前言：年底了，骚年们有没有很鸡冻——年终奖有盼头了。之前在介绍事件的这篇 [C#基础系列——再也不用担心面试官问我“事件”了](http://www.cnblogs.com/landeanfen/p/4721525.html) 也聊到了面试，引起了很多园友的共鸣。在不久的年后，热门的跳槽季就要来了，如果你准备换工作，本篇或多或少能帮到你；如果你不打算换，没关系，看看也无妨，看看是否有那么几道题能够引起你的一点点感触；又或者你是面试官，老实交代，是否问过其中的某些~~不管你是否准备面试，用一句自嘲的话来说：工作经验可以没有，但换工作的经验咱可不能缺~~

对于一些基础的笔试和简单的面试题，博主就不照本宣科了，博客园一搜一大把：[http://zzk.cnblogs.com/s?t=b&w=.net%20%E9%9D%A2%E8%AF%95%E9%A2%98](http://zzk.cnblogs.com/s?t=b&w=.net%20%E9%9D%A2%E8%AF%95%E9%A2%98" \t "_blank)。本篇就结合博主遇到以及网上经常看到的一些面试题做一个总结。PS：由于博主用的主要编程语言是C#，所以本篇主要介绍关于.net面试中的常见问题。

**一、对于 Web 性能优化，您有哪些了解和经验吗？**

**出现指数：五颗星**

主要考点：这道题是博主在博客园的新闻里面看到的，回想之前几年的面试经历，发现此题出现概率还是比较高的。因为它的考面灰常广，可以让面试官很快了解你的技术涉及面以及这些技术面的深度。

参考答案：这个问题可以分前端和后端来说。

1、前端优化

（1）减少 HTTP 请求的次数。我们知道每次发送http请求，建立连接和等待相应会花去相当一部分时间，所以在发送http请求的时候，尽量减少请求的次数，一次请求能取出的数据就不要分多次发送。

（2）启用浏览器缓存，当确定请求的数据不会发生变化时，能够直接读浏览器缓存的就不要向服务端发送请求。比如我们ajax里面有一个参数能够设置请求的时候是否启用缓存，这种情况下就需要我们在发送请求的时候做好相应的缓存处理。

（3）css文件放在<head>里面，js文件尽量放在页面的底部。因为请求js文件是很花费时间，如果放在<head>里面，就会导致页面的DOM树呈现需要等待js文件加载完成。这也就是为什么很多网站的源码里面看到引用的文件放在最后的原因。

（4）使用压缩的css和js文件。这个不用多说，网络流量小。

（5）如果条件允许，尽量使用CDN的方式引用文件，这样就能减少网络流量。比如我们常用的网站<http://www.bootcdn.cn/>。

（6）在写js和css的语法时，尽量避免重复的css，尽量减少js里面循环的次数，诸如此类。

2、后端优化：

（1）程序的优化：这是一个很大的话题，我这里就选几个常见的。比如减少代码的层级结构、避免循环嵌套、避免循环CURD数据库、优化算法等等。

（2）数据库的优化：（由于数据库优化不是本题重点，所以可选几个主要的来说）比如启用数据库缓存、常用的字段建索引、尽量避免大事务操作、避免select \* 的写法、尽量不用in和not in 这种耗性能的用法等等。

（3）服务器优化：（这个可作为可选项）负载均衡、Web服务器和数据库分离、UI和Service分离等等。

**二、MVC路由理解？（屡见不鲜）**

**出现指数：五颗星**

主要考点：此题主要考点是MVC路由的理解。

参考答案：

1、首先我们要理解MVC中路由的作用：url Routing的作用是将浏览器的URL请求映射到特定的MVC控制器动作。

2、当我们访问http://localhost:8080/Home/Index这个地址的时候，请求首先被UrlRoutingModule截获，截获请求后，从Routes中得到与当前请求URL相符合的RouteData对象，将RouteData对象和当前URL封装成一个RequestContext对象，然后从Requestcontext封装的RouteData中得到Controller名字，根据Controller的名字，通过反射创建控制器对象，这个时候控制器才真正被激活，最后去执行控制器里面对应的action。

**三、谈谈你觉得做的不错系统，大概介绍下用到了哪些技术？**

**出现指数：五颗星**

主要考点：这是一道非常开放的面试题。博主遇到过好几家公司的面试官都问道了这个，博主觉得他们是想通过这个问题快速了解面试者的技术水平。此题只要结合你最近项目用到的技术谈谈就好了。

