# 一.能力

# 1.觉得自己有哪些优势 ？

首先作为应届生，我能全身心的投入到公司工作；

我觉得我在学习方面的优势就是：自学能力强，有毅力。我对一些比较新潮的技术我都非常愿意去学它，来提升自己并且为公司创造价值；

在工作方面的优势就是，做事积极性高，有上进心。能按时或者说提前完成上级所布置的任务，能积极地配合同事完成工作；

在生活方面，我为人随和，容易相处，适应能力强，能尽快的融入公司，谢谢！

**注意突出自己的差异点,与别人有什么不同！！！**

# **你最擅长什么?**

在技术方面我最擅长的就是举一反三。比如说碰到一个很平常的技术，我会想会不会还有更简单，更炫的技术来解决，哈哈，其实我是一个追求完美的人。

# 二.

# 1.项目中遇到了哪些困难，如何解决的？

做项目遇到困难当然是不可避免的，有些项目可能会遇到自己没有学到的知识，那这时候我就会花最短的时间去学会这些新技术，然后把它应用在项目中；有时候组员做项目进度慢，我也会进行协调；还有项目中遇到的报错就都是小意思了，自己用debug进行调试，实在是不会的我会问问来时同学来解决。谢谢。

# 2.项目中用到了哪些技术？

现在做的项目 我会用到三大框架SSH,SSM的技术，前端也会用的一些前端的框架 ，比如说EasyUI ，bootstrap;

# 3.对公司有什么要了解的？还有什么要问的？

我想问一下我在贵公司的实习期有多长，我多久可以转正？还有我实习期间我大概做些什么，有没有项目给我练习，如果我有不会的我可以问谁？

# 需求文档有哪些，敏捷开发的流程？



# 对mvc的了解？及执行过程。

mvc 是一种设计模式。struts框架，springMVC都基于这种框架；

Mvc中的M 表示三层中的mode层，表示业务数据和业务处理，相当于JavaBean。一个模型能为多个视图提供数据。这提高了应用程序的重用性；

V表示三层中的View层，用来显示数据，并接受用户的输入。视图不进行任何业务逻辑处理，

C 表示controller，用来业务逻辑的处理。处理后会返回一个视图；

过程如下:

1. 用户向服务器发送请求，请求被Spring 前端控制Servelt DispatcherServlet捕获；

2.  DispatcherServlet对请求URL进行解析，得到请求资源标识符（URI）。然后根据该URI，调用HandlerMapping获得该Handler，最后以HandlerExecutionChain对象的形式返回给DispatcherServlet；

 3.  DispatcherServlet 根据获得的Handler，选择一个合适的HandlerAdapter

4.  提取Request中的模型数据，填充Handler入参，**开始执行Handler（Controller)**。

5.  Handler执行完成后，向DispatcherServlet 返回一个ModelAndView对象；

6.  DispatcherServlet 根据返回的ModelAndView，选择一个适合的ViewResolver解析完之后将具体的view返回给DispatcherServlet ；

7. DispatcherServlet对拿到的视图view进行渲染（将Model的数据填充到View中，得到一个可展示的视图页面）

8. 将渲染结果返回给客户端。

# 6.对未来三年的职业规划是什么？

第一年，我会熟悉公司的业务流程，尽快的融入公司，然后提升自己技术。

第二年，我会看我在公司有没有晋升的机会，如果有的话我会努力争取。

第三年，如果我的工作发展不错，相对稳定的话，我会在公司长期发展。

# 对薪资有什么要求？

作为应届毕业生我对薪资没有太大的要求，我想我能学到更多的知识来提升自己，为公司创造价值。但我来到这个城市，我想我的薪资应该能支撑我一个月的花销，当然我是非常节约的。

# 8.讲一下你做过的项目（举例说明）。

# 9.进入公司如何展开你的工作?

我会尽快的熟悉公司的业务流程，搞好同事之间的关系，尽快融入公司。然后在实习期间尽快提升自己的技术，为公司创造价值。

# 10.你对加班有什么看法？

我不反对加班，我觉得只要不是加班到凌晨那种加班我都能接受。毕竟加班能提升自己的技术的嘛。

# 说说对Mybatis的理解？

MyBatis是ibatis的升级版，是一种orm工具，简化了数据库方面的操作，另外MyBatis消除了几乎所有的JDBC代码和参数的手工配置以及结果集的检索，myBatis使用简单的xml或者注解用于配置和原始映射，将接口和java的对象，映射成数据库中的记录；

