|  |
| --- |
| 要运行的Controller务必要和Application文件平级或下级，否则就会出现Whitelabel Error Page错误，表示无法访问到了。 |

|  |
| --- |
| 多配置文件还是非常有用的方法，可以针对不同的数据源进行切换。 |

|  |
| --- |
| 将配置文件做成接口传给客户端，可以让客户端去访问服务器的配置，也可以传输自己定义的配置内容 |

|  |
| --- |
| 传输配置文件中的相关配置项，若配置项的值为中文，那么极有可能在传输到浏览器上时发生乱码。原因是idea默认把配置文件设置成GBK编码，而传输的时候又采用UTF-8编码，所以编码不一致导致中文传输时是乱码。需要在项目设置中修改file encoding，统一成UTF-8即可。 |

|  |
| --- |
| 还有一种读取配置文件的方式就是采用Entity的方式将配置文件读成一个实体类，不过有错误，还不知道该怎么解决，但这个错误并不影响运行，总体来看还是采用@Value的方式更简单直接好用。 |

|  |
| --- |
| 服务端返回页面感觉用处不大，前端就应该单独开发页面，采用当前流行的开发框架，然后接收服务端发过来的JSON数据填充。通过服务端开发JSP或者其他页面并不好用。  因此，@RestController注解就非常好用，它相当于@Controller加@ResponseBody。 |

|  |
| --- |
| 上一条说的@RestController很好用，其实用处也不大，因为毕竟只是个简便写法。包括@GetMapping、@PostMapping、@PutMapping、@DeleteMapping都没什么大用处。 |

|  |
| --- |
| 不需要把返回值写成List<City>或者String，返回的列表会自动的被Spring MVC中的Jackson包转成Json。    driver-class-name不是以前的写法了，现在中间多了个cj。另外，有可能在我通过Generator生成Mapper的时候我设置了serverTimezone，导致我现在的这个url必须要加上serverTimezone字段才能编译通过。到底是不是Generator导致的还没试验过。  通过Generator.xml生成的Mapper.xml不知怎么的出现了重复的代码，以后生成之后需要检查一下是否又代码的重复。    Mybatis生成的Mapper文件需要加@Mapper注解，而不是普通DAO的@Repository。 |

|  |
| --- |
| 涉及到insert、update、delete等操作的时候务必要开启事务。Spring boot的事务开启分两步，一是在主类上写@EnableTransactionManagement开启事务；二是在具体要实施增删改的ServiceImpl类的相关方法上写@Transactional。 |

|  |
| --- |
| RESTful风格的URL我认为是非常难用的。根本无法知道各个参数到底是干什么用的，让客户端开发该如何传参？ |

|  |
| --- |
| 热部署功能非常有用！极大的提高开发效率。但是IDEA比不了Eclipse的地方就是没有Ctrl+S，还需要Build Module才能执行热部署。 |

|  |
| --- |
| 以上三图是配置Redis的步骤。 |

|  |
| --- |
|  |
| 遇到了这个问题，说明Redis没有带上配置文件，解决办法如下： |

|  |
| --- |
| 解决配置文件问题之后又遇到了这个问题。从字面意思上来讲应该是某种安全机制没有配置通过。问题就出在MySQL8上，需要在url中再加一个字段： |

|  |
| --- |
| Redis数据杂乱无章，无法阅读，不方便测试。需要设置一个序列化器来帮助我们看到结果。    把以上代码放到ServiceImpl中，运行结果如下：    至少Key是能看了。Value还是一堆乱码。 |

|  |
| --- |
| Synchronized控制进入数据库的只能是一个请求，防止缓存穿透。 |

|  |
| --- |
| 拦截器是非常重要的功能，能帮助我们处理一些分散的业务逻辑，例如登录，我们不需要在每个Controller中编写是否登录的工作，只需要通过拦截器去拦截就可以了。 |

|  |
| --- |
| 之前设置过IDEA项目的file encoding，并没有设置工程内部的http.encoding，但无论是浏览器还是服务器互传中文都不会出现乱码。以后如果遇到乱码的话，首先考虑是否在项目内设置了http.encoding，然后考虑编译器是否设置了file encoding。然后再考虑tomcat、linux等外部环境的问题。 |

|  |
| --- |
| 部署项目分为war和jar，war直接放到Tomcat的webapps目录下，启动Tomcat即可。jar包则运行java -jar XXX.jar即可，运行的是Spring Boot内嵌的Tomcat。打war包和打jar包没感觉有太多区别，直接运行maven install即可。  整个部署过程就相当于远程操作一台电脑，只要通过命令把程序运行起来即可，非常简单。 |