参考答案：

就拿我之前做过的一个项目为例来简单说明一下吧。项目分为客户端和服务端，客户端分为BS客户端和CS客户端，BS客户端采用MVC 5.0的框架，CS客户端是Winform项目，服务端使用WebApi统一提供服务接口，考虑以后可能还要扩展手机端，所以服务接口的参数和返回值使用通用的Json格式来传递数据。

1、服务端采用的面向接口编程，我们在软件架构的过程中，层和层之间通过接口依赖，下层不是直接给上层提供实现，而是提供接口，具体的实现以依赖注入的方式在运行的时候动态注入进去。MEF就是实现依赖注入的一种组件。它的使用使得UI层不直接依赖于BLL层，而是依赖于中间的一个IBLL层，在程序运行的时候，通过MEF动态将BLL里面的实现注入到UI层里面去，这样做的好处是减少了层与层之间的耦合。服务端的异常里面、权限验证、日志记录等通用功能使用了AOP拦截的机制统一管理，项目中使用的是Postsharp这个组件，很好地将通用需求功能从不相关的类当中分离出来，提高了代码的可维护性。

2、BS的客户端采用的jquery+bootstrap 的方式，所有页面采用流式布局，能更好适应各种不同的终端设备（PC、手机）。项目中使用了各种功能强大的bootstrap组件，能适应各种复杂的业务需求。

**四、Js继承实现。**

**出现指数：五颗星**

主要考点：这道题考验面试者对js理解的深度。根据博主的经历，这种题一般在笔试出现的几率较大，为什么把它放在这里，因为它确实太常见了。其实js实现继承的方式很多，我们只要写好其中一种就好了。

参考答案：原型链继承

[复制代码](javascript:void(0);)

function Person(name, age){

this.name = name;

this.age = age;

}

//通过原型链给Person添加一个方法

Person.prototype.getInfo = function(){

console.log(this.name + " is " + this.age + " years old!");

}

function Teacher(staffId){

this.staffId = staffId;

}

//通过prototype生命Teacher继承Person

Teacher.prototype = new Person();

var will = new Teacher(1000);

will.name = "Will";

will.age = 28;

will.getInfo();

[复制代码](javascript:void(0);)

**五、谈谈你对设计模式的认识？结合你用得最多的一种设计模式说说它的使用。**

**出现指数：五颗星**

主要考点：不用多说，这题考的就是对设计模式的理解。一般为了简单可能会要求你写一个单例模式，注意最好是写一个完整点的，考虑线程安全的那种。然后会让你说说你在项目中什么情况下会用到这种模式

参考答案：

通用写法

[复制代码](javascript:void(0);)

　　 public class Singleton

{

// 定义一个静态变量来保存类的实例

private static Singleton uniqueInstance;

// 定义一个标识确保线程同步

private static readonly object locker = new object();

// 定义私有构造函数，使外界不能创建该类实例

private Singleton()

{

}

/// <summary>

/// 定义公有方法提供一个全局访问点,同时你也可以定义公有属性来提供全局访问点

/// </summary>

/// <returns></returns>

public static Singleton GetInstance()

{// 双重锁定只需要一句判断就可以了

if (uniqueInstance == null)

{

lock (locker)

{

// 如果类的实例不存在则创建，否则直接返回

if (uniqueInstance == null)

{

uniqueInstance = new Singleton();

}

}

}

return uniqueInstance;

}

}

[复制代码](javascript:void(0);)

单例模式确保一个类只有一个实例,并提供一个全局访问点，它的使用场景比如任务管理器整个系统中应该只有一个把，再比如操作文件的对象，同一时间我们只能有一个对象去操作文件吧。最重要的，比如我们项目中用得非常多的功能→日志记录，在一个线程中，记录日志的对象应该也只能有一个吧。单例模式的目的是为了保证程序的安全性和数据的唯一性。或者你也可以结合你使用的其他设计模式来说明。

**六、IIS的工作原理？**

**出现指数：四颗星**

主要考点：此题主要考的是.net framework和IIS是如何结合呈现页面的。这是一个有点复杂的过程，面试的时候不可能说得完整，那么我们就抓住几个关键点说说就可以。其实博主也不能完全理解这个过程，今天正好借这个机会温习下。

