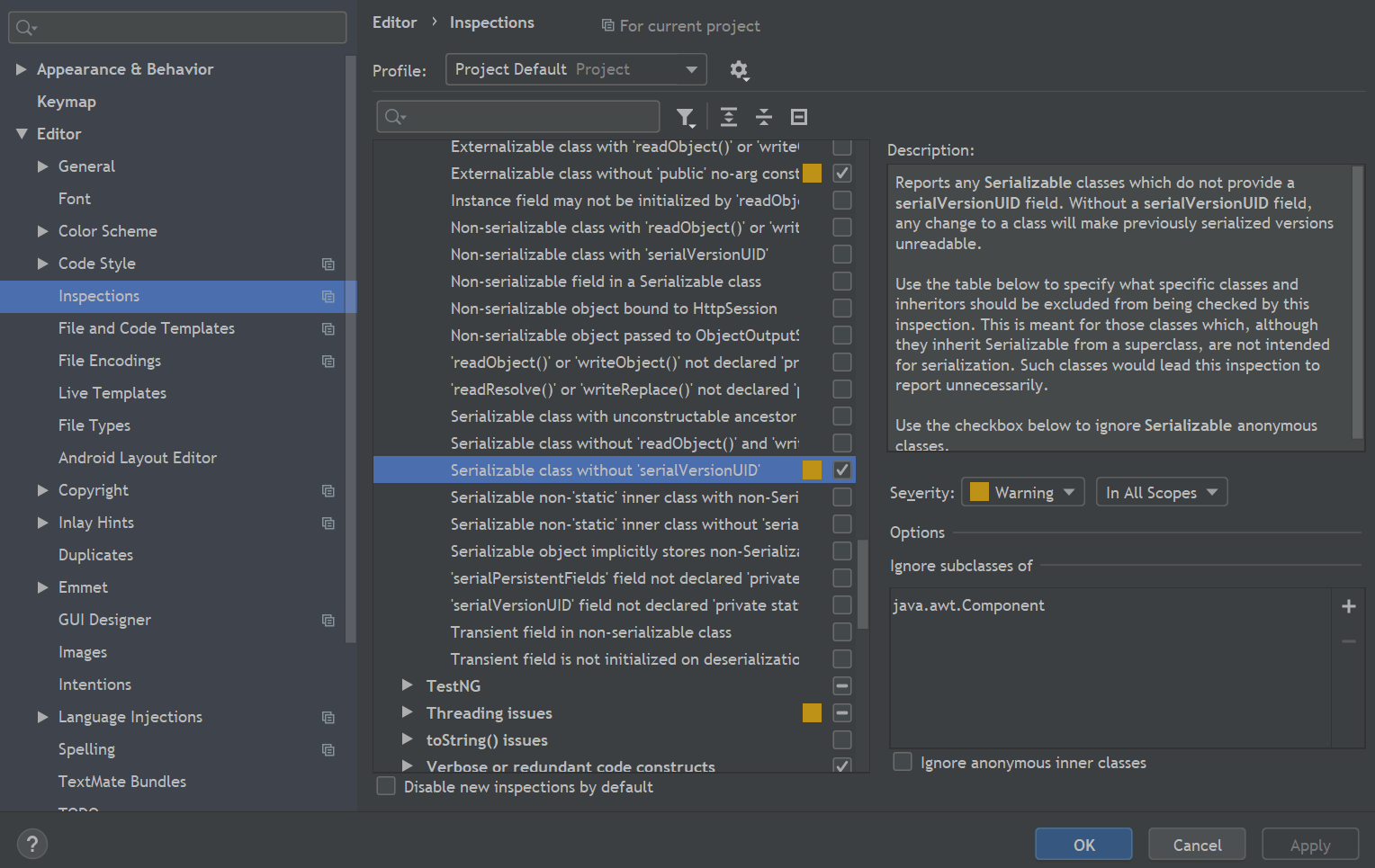
1. entity实体类自动生成serialversionid序列号（Settings）



