

JAVA 面试攻略 - FAITH 的博客 - CSDN 博客

JAVA 项目面试攻略详解

想要成为合格的 **Java** 程序员或工程师到底需要具备哪些专业技能，在面试之前到底需要准备哪些东西呢？面试时面试官想了解你的什么专业技能，以下都是一个合格 JAVA 软件工程师所要具备的。

一、专业技能

1. 熟练的使用 Java 语言进行面向对象程序设计，有良好的编程习惯，熟悉常用的 JavaAPI，包括集合框架、多线程（并发编程）、I/O（NIO）、Socket、JDBC、XML、反射等。
2. 熟悉基于 JSP 和 Servlet 的 JavaWeb 开发，对 Servlet 和 JSP 的工作原理和生命周期有深入了解，熟练的使用 JSTL 和 EL 编写无脚本动态页面，有使用监听器、过滤器等 Web 组件以及 MVC 架构模式进行 JavaWeb 项目开发经验。
3. 对 Spring 的 IoC 容器和 AOP 原理有深入了解，熟练的运用 Spring 框架管理各种 Web 组件及其依赖关系，熟练的使用 Spring 进行事务、日志、安全性等的管理，有使用 SpringMVC 作为表示层技术以及使用 Spring 提供的持久化支持进行 Web 项目开发的经验，熟悉 Spring 对其他框架的整合。
4. 熟练的使用 Hibernate、MyBatis 等 ORM 框架，熟悉 Hibernate 和 MyBatis 的核心 API，对 Hibernate 的关联映射、继承映射、组件映射、缓存机制、事务管理以及性能调优等有深入的理解。
5. 熟练的使用 HTML、CSS 和 JavaScript 进行 Web 前端开发，熟悉 jQuery 和 Bootstrap，对 Ajax 技术在 Web 项目中的应用有深入理解，有使用前端 MVC 框架（AngularJS）和 JavaScript 模板引擎（HandleBars）进行项目开发的经验。
6. 熟悉常用的关系型数据库产品（MySQL、Oracle），熟练的使用 SQL 和 PL/SQL 进行数据库编程。
7. 熟悉面向对象的设计原则，对 GoF 设计模式和企业应用架构模式有深入的了解和实际开发的相关经验，熟练的使用 UML 进行面向对象的分析和设计，有 TDD（测试驱动开发）和 DDD（领域驱动设计）的经验。
8. 熟悉 Apache、NginX、Tomcat、WildFly、Weblogic 等 Web 服务器和应用服务器的使用，熟悉多种服务器整合、集群和负载均衡的配置。
9. 熟练的使用产品原型工具 Axure，熟练的使用设计建模工具 PowerDesigner 和 Enterprise Architect，熟练的使用 Java 开发环境 Eclipse 和 IntelliJ，熟练的使用 **前端开发** 环境 WebStorm，熟练的使用软件 **版本控制** 工具 SVN 和 **Git**，熟练的使用项目构建和管理工具 Maven 和 Gradle。

说明：

上面罗列的这些东西并不是每一项你都要烂熟于心，根据企业招聘的具体要求可以做相应的有针对性的准备。但是前 6 项应该是最低要求，是作为一个 Java 软件工程师必须要具备的专业技能。

二、项目经验

1. 项目介绍

本系统是 X 委托 Y 开发的用于 Z 的系统，系统包括 A、B、C、D 等模块。系统使

用了 Java 企业级开发的开源框架 E 以及前端技术 F。表示层运用了 G 架构，使用 H 作为视图 I 作为控制器并实现了 REST 风格的请求；业务逻辑层运用了 J 模式，并通过 K 实现事务、日志和安全性等功能，通过 L 实现缓存服务；持久层使用了 M 封装 CRUD 操作，底层使用 N 实现数据存取。整个项目采用了 P 开发模型。

说明：

E 通常指 **spring**（Java 企业级开发的一站式选择）；

F 最有可能是 **jQuery** 库及其插件或者是 Bootstrap 框架，当然如果要构建单页应用（SPA）最佳的方案是前端 MVC 框架（如 **AngularJS**）和 **JavaScript** 模板引擎（如 HandleBars）；G 显然是 MVC（模型 - 视图 - 控制），最有可能的实现框架是 Spring MVC，除此之外还有 Struts 2、JSF 以及 Apache 为 JSF 提供的 MyFaces 实现；

H 是 JSP，JSP 作为 MVC 中的 V，也可使用模板引擎（如 Freemarker 和 Velocity）来生成视图还可以是各种文档或报表（如 Excel 和 PDF 等）；

I 是 Servlet 或者自定义的控制器，他们是 MVC 中的 C，当然 Spring MVC 中提供了作为前端控制器的 DispatcherServlet；

J 通常是事务脚本；

K 应该是 AOP（面向切面编程）技术；

L 目前广泛使用的有 memcached 和 **Redis**；

M 的选择方案很多，最有可能的是 **hibernate** 和 MyBatis，也可以两种技术同时运用，但通常是将增删改交给 Hibernate 来处理，而复杂的查询则由 MyBatis 完成，此外 TopLink、jOOQ 也是优秀的持久层解决方案；

N 底层的数据存取传统上是使用关系型 **数据库**，可以是 **MySQL**、**Oracle**、SQLServer、DB2 等，随着 **大数据** 时代的来临，也可以采用 NoSQL（如 **MongoDB**、MemBase、BigTable 等）和其他大数据存取方案（如 GFS、HDFS 等）；

