2.1java

## 2.2Springboot

Spring团队决定他们希望为开发人员提供一些相对自动化配置过程的实用程序，并加快构建和部署Spring应用程序的过程，因此他们发明了Spring Boot。Spring Boot是一个实用程序项目，旨在轻松构建基于Spring，生产就绪的应用程序和服务，而不用担心。它提供了使用最小的代码/配置开箱即用的Spring Web应用程序启动和运行的最短路径。有许多特定于Spring Boot的功能，如依赖管理，自动配置和嵌入式servlet容器。

为了加速依赖关系管理过程，Spring Boot隐式地为每种类型的Spring应用程序打包所需的兼容第三方依赖项，并使用starters将它们公开给开发人员。

Spingboot框架内部实现了自己的依赖关系描述符，在程序中使用依赖关系描述符可以直接使用Spring框架自动配置的一站式服务。如果想创建一个Spring Web应用程序，只需添加spring-boot-starter-web依赖项，默认情况下，这将拉出所有常用的库来开发Spring MVC应用程序，如spring-webmvc，jackson-json ，validation-api和Tomcat。换句话说，Spring Boot收集所有公共依赖项并在一个地方定义它们，并允许开发人员使用它们，而不是每次创建新应用程序时重新发明轮子。因此，pom.xml变得比传统Spring应用程序使用的小得多。

Springboot相比较与原始的SpringMVC框架，另一大特点就是自动配置。在Springboot配置文件中配置依赖项就会为应用程序自动配置，例如，在配置文件插入spring-boot-starter-web。Spring Boot会自动配置常见的Bean，如DispatcherServlet，ResourceHandlers，MessageSource。此外，如果使用的是spring-boot-starter-jdbc，Spring Boot会自动注册DataSource，EntityManagerFactory和TransactionManager bean，并从application.properties文件中读取连接详细信息。

默认情况下，在Springboot用用程序项目在创建时就会内置一个tomcat服务器，为开发者减少了配置服务器的操作，为开发部署节省了时间，提高了效率。开发人员无需担心设置servlet容器并在其上部署应用程序。应用程序可以使用其嵌入式服务器作为可运行的jar文件自行运行。如果需要使用单独的HTTP服务器，则只需要排除默认依赖项，Spring Boot为HTTP服务器提供单独的启动程序，以帮助使此过程尽可能简单。使用嵌入式服务器创建独立的Web应用程序不仅便于开发，而且是企业级应用程序的合法解决方案，并且在微服务领域越来越有用。能够将整个服务（例如，用户身份验证）包装在暴露API的独立且完全可部署的工件中，使得分发和部署更加快速和易于管理。

## 2.3mybatis

MyBatis数据映射器框架使关系数据库与面向对象的应用程序的使用变得更加容易。 MyBatis使用XML描述符将对象与存储过程或SQL语句耦合在一起。 简单性是MyBatis数据映射器相对于对象关系映射工具的最大优势。要使用MyBatis数据映射器，您需要依赖自己的对象，XML和SQL。 没有什么可以学到的，你还不知道。 使用MyBatis Data Mapper，可以轻松获得SQL和存储过程的全部功能。

## 2.4shiro

Apache Shiro的首要目标是易于使用和理解。

以下是Apache Shiro可以做的一些事情：

验证用户以验证其身份

为用户执行访问控制，例如：

确定是否为用户分配了某个安全角色

确定是否允许用户执行某些操作

在任何环境中使用Session API，即使没有Web容器或EJB容器也是如此。

在身份验证，访问控制或会话生命周期内对事件做出反应。

聚合用户安全数据的1个或多个数据源，并将其全部显示为单个复合用户“视图”。

启用单点登录（SSO）功能

无需登录即可为用户关联启用“记住我”服务

...

还有更多 - 所有这些都集成在一个易于使用的内聚API中。

Shiro尝试为所有应用程序环境实现这些目标 - 从最简单的命令行应用程序到最大的企业应用程序，而不会强制依赖其他第三方框架，容器或应用程序服务器。当然，该项目旨在尽可能地融入这些环境，但它可以在任何环境中开箱即用

Apache Shiro是一个具有许多功能的综合应用程序安全框架。 下图显示了Shiro关注其能量的位置，此参考手册将以类似方式组织：

Shiro针对Shiro开发团队所称的“应用程序安全的四大基石” - 身份验证，授权，会话管理和加密：

身份验证：有时称为“登录”，这是证明用户是他们所说的人的行为。

授权：访问控制的过程，即确定“谁”可以访问“什么”。

会话管理：即使在非Web或EJB应用程序中，也可以管理特定于用户的会话。

密码学：使用加密算法保持数据安全，同时仍然易于使用。

还有其他功能可以在不同的应用程序环境中支持和强化这些问题，尤其是：

Web支持：Shiro的Web支持API可帮助轻松保护Web应用程序。

缓存：缓存是Apache Shiro API中的第一层公民，可确保安全操作保持快速高效。

并发：Apache Shiro支持具有并发功能的多线程应用程序。

测试：存在测试支持以帮助您编写单元和集成测试，并确保您的代码按预期受到保护。

“运行方式”：允许用户假定其他用户的身份（如果允许）的功能，有时在管理方案中很有用。

“记住我”：记住用户在会话中的身份，因此他们只需要在必要时登录。