject="${generator.targetJavaProject}"/>
        <sqlMapGenerator targetPackage="${generator.sqlMapGenerator}"  targetProject="${generator.targetXMLProject}"/>
        <javaClientGenerator targetPackage="${generator.javaClientGenerator}" targetProject="${generator.targetJavaProject}" type="XMLMAPPER" />
        <table tableName="%" >
          <generatedKey column="id" sqlStatement="Mysql" identity="true"/>
        </table>
        <!--  多个表
        <table tableName="orders" >
          <generatedKey column="id" sqlStatement="Mysql" identity="true"/>
        </table>
        -->
    </context>
</generatorConfiguration>
```

启动

右键项目 run -> Run Configurations

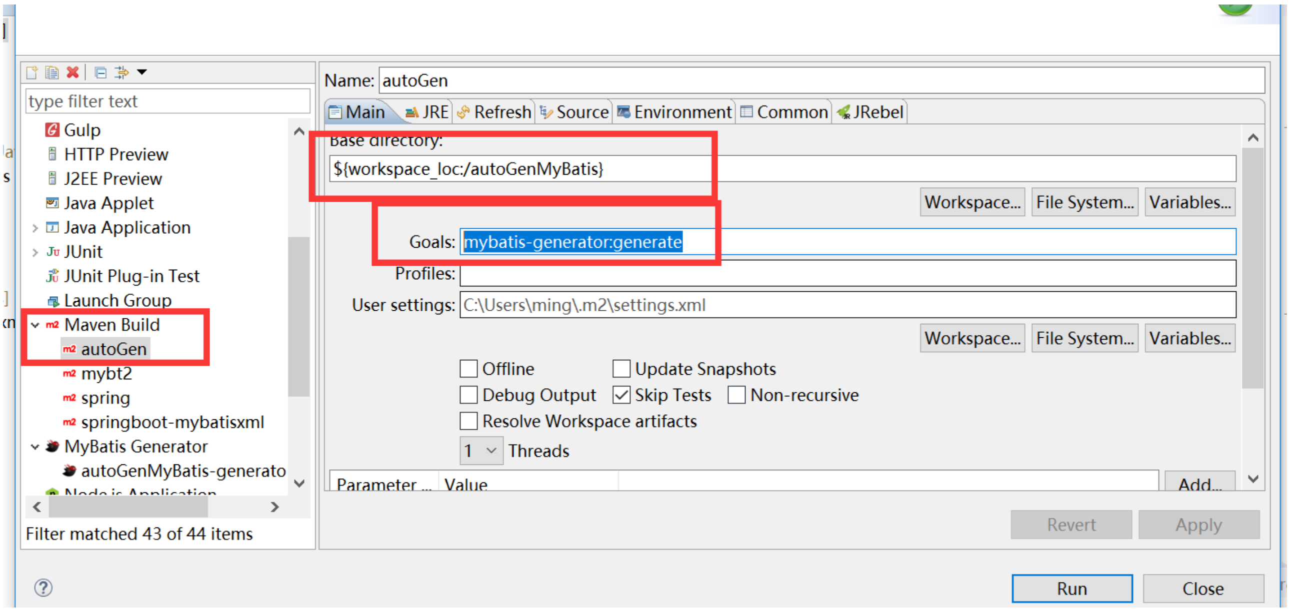


指定项目；

Goals: mybatis-generator:generate

勾选skip tests

之后运行。