参考答案：

1、当客户端发送HTTP Request时，服务端的HTTP.sys（可以理解为IIS的一个监听组件） 拦截到这个请求；

2、HTTP.sys 联系 WAS 向配置存储中心请求配置信息。

3、然后将请求传入IIS的应用程序池。

4、检查请求的后缀，启动aspnet\_isapi.dll这个dll，这个dll是.net framework里面的，也就是说到这一步，请求进入了.net framework的管辖范围。

5、这个时候如果是WebForm，开始执行复杂的页面生命周期（HttpRuntime→ProcessRequest→HttpContext→HttpHandler）；如果是MVC，则启动mvc的路由机制，根据路由规则为URL来指定HttpHandler。

6、httpHandler处理请求后，请求结束，给出Response，客户端处理响应，整个过程结束。

**七、Http协议**

**出现指数：四颗星**

主要考点：此题主要考对于web里面http协议的理解。

参考答案：

1、http协议是浏览器和服务器双方共同遵循的规范，是一种基于TCP/IP应用层协议。

2、http是一种典型的请求/相应协议。客户端发送请求，请求的内容以及参数存放到请求报文里面，服务端收到请求后，做出相应，返回相应的结果放到相应报文里面。通过F12可以查看请求报文和相应报文。

3、http协议是”无状态”的，当客户端向服务端发送一次http请求后，服务端收到请求然后返回给客户端相应的结果，服务器会立即断开连接并释放资源。在实际开发过程中，我们有时需要“保持”这种状态，所以衍生出了Session/Cookie这些技术。

4、http请求的方式主要有get/post。

5、http状态码最好记几个，博主有一次面试就被问到了。200（请求成功）、404（请求的资源不存在）、403（禁止访问）、5xx（服务端错误）

**八、数据库优化经验（后端工程师非常常见）**

**出现指数：四颗星**

主要考点：此题考察后端工程师操作数据库的经验。说实话，数据库是博主的弱项，博主觉得对于这种考题，需要抓住几个常用并且关键的优化经验，如果说得不对，欢迎大家斧正。

参考答案：

1、数据库运维方面的优化：启用数据库缓存。对于一些比较常用的查询可以采用数据库缓存的机制，部署的时候需要注意设置好缓存依赖项，防止“过期”数据的产生。

2、数据库索引方面的优化：比如常用的字段建索引，联合查询考虑联合索引。（PS：如果你有基础，可以敞开谈谈聚集索引和非聚集索引的使用场景和区别）

3、数据库查询方面的优化：避免select \* 的写法、尽量不用in和not in 这种耗性能的用法等等。

4、数据库算法方面的优化：尽量避免大事务操作、减少循环算法，对于大数据量的操作，避免使用游标的用法等等。

**九、关于代码优化你怎么理解？你会考虑去代码重构吗？**

**出现指数：四颗星**

主要考点：此题考的是面试者对代码优化的理解，以及代码如何重构的相关知识。

参考答案：

1、对于代码优化，之前的公司每周会做代码审核，审核的主要作用就是保证代码的正确性和执行效率，比如减少代码的层级结构、避免循环嵌套、避免循环CURD数据库、尽量避免一次取出大量数据放在内存中（容易内存溢出）、优化算法等。

2、对于陈旧代码，可能很多地方有调用，并且开发和维护人员很有可能不是同一个人，所以重构时要格外小心，如果没有十足的把握，不要轻易重构。如果必须要重构，必须做好充分的单元测试和全局测试。

**十、谈谈你的优点和缺点？**

**出现指数：四颗星**

主要考点：这道题让人有一种骂人的冲动，但是没办法，偏偏很多所谓的大公司会问这个。比如华为。这个问题见仁见智，答案可以自己组织。

参考答案：

优点：对于新的技术学习能力强，能很快适应新环境等等

缺点：对技术太过于执着等等

**十一、关于服务器端 MVC 架构的技术实现，您是怎样理解的？这种架构方式有什么好处？您在项目中是如何应用这一架构的？**

**出现指数：三颗星**

主要考点：此题主要考的对于MVC这种框架的理解。

参考答案：MVC，顾名思义，Model、View、Controller。所有的界面代码放在View里面，所有涉及和界面交互以及URL路由相关的逻辑都在Controller里面，Model提供数据模型。MVC的架构方式会让系统的可维护性更高，使得每一部分更加专注自己的职责，并且MVC提供了强大的路由机制，方便了页面切换和界面交互。然后可以结合和WebForm的比较，谈谈MVC如何解决复杂的控件树生成、如何避免了复杂的页面生命周期。