# 11.1 说说**对struts2的理解**？

1、struts 是一个按MVC模式设计的Web层框架，其实它就是一个大大的servlet，这个servlet名为ActionServlet

**struts的优缺点：  
优点：**1.实现MVC模式，结构清晰，使开发者只关注业务逻辑的实现。  
2.有丰富的tag可以使用，struts的标签库，JSTL，EL，可以提高开发效率。  
3.通过一个配置文件，可以把握整个系统各部分之间的联系。使系统的脉络更加清晰。  
4.提供Exception处理机制。  
5.数据库连接池管理。  
6.支持I18N  
**缺点：**  
1.对事件支持不够. 在struts中，实际是一个表单Form对应一个Action类(或DispatchAction)，换一句话说：在Struts中实际是一个表单只能 对应一个事件，struts这种事件方式称为application event，application event和component event相比是一种粗粒度的事件;

**一．Struts2怎么处理一个用户请求的？**  
     1，首先用户发出一个请求，通过filterDispatcher拦截器拦截下来。   
     2， filterDispatcher会询问ActionMapper来决定这个请求是否需要调用某个Action。   
     3， 这个ActionMapper封装了请求的信息。如果ActionMapper决定需要调用某个Action，FilterDispatcher会按信息(action的名字,调用的方法名)创建一个Action的代理对象。实际上对Action的调用就是通过这个代理对象来实现的。   
     4，把请求交给这个代理，如果用户提交的请求没有指定方法，那么Struts2则默认excute方法。这个代理会创建一个调度器的实例。   
     这个调度器里面定义了一个invoke方法，这个方法就实现了拦截器的交互和执行Action里的excute方法。   
     5， 拦截器是以value栈的形式调用。   
      Struts2默认的有18个拦截器。第一个拦截器是一个异常拦截器，这样的话，我第一个拦截器把异常捕获到，那么以后就不用捕获其他拦截器出现的异常了。   
    6， 当Action执行完以后，调度器就会负责根据struts.xml中的配置找到对应的返回结果。   
     7， 然后调度器就返回所有的拦截器，通过response把结果集返回给客户。

# SVN和GIT是什么，区别是什么?

Svn 和git 都属于代码版本控制的工具

最大的区别在于：

Git是分布式的，而Svn不是分布的；

1.Git下载下来后，在OffLine状态下可以看到所有的Log,SVN不可以；

2.刚开始用时很狗血的一点，SVN必须先Update才能Commit,忘记了合并时就会出现一些错误，git还是比较少的出现这种情况。

3. Svn中的分支是一个完整的目录。且这个目录拥有完整的实际文件。如果工作成员想要开啟新的分支，那将会影响“全世界”！每个人都会拥有和你一样的分支。

而 git每个工作成员可以任意在自己的本地版本库开啟无限个分支。我可以开一个分支，做我喜欢的事。完全不需担心妨碍其他工作成员。只要我不合并及提交到主要版本库，没有一个工作成员会被影响。等到我不需要这个分支时， 我只要把它从我的本地版本库删除即可。无痛无痒。

# Spring 的注解是什么（关键字）?

@Component:标准一个普通的spring Bean类。

@Controller:标注一个控制器组件类。

@Service:标注一个业务逻辑组件类。

@Repository:标注一个DAO组件类。

# 数据库为什么要用索引？

索引会增加速度，DB在执行一条Sql语句的时候，默认的方式是根据搜索条件进行全表扫描，遇到匹配条件的就加入搜索结果集合。如果我们对某一字段增加索引，查询时就会先去索引列表中一次定位到特定值的行数，大大减少遍历匹配的行数，所以能明显增加查询的速度；索引比并不是越多越好，因为索引是需要存储空间的，而且有个致命缺点是对于update/insert/delete的每次执行，字段的索引都必须重新计算更新。