Name: autoGen

Base directory: ${workspace_loc:/autoGenMyBatis}

Goals: mybatis-generator:generate

Profiles:

User settings: C:\Users\ming\.m2\settings.xml

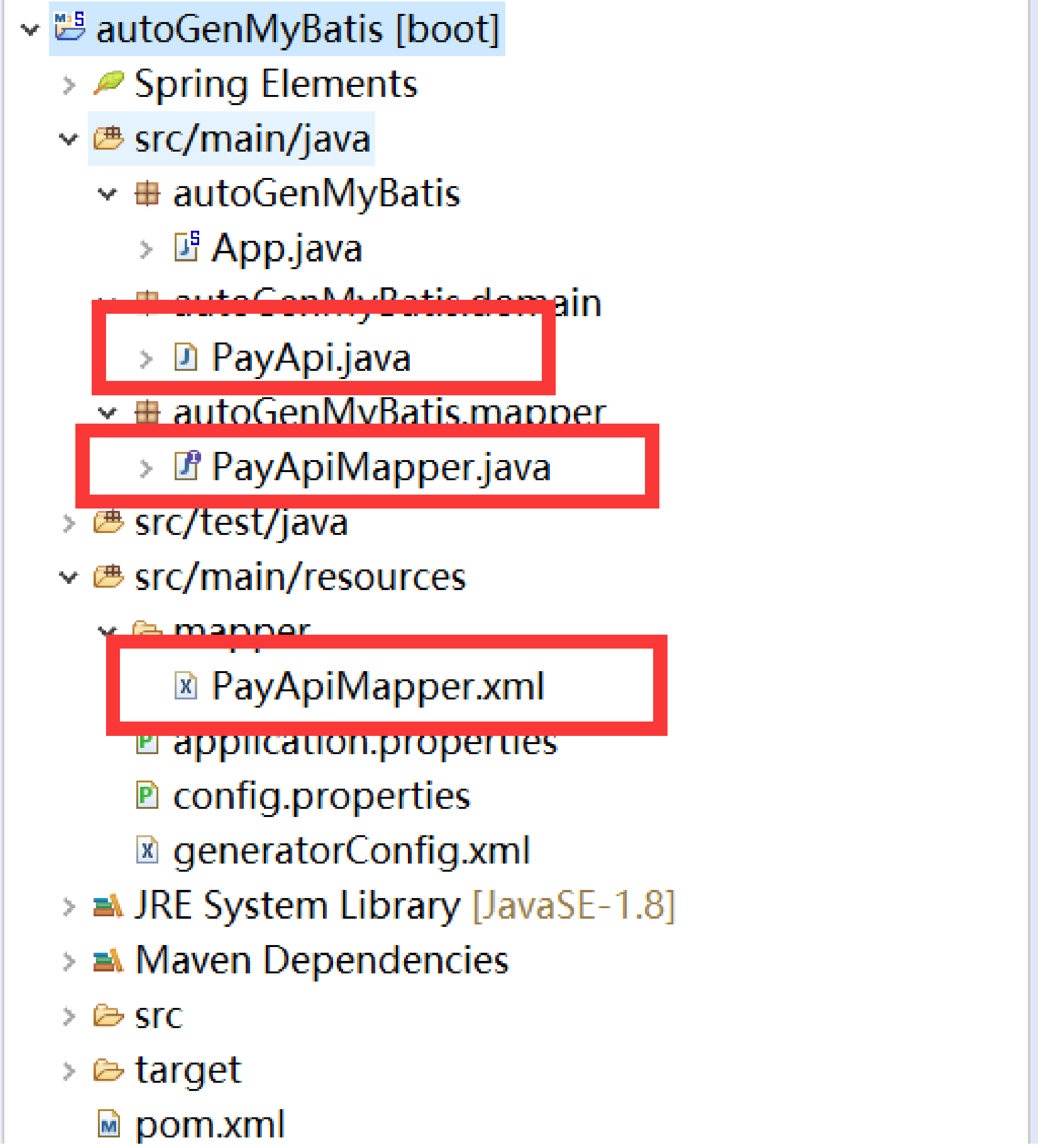
□ Offline □ Update Snapshots

□ Debug Output ☑ Skip Tests □ Non-recursive

□ Resolve Workspace artifacts

1 Threads

生成如下文件



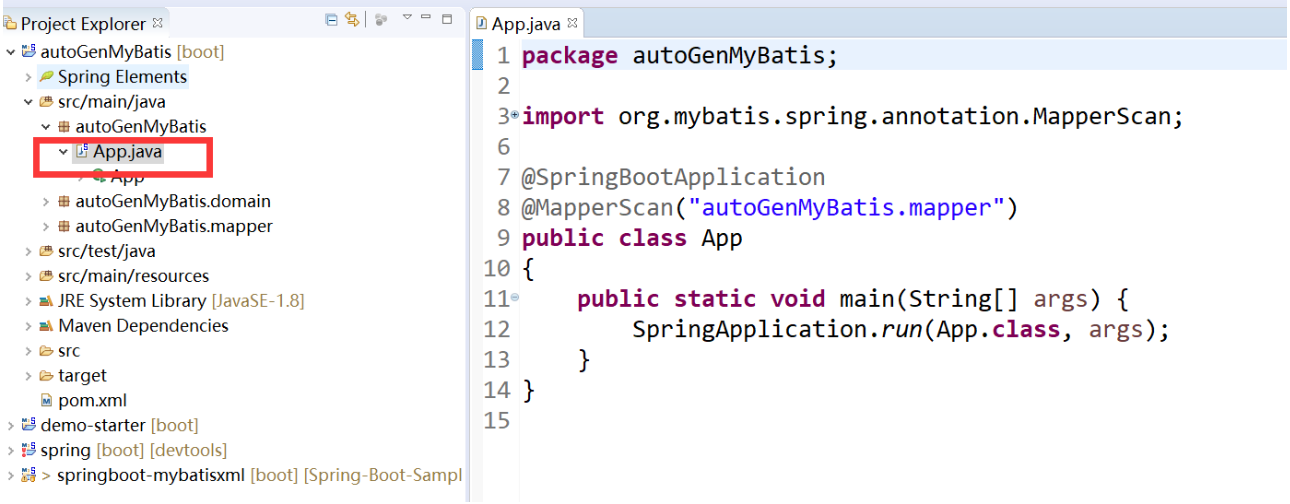
```
autoGenMyBatis [boot]
  Spring Elements
  src/main/java
    autoGenMyBatis
      App.java
    autoGenMyBatis.domain
      PayApi.java
    autoGenMyBatis.mapper
      PayApiMapper.java
  src/test/java
  src/main/resources
    mapper
      PayApiMapper.xml
    application.properties
    config.properties
    generatorConfig.xml
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
  target
  pom.xml
```

实现自己定制的方法

