**QA问答系统改进建议**

**建议一、显示改进**

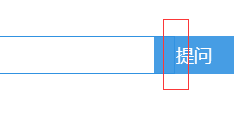
用户体验:UE

用户体验并不是指产品本身是如何工作的，而是指产品如何和外界联系并发挥作用，也就是人们如何接触或者使用它。Web中的用户体验是指用户在访问平台的过程中的全部体验，它包括平台是否有用，疑惑或者bug程度，功能是否易用、简约，界面设计，交互设计是否友好等方面。

用户界面设计：UI

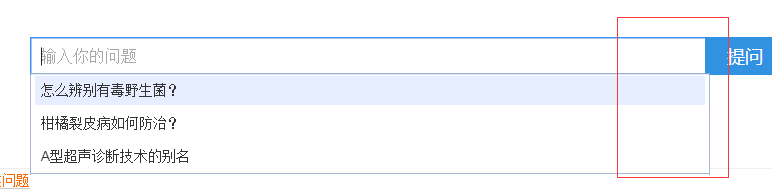
用户界面设计，在很大程度上就是在探讨如何让产品的界面更加具有可用性，如何让用户有更良好的体验。通过这种界面，用户嫩更方便的完成任务， 简约风格的UI更易于用户使用， 合适的字体大小，温和的背景色，合理的按钮位置。保持一致，考虑标准 一致的界面风格易于使用，遵循标准更是使应用程序给用户以专业的感觉。提供验证与纠错 “预防”、“保护”与“通知”是帮助用户改正错误的好的实践。

针对首页的显示发现以下问题可以改进：

1.当鼠标放置于提问搜索按钮上时发现多出背景线，效果如下图

建议去掉背景线

2.针对输入的搜索过的历史查询记录，发现文板框边线并没有对齐效果如下图



建议对齐边框

**建议二、反馈功能的改进**

问答系统中同类产品如下

**新浪爱问**

爱问”包含了一系列传统搜索所不具备的人性化功能：对[视频搜索](http://baike.baidu.com/view/957993.htm" \t "_blank)的支持实现了内容形式上的丰富和扩展；具有浓郁地区特色的[本地搜索](http://baike.baidu.com/view/8643.htm)为网民提供包括地图在内的生活、娱乐、出行等所需的各地区相关信息；而收藏夹则是非常方便与网友共享的网页收藏工具。

与GOOGLE及百度这样的[搜索门户](http://baike.baidu.com/view/1011561.htm" \t "_blank)相比，新浪作为最大的综合门户做搜索有着独特优势。新浪的网民流量与质量都是单[一搜](http://baike.baidu.com/view/42072.htm)索门户所难以比拟的。大流量、高黏性的网民行为使新浪对网民的需求判断更为精准。其次，单一搜索门户所采用的竞价排名商业模式，很容易影响搜索结果的客观性，常使搜索结果与网民需求之间产生错位。而这一点对新浪来说则不是问题，爱问的结果完全是根据网民的行为需求来排序的。

1. **百度知道**

[百度](http://baike.baidu.com/view/262.htm)[知道](http://baike.baidu.com/view/7290.htm)，是用户自己根据具有针对性地提出问题，通过积分奖励机制发动其他用户，来解决该问题的[搜索模式](http://baike.baidu.com/view/4755891.htm)。 同时，这些问题的答案又会进一步作为搜索结果，提供给其他有类似疑问的用户，达到分享知识的效果。

百度知道的最大特点，就在于和搜索[引擎](http://baike.baidu.com/view/53607.htm)的完美结合，让用户所拥有的隐性知识转化成[显性知识](http://baike.baidu.com/view/68071.htm)，用户既是百度知道内容的使用者，同时又是百度知道的创造者，在这里累积的知识数据可以反映到搜索结果中。通过用户和[搜索引擎](http://baike.baidu.com/view/1154.htm)的相互作用，实现[搜索引擎](http://baike.baidu.com/view/1154.htm)的社区化。

[百度知道](http://baike.baidu.com/view/402.htm)也可以看作是对[搜索引擎](http://baike.baidu.com/view/1154.htm)功能的一种补充，让用户头脑中的隐性知识变成显性知识，通过对回答的[沉淀](http://baike.baidu.com/view/127403.htm)和组织形成新的[信息](http://baike.baidu.com/view/1527.htm)库，其中信息可被用户进一步检索和利用。这意味着，用户既是[搜索引擎](http://baike.baidu.com/view/1154.htm)的使用者，同时也是[创造者](http://baike.baidu.com/view/631988.htm)。百度知道可以说是对过分依靠技术的[搜索引擎](http://baike.baidu.com/view/1154.htm)的一种[人性化](http://baike.baidu.com/view/66744.htm)完善。