# 数据库（建表，查询，排序，事务逻辑和存储过程）

# json和xml的区别?

XML是标准通用标记语言 (SGML) 的子集，非常适合 Web 传输。

XML使用DTD(document type definition)文档类型定义来组织数据;格式统一，具有跨平台和跨语言的特性；

JSON(**[JavaScript](http://lib.csdn.net/base/javascript" \o "JavaScript知识库" \t "http://blog.csdn.net/wangpanbaoding/article/details/_blank)** Object Notation)一种轻量级的数据交换格式，JSON采用兼容性很高的、完全独立于语言文本格式，可在不同平台之间进行数据交换。

# aop原理是什么?

面向切面编程，把散落在程序中的公共部分提取出来，做成切面类，这样的好处在于，代码的可重用，一旦涉及到该功能的需求发生变化，只要修改该代码就行，否则，你要到处修改，如果只要修改1、2处那还可以接受，万一有1000处呢。

AOP底层的东西就是**JDK动态代理和CGLIB代理**，说白了就是增强类的功能。

最常用的AOP应用在数据库连接以及事务处理上。

# 18.Spring的优点？

## spring 的优点？ 1.降低了组件之间的耦合性 ，实现了软件各层之间的解耦  2.可以使用容易提供的众多服务，如事务管理，消息服务等  3.容器提供单例模式支持  4.容器提供了AOP技术，利用它很容易实现如权限拦截，运行期监控等功能  5.容器提供了众多的辅助类，能加快应用的开发  6.spring对于主流的应用框架提供了集成支持，如hibernate，JPA，Struts等  7.spring属于低侵入式设计，代码的污染极低  8.独立于各种应用服务器  9.spring的DI机制降低了业务对象替换的复杂性  10.Spring的高度开放性，并不强制应用完全依赖于Spring，开发者可以自由选择spring的部分或全部

## **19.面向切面编程在spring中主要表现为两个方面** 1.面向切面编程提供声明式事务管理  2.spring支持用户自定义的切面

# 线程死锁的原因

所谓死锁，是指多个进程循环等待它方占有的资源而无限期地僵持下去的局面。很显然，如果没有外力的作用，那麽死锁涉及到的各个进程都将永远处于封锁状态。从上面的例子可以看出，计算机系统产生死锁的根本原因就是资源有限且操作不当。

1. 产生死锁的必要条件

〈1〉互斥条件。即某个资源在一段时间内只能由一个进程占有，不能同时被两个或两个以上的进程占有。这种独占资源如CD-ROM驱动器，打印机等等，必须在占有该资源的进程主动释放它之后，其它进程才能占有该资源。这是由资源本身的属性所决定的。如独木桥就是一种独占资源，两方的人不能同时过桥。

    〈2〉不可抢占条件。进程所获得的资源在未使用完毕之前，资源申请者不能强行地从资源占有者手中夺取资源，而只能由该资源的占有者进程自行释放。如过独木桥的人不能强迫对方后退，也不能非法地将对方推下桥，必须是桥上的人自己过桥后空出桥面（即主动释放占有资源），对方的人才能过桥。

    〈3〉占有且申请条件。进程至少已经占有一个资源，但又申请新的资源；由于该资源已被另外进程占有，此时该进程阻塞；但是，它在等待新资源之时，仍继续占用已占有的资源。还以过独木桥为例，甲乙两人在桥上相遇。甲走过一段桥面（即占有了一些资源），还需要走其余的桥面（申请新的资源），但那部分桥面被乙占有（乙走过一段桥面）。甲过不去，前进不能，又不后退；乙也处于同样的状况。

    〈4〉循环等待条件。存在一个进程等待序列{P1，P2，...，Pn}，其中P1等待P2所占有的某一资源，P2等待P3所占有的某一源，......，而Pn等待P1所占有的的某一资源，形成一个进程循环等待环。就像前面的过独木桥问题，甲等待乙占有的桥面，而乙又等待甲占有的桥面，从而彼此循环等待。

  上面我们提到的这四个条件在死锁时会同时发生。也就是说，只要有一个必要条件不满足，则死锁就可以排除。