```
public interface PayApiMapper extends Mapper<PayApi> {
    //注解方式
    @Select("select * from pay_api where code = #{code}")
    PayApi selectByCode(@Param("code") String code);
    //xml方式
    PayApi selectByName(@Param("name") String name);
    //Example方式
    default List<PayApi> selectByTkExample() {
        Example example = new Example(PayApi.class);
        example.setOrderByClause("id desc");
        Example.Criteria criteria = example.createCriteria();
        criteria.andEqualTo("status", 1);
        return this.selectByExample(example);
    };
    //Mapper提供的一些默认实现
}
```

测试

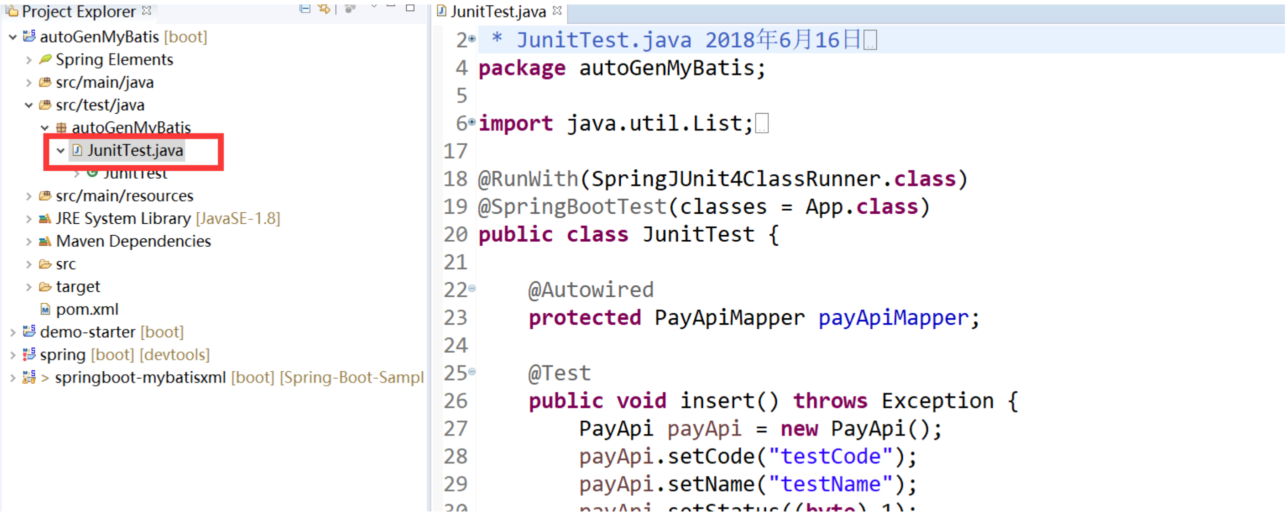
添加启动类