**十二、网站优化：网站运行慢，如何定位问题？发现问题如何解决？**

**出现指数：三颗星**

主要考点：此题和问题一类似，考察Web的问题定位能力和优化方案。

参考答案：

浏览器F12→网络→查看http请求数以及每个请求的耗时，找到问题的根源，然后依次解决，解决方案可以参考问题一里面的Web优化方案。

**十三、说说你最擅长的技术？并说说你是如何使用的？**

**出现指数：三颗星**

主要考点：这是一道非常开放的面试题。最初遇到这种问题，博主很想来一句：你妹，这叫什么问题！但确实有面试官问到。回头想想，其实此题考查你擅长的技术的涉及深度。其实博主觉得对于这个问题，可以结合你项目中用到的某一个技术来说就好了。

参考答案：

 简单谈谈MEF在我们项目里面的使用吧。

在谈MEF之前，我们必须要先谈谈DIP、IOC、DI

依赖倒置原则（DIP）：一种软件架构设计的原则（抽象概念）

控制反转（IoC）：一种反转流、依赖和接口的方式（DIP的具体实现方式）。

依赖注入（DI）：IoC的一种实现方式，用来反转依赖（IoC的具体实现方式）。

什么意思呢？也就是说，我们在软件架构的过程中，层和层之间通过接口依赖，下层不是直接给上层提供实现，而是提供接口，具体的实现以依赖注入的方式在运行的时候动态注入进去。MEF就是实现依赖注入的一种组件。它的使用使得UI层不直接依赖于BLL层，而是依赖于中间的一个IBLL层，在程序运行的时候，通过MEF动态将BLL里面的实现注入到UI层里面去，这样做的好处是减少了层与层之间的耦合。这也正是面向接口编程方式的体现。

**十四、自己写过JS组件吗？举例说明。**

**出现指数：三颗星**

主要考点：此题考的js组件封装和js闭包的一些用法。一般来说，还是笔试出现的几率较大。

参考答案：自定义html的select组件

[复制代码](javascript:void(0);)

//combobox

(function ($) {

$.fn.combobox = function (options, param) {

if (typeof options == 'string') {

return $.fn.combobox.methods[options](this, param);

}

options = $.extend({}, $.fn.combobox.defaults, options || {});

var target = $(this);

target.attr('valuefield', options.valueField);

target.attr('textfield', options.textField);

target.empty();

var option = $('<option></option>');

option.attr('value', '');

option.text(options.placeholder);

target.append(option);

if (options.data) {

init(target, options.data);

}

else {

//var param = {};

options.onBeforeLoad.call(target, option.param);

if (!options.url) return;

$.getJSON(options.url, option.param, function (data) {

init(target, data);

});

}

function init(target, data) {

$.each(data, function (i, item) {

var option = $('<option></option>');

option.attr('value', item[options.valueField]);

option.text(item[options.textField]);

target.append(option);

});

options.onLoadSuccess.call(target);

}

target.unbind("change");

target.on("change", function (e) {

if (options.onChange)

return options.onChange(target.val());

});

}

$.fn.combobox.methods = {

getValue: function (jq) {

return jq.val();

},

setValue: function (jq, param) {

jq.val(param);

},

load: function (jq, url) {

$.getJSON(url, function (data) {

jq.empty();

var option = $('<option></option>');

option.attr('value', '');

option.text('请选择');

jq.append(option);

$.each(data, function (i, item) {

var option = $('<option></option>');

option.attr('value', item[jq.attr('valuefield')]);

option.text(item[jq.attr('textfield')]);

jq.append(option);

});

});

}

};

$.fn.combobox.defaults = {

url: null,

param: null,

data: null,

valueField: 'value',

textField: 'text',

placeholder: '请选择',

onBeforeLoad: function (param) { },

onLoadSuccess: function () { },

onChange: function (value) { }

};

})(jQuery);

[复制代码](javascript:void(0);)

调用的时候

$("#sel\_search\_orderstatus").combobox({

url: '/apiaction/Order/OrderApi/GetOrderStatu',

valueField: 'VALUE',

textField: 'NAME'

});

就能自动从后台取数据，注意valueField和textField对应要显示和实际值。

**十五、自己写过多线程组件吗？简要说明！**

**出现指数：三颗星**

主要考点：此题是两年前博主在携程的一次电话面试中遇到的，其他地方基本上没遇到过，其实到现在也不能理解当时面试官问这个问题的目的。但我想，此问题必有出处，估计面试官是想了解你对多线程以及线程池等的理解深度。