1. **360问答**

360问答是[360搜索](http://baike.baidu.com/view/9199096.htm" \t "_blank)旗下产品，由用户有针对性地提出问题，并由问答本身的奖惩机制来发动其他用户来解决问题。同时，这些问题的答案又会进一步作为搜索结果，提供给其他有类似疑问的用户，达到分享知识的效果，以此营造“你问大家答”的良好网络知识氛围。同时，依托于奇虎360强大的安全技术支持，360问答在“反作弊、反广告、反垃圾”方面一直成绩显著，致力于为用户打造一个干净、安全、可靠的问答环境。

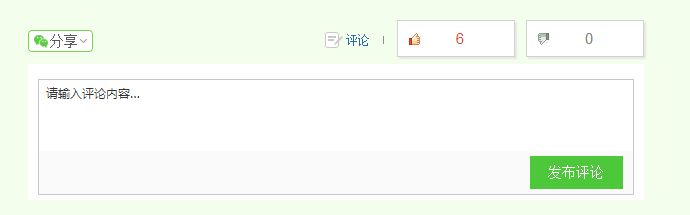
1. **腾讯搜搜**

搜搜主要包括网页搜索、综合搜索、图片搜索、音乐搜索、论坛搜索、搜吧等16项产品，通过互联网信息的及时获取和主动呈现，为广大用户提供实用和便利的搜索服务。用户既可以使用网页、音乐、图片等搜索功能寻找海量的内容信息，也可以通过搜吧、论坛等产品表达和交流思想

1. **知乎**

知乎是一个真实的网络[问答](http://baike.baidu.com/view/166664.htm" \t "_blank)社区，社区氛围友好与[理性](http://baike.baidu.com/subview/30805/12170491.htm)，连接各行各业的精英。[用户](http://baike.baidu.com/view/200012.htm)分享着彼此的[专业知识](http://baike.baidu.com/view/267547.htm)、经验和见解，为中文互联网源源不断地提供高质量的信息。

通过以上同类产品的对比发现 可以增加用户反馈功能

****

从2009年社交网络平台facebook在页面中加入”Like”按钮，点赞功能就在各类社交网络中沿承下来，成为人们在网络交流互动中离不开的工具。无论好事还是倒霉事，都要习惯性地去点一下赞留个印记。

用户可以根据点赞数量和评论内容判断问题是否对自己有帮助，也增加了用户的交互反馈。分享功能



当前互联网广泛流行，面对社交网络，QQ、微信、微博等已经成为人们日程生活的一部分，将问答系统分享到社交工具可以大大增加网络访问量和认知度。

**建议三、首页事例动态化**

  sthml是一种服务器 API，shtml是服务器动态产成的html。 shtml是一种基于SSI技术的文件。服务器端包含 SSI，是英文 Server Side Includes的简写。SSI是一种可以指挥服务器动态声称网页内容的HTML指令。通常SSI可以用来确保网页中的一些通用内容，比如版权信息、联系方式、或导航信息在全站所有网页中保持一致。使用SSI显示这些通用内容还可以减轻网站维护的工作量，因为当需要修改这些通用信息的时候，只需要改一个地方就可以了，而不需要逐个修改每一个网页。各个服务器对SSI命令的支持各有不同，但 #include 和 #exec 是通用的。使用 SSI 的页面文件通常都使用扩展名.shtml，而不是.html 或 .htm，这样以便服务器能够辨认出哪些页面包含SSI指令，这些页面需要先经过服务器处理，翻译执行其中的SSI指令，然后才发送给客户端浏览器。

 　　多数网站寄存服务都支持 SSI – 但是也有一些不支持，特别是一些免费的网站寄存服务。你可以联系你的网站寄存公司询问他们是否支持SSI，也可以通过以下方式来自己测试一下你的服务器是否支持SSI:

1. 拷贝以下HTML内容，保存为文件名test.shtml

<html><head></head>

<body><!–#echo var=”DATE\_LOCAL” –></body>

</html>

2. 将这个文件上载到你的服务器上，然后用浏览器浏览服务器上的这个网页。

3. 如果看到网页显示当前日期，则你的服务器支持 SSI。否则，多半可能你的服务器不支持SSI。你可以尝试与你的网站寄存公司联系看他们是否可以为你开通SSI服务。