```
package autoGenMyBatis;

import org.mybatis.spring.annotation.MapperScan;

@SpringBootApplication
@MapperScan("autoGenMyBatis.mapper")
public class App
{
    public static void main(String[] args) {
        SpringApplication.run(App.class, args);
    }
}

```

```
@SpringBootApplication
@MapperScan("autoGenMyBatis.mapper")
public class App
{
    public static void main(String[] args) {
        SpringApplication.run(App.class, args);
    }
}
```

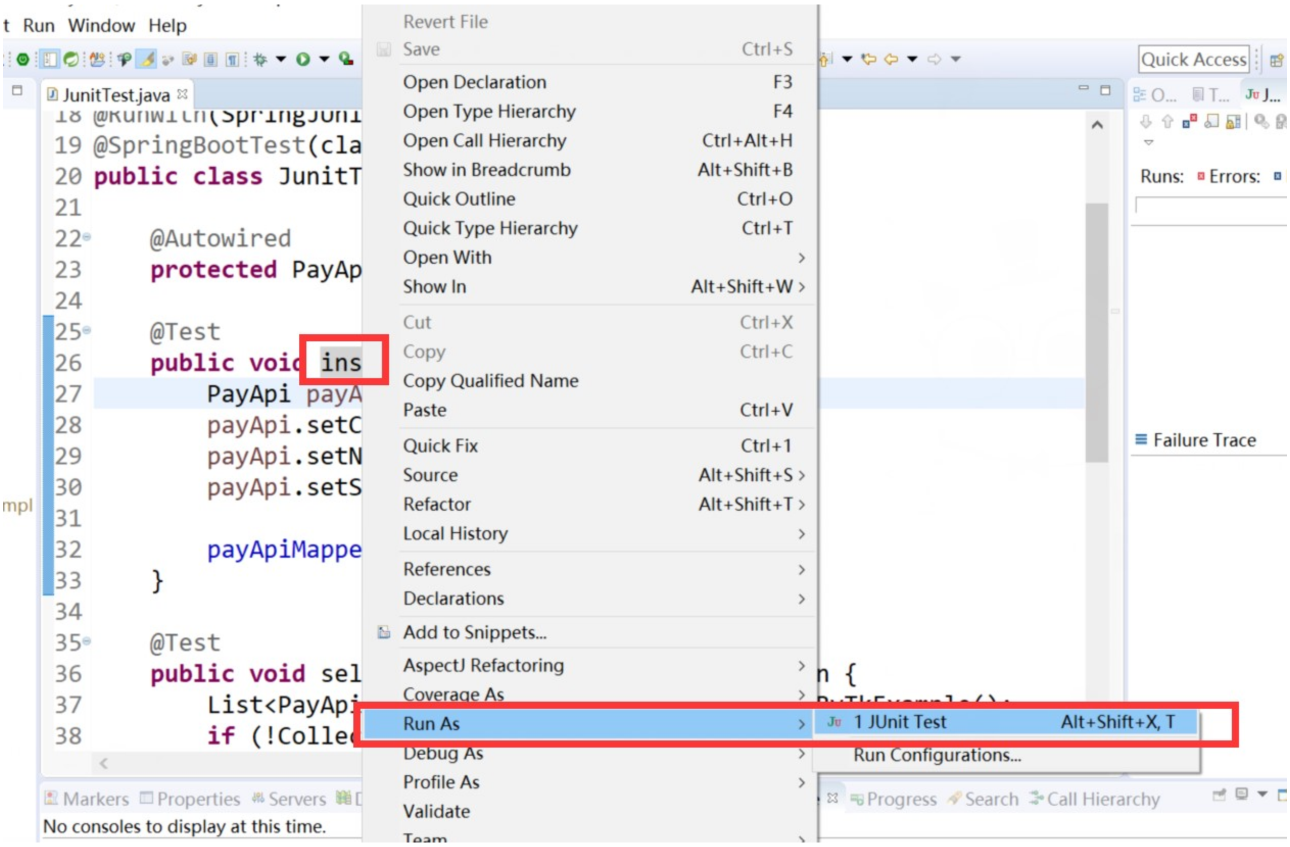
添加测试类



```
 * JunitTest.java 2018年6月16日
package autoGenMyBatis;

import java.util.List;

@RunWith(SpringJUnit4ClassRunner.class)
@SpringBootTest(classes = App.class)
public class JunitTest {

    @Autowired
    protected PayApiMapper payApiMapper;

    @Test
    public void insert() throws Exception {
        PayApi payApi = new PayApi();
        payApi.setCode("testCode");
        payApi.setName("testName");
        payApi.setStatus((byte) 1);
```

```
@RunWith(SpringJUnit4ClassRunner.class)
@SpringBootTest(classes = App.class)
public class JunitTest {
    @Autowired
    protected PayApiMapper payApiMapper;
    @Test
    public void insert() throws Exception {
        PayApi payApi = new PayApi();
        payApi.setCode("testCode");
        payApi.setName("testName");
        payApi.setStatus((byte) 1);
        payApiMapper.insert(payApi);
    }
    @Test
    public void selectByTkExample() throws Exception {
        List<PayApi> selected = payApiMapper.selectByTkExample();
        if (!CollectionUtils.isEmpty(selected)) {
            for (PayApi p : selected) {
                System.out.println(p.getId());
            }
        }
    }
    @Test
    public void selectByAnnotation() throws Exception {
        PayApi payApi = payApiMapper.selectByCode("wxpay");
        System.out.println(payApi.getId());
    }
    @Test
    public void selectByXmlMapper() throws Exception {
        PayApi payApi = payApiMapper.selectByName("微信支付");
        System.out.println(payApi.getId());
    }
}
```

对着方法名称运行即可



```
@RunWith(SpringJUnI
@SpringBootTest(cla
public class JunitT

    @Autowired
    protected PayAp

    @Test
    public void ins
        PayApi payA
        payApi.setC
        payApi.setN
        payApi.setS

        payApiMappe
    }

    @Test
    public void sel                                    n {
        List<PayApi                                    PyTkExample();
        if (!Colle
```

No consoles to display at this time.