纯后端演示项目文案

这是一期纯后端示例教学视频，主要是为了能够提高大家的开发效率，减少不必要的学习时间，从而腾出更多精力到重难点上去。

首先是创建项目。我们都知道，web项目其实是有一定结构的配置文件和java代码文件，所以在构建SSM项目的时候会选择maven并创建项目，其中的许多配置文件依然需要手动编辑。而SpringBoot框架本身就是为了减少配置，所以这次虽然用到了maven的依赖查找进行导包，但是构建不选择maven。我们选择Spring Initializr。不要问为什么不是er，问就是前辈的智慧。

IDEA内置了SpringBoot框架的搭建，会方便很多。eclipse用户也别急，我们可以去官网下载源代码，但是需要翻墙。这里我把整个代码给出来，你可以自行导入。IDEA就不要导入了，可能会遇到不可解决的问题。

我们接下来需要用到的就是一些现代科技非常常用的东西了。首先全部列出来：RESTful、Redis、MyBatisPlus，还有很多很细节的地方，包括设计模式、lombok插件、postman测试工具、banner文字彩蛋等等，本视频中会一一进行讲解。

首先我们不应当直接开始接触。我们首先需要将全局异常处理解决了。如果你有SSM编程经验，你一定会知道Interceptor+Handler进行全局异常处理。SpringBoot类似，但是简化了很多，直接就是Handler。下面看代码。

自定义异常类——异常信息——全局异常处理器——postman工具测试——成功

接下来我们再来一一介绍。RESTful是一套api设计思想和项目框架，以往虽然见名知义的api命名方式没有问题，但还是过于杂乱。所以RESTful就作为规范出现在了人们的视野中，统一了前后台接口的规范。框架在这里就不详细展开了，直接开始实战。

编写API——参数注解——postman工具测试——成功

然后就是数据库连接。MyBatis虽然强大，但是所有功能全部手写，多少还是有点麻烦。所以这里介绍一种全新的模式：MyBatisPlus。完全免去了SQL语言的编写，CRUD再也不用自己编写了。

准备数据库——导入包——dao——service——controller——postman工具测试——成功

数据准备完了，我们来开始Redis。首先要明确，Redis只能缓存，没有别的功能。明确了这一点后，我们再来开始编码。

导入包—redisconfig—redisfactory—redisservice—controller—postman工具测试—成功

实际上SpringBoot框架依然是使用Spring作为对象管理、SpringMVC进行事务管理，所以实际上还是和SSM一样的开发流程。来，把这段话背下来：

用户在浏览器操作产生数据后，发送到服务端。由DispatcherServlet拦截，并通过查找Handler Mapping寻找合适的HandlerAdapter，接着转发到对应的Controller。Controller调用Service的方法，数据便以参数的形式流入Service。方法结束后以返回值的形式再次返回Controller。Controller将数据封装如ModelAndView，并由视图解析器进行解析。准备发回时，由DispatcherServlet再次进行拦截，最后发到浏览器回显。