1. Project Structure : facets设置

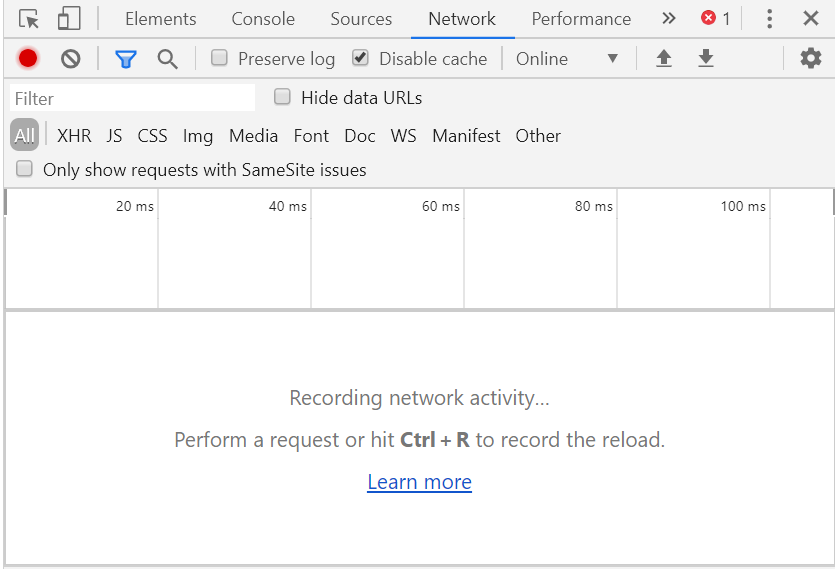
Deployment path为WEB-INF\web.xml

Web resource directory 为web root实际目录

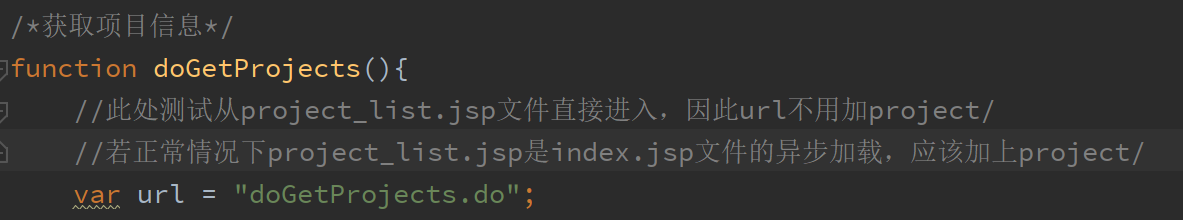
1. 设置完artifact的属性再配置tomcat（deployment）
2. MyBatis的Mapper文件一定不要放在java里！！！就算配置了也会扫描不到，老实按照规范放在resources目录下面。
3. Mapper文件的sql语句最好不要加“；”，因为数据库不只有mysql一种语法规范。
4. 现在的mysql的driver名已经改了，变为：com.mysql.cj.jdbc.Driver
5. url需要加上时区：

url=jdbc:mysql:///ssm\_travel?useUnicode=true&characterEncoding=UTF-8&allowMultiQueries=true&useSSL=true&serverTimezone=UTC

1. <beans>中的引用URL需要保证是偶数，因为两两配对。
2. 需要引用javax.Servlet依赖，不然单元测试时会报错。
3. Mybatis（Dao）的接口不用写实现类，Spring内部会生成其实现类，因此dao层不用加注解（类似于@Component，@Service）。
4. 引入jQuery文件需要放到第一个使用的<script>之前，不然会识别不了。
5. 更新了js文件但是浏览器显示的还是旧的文件，因为缓存问题，解决方法：将Disable cache选项勾中。（如下）、



1. js路径文件的路径问题：



1. HTTP status 500 servlet…

解决方法：

1. 检测tomcat是否正常启动
2. 若启动正常，则清浏览器缓存，假如启动失败检测代码问题（例数据库，路径，ClassNotFoundException）
3. BindingException:Invalid bound statement

解决方法：

1. DAO对象和Mapper文件的关联（一一对应）
2. 访问页面出现404错误：

解决方法：

1. 检测Tomcat启动是否正常
2. 检测访问路径的资源是否存在
3. 空指针问题：

解决方法：

1. 看执行日志信息，找对应问题的那行代码。
2. 看对应的注解是否加上（例如：@Service，@Autowired）
3. 通过Logger类（日志）打印出来看，例如projectService对象空指针：

Import java.util.logging.Logger;

private Logger logger = Logger.getLogger(ProjectController.class.getSimpleName());

// getSimpleName()为类的名字，getName()为包含包名的类名。

logger.info(“projectService = ” + projectService); //将日志信息打印出来