P 是项目的开发模型，可以是瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型、RAD 模型等。

2. 项目开发流程

1、可行性分析：可行性分析报告、项目开发计划书；

2、需求分析：需求规格说明书 OOAD（用例图、时序图、活动图）；

界面原型：帮助理解需求、业务层设计时推导事务脚本；

3、设计：概要设计说明书、详细设计说明书；

* 抽取业务实体（领域对象）：类图、E-R 图（概念设计阶段）；

分层架构：确定各层的技术实现方案（具体到使用的框架、数据库服务器、应用服务器等）；

* 业务层设计：事务脚本模式（事务：用户发送一次请求就是一个事务）；

脚本：一个方法或一个函数；

* 事务脚本：把一次请求封装为一个方法或一个函数；事务脚本模式：一个事务开始于脚本的打开，终止于脚本的关闭）；

* 业务层涉及的对象有三种类型：事务脚本类（封装了业务的流程）、数据访问对象（DAO，封装了持久化操作）、数据传输对象（DTO，封装了贫血 / 贫血领域对象），三者之间的关系是事务脚本类组合（聚合）数据访问对象，这二者都依赖了数据传输对象正向工程（UML 类图生成 Java 代码）和逆向工程（Java 代码生成 UML 类图）数据库物理设计（ER 图转换成表间关系图、建库和建表、使用工具插入 **测试** 数据）；

3、编码；

4、测试：测试报告、缺陷报告；

* 单元测试：对软件中的最小可测试单元进行检查和验证，在 Java 中是对类中的方法进行测试。可以使用 JUnit 工具来实现。

- * 集成测试：集成测试也叫组装测试或联合测试。在单元测试的基础上，将所有模块按照设计要求组装成为子系统进行测试；
 - * 系统测试：将已经确认的软件、硬件、外设、网络等元素结合在一起，进行信息系统的各种组装测试和确认测试，系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地方，从而提出更加完善的方案；
 - * 验收测试：在软件产品完成了单元测试、集成测试和系统测试之后，产品发布之前所进行的 **软件测试** 活动。它是技术测试的最后一个阶段，也称为交付测试。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务；
- 5、交付和维护：用户手册、操作手册；

三、项目管理

1. 版本控制：CVS/SVN/Git
2. 自动构建：Ant/Maven/Ivy/Gradle
3. 持续集成：Hudson/Jenkins

四、系统架构

1. 负载均衡服务器：F5、A10；
2. 应用服务器： HTTP 服务器：Apache、NginX（HTTP、反向代理、邮件代理服务器）；
Servlet 容器：Tomcat、Resin
3. EJB 容器：WildFly（JBoss Application Server）、GlassFish、Weblogic、WebSphere；
4. 数据库服务器：MySQL、Eclipse、Oracle；

五、第三方工具（插件）应用

1. 图表工具：基于 jQuery 的图表插件（如 jqChart、Flot、Charted 等）、Chart.js、Highcharts 等。
2. 报表工具：Pentaho Reporting、iReport、DynamicReports 等。
3. 文档处理：POI、iText 等。
4. 工作流引擎：jBPM、OpenWFE、Snaker、SWAMP 等。
5. 作业调度：Quartz、JobServer、Oddjob 等。
6. 缓存服务：EhCache、memcached、SwarmCache 等。
7. 消息队列：Open-MQ、ZeroMQ、ActiveMQ 等。
8. 安全框架：Shiro、PicketBox 等。
9. 搜索引擎：IndexTank、Lucene、ElasticSearch 等。
10. Ajax 框架：jQuery、ExtJS、DWR 等。
11. UI 插件：EasyUI、MiniUI 等。
12. 富文本框：UEditor、CKEditor 等。

六、面试提问

1. 项目是为哪个公司开发的？项目的投入是多少？
2. 有多少人参与了项目开发？整个团队中，测试人员、开发人员、项目经理比例是多少？
3. 项目开发了多长时间？项目总的代码量有多少？你的代码量有多少？
4. 项目采用了怎样的开发模型或开发流程？项目的架构是怎样的？项目的技术选型是怎样的？
5. 你在项目中担了怎样的职责？是否经常开会或加班？项目完成后有哪些收获或是经验教训？
6. 项目中最困难的部分是什么？如何解决团队开发时遇到的各种冲突？

七、面试时可以反问面试官的问题

1. 我注意到你们使用了 X 技术，请问你们是如何解决 Y 问题的？
2. 为什么你们的产品使用了 X 技术而不是 Y 技术？据我所知，X 技术虽然有 A、B、C 等好处，但也存在 D 和 E 问题，而 Y 技术可以解决 D 和 E 问题。
3. 我对您说的 X 技术不是太熟悉，但我感觉它是一个不错的解决方案，您能多讲讲它的工作原理吗？
4. 你们团队是如何进行项目规划的？一周会有几次例会？每周的代码量大概是多少？
5. 就 X 问题我能想到的解决方案目前就只有 Y 了，请问您会怎么解决这个问题？

八、S.A.R. 法则

S.A.R.法则是指先描述问题的场景，然后解释你采取的行动，最后陈述结果。

九、面谈原则

1. 声音适中、口齿清楚、面带微笑、个人形象精神，讲礼貌。
2. 主动跟面试官聊他感兴趣的话题，让面试官满意。
3. 主动的大胆的跟面试官聊自己熟悉有经验的内容：项目、技术点等，做到扬长避短。
4. 一定要表达自己非常想得到这个岗位，自己很愿意学习不懂的技术。
5. 根据自己的实际水平提出合理的待遇要求。

十、自我评价

1. 学习能力（搞 IT 行业的人需要不断的了解新的技术、工具和方法）
2. 团队意识（相比个人英雄主义，IT 行业更倡导团队合作）
3. 抗压能力（很多 IT 企业的工作强度相对来说还是比较大的）

十、最后作者寄语

最后祝各位可爱的程序员，鼓足勇气，找到自己满意的工作。

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验。