在asp.net mvc中，生成纯html静态页面除了有利于seo外，还可以减轻网站的负载能力和提高网站性能。常见的静态化方式是使用HTML页面作为静态化模板，但对开发人员来说将会添加额外的工作量，毕竟同时维护模板和动态页面，是十分烦人的任务。但是如果直接使用ASP.NET MVC本身的视图作为模板，由于使用了视图，开发起来将会更简单。

1.使用视图引擎进行静态化操作，主要使用到了两个类：[ViewEngines类](https://msdn.microsoft.com/zh-cn/library/system.web.mvc.viewengines(v=vs.118).aspx)和[ViewContext类](https://msdn.microsoft.com/zh-cn/library/system.web.mvc.viewcontext(v=vs.118).aspx)。通过ViewEngines类的FindView方法或FindPartialView方法找到要进行静态化的视图页面，然后使用ViewContext类将数据模型填充到视图模板中，获取页面内容后生成静态页面。视图引擎是一个静态的ViewEngineCollection类型对象，可以包含所有已注册的视图引擎。FindView方法迭代ViewEngineCollection中注册的视图引擎，并在每个视图引擎上调用FindView方法，并把视图名称作为参数传入。这就是ViewEngineCollection询问每个视图引擎能否渲染指定视图的方式。FindPartialView方式的工作机制与FindView几乎一样，只是它关注于查找部分视图。

2.使用html模板加上自定义占位符是常见的页面静态化方法，本文将分享在asp.net mvc项目中使用静态模板进行静态化的具体代码。其实具体的实现原理还是蛮简单的，这种方法需要编写对应的html模板页，模板中需要进行数据替换的地方定义好自定义占位符，最后在生成HTML静态页面的时候，将自定义占位符替换成要显示的数据。

比如模板页面中，定义了一个@template\_placeholder\_articlecontent占位符（表示文章的内容），我们就可以使用string.Replace方法将占位符标识替换成所需的文章内容，或者使用正则表达式进行替换。将数据填充到模板后，通过IO操作将网页内容保存为html页面。

目前首页展示效果

所以首页可以用MVC页面静态化和 shtml技术实现 改进成动态的显示

**建议四、利用反向代理服务器加速页面访问速度**

Nginx是一款轻量级的Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。由俄罗斯的程序设计师Igor Sysoev所开发，供俄国大型的入口网站及搜索引擎Rambler（俄文：Рамблер）使用。其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、新浪、网易、腾讯等。Nginx专为性能优化而开发，性能是其最重要的考量， 实现上非常注重效率 。

Nginx的优点有：

1、高并发连接：官方测试能够支撑5万并发连接，在实际生产环境中跑到2～3万并发连接数。

2、内存消耗少：在3万并发连接下，开启的10个Nginx 进程才消耗150M内存（15M\*10=150M）。

3、配置文件非常简单：风格跟程序一样通俗易懂。

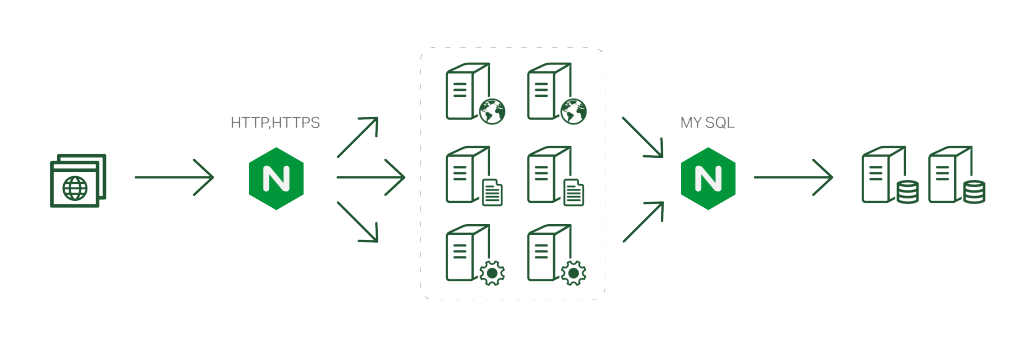
4、成本低廉：Nginx为开源软件，可以免费使用。

5、支持Rewrite重写规则：能够根据域名、URL的不同，将 HTTP 请求分到不同的后端服务器群组。

6、内置的健康检查功能：如果 Nginx Proxy 后端的某台 Web 服务器宕机了，不会影响前端访问。

7、节省带宽：支持 GZIP 压缩，可以添加浏览器本地缓存的 Header 头。

8、稳定性高：用于反向代理，宕机的概率微乎其微。



采用Nigx和IIS分布式部署可以加快页面访问速度，提高网站的并发量。

**建议五、Web安全**

开放式Web应用程序建设离不开网络安全建设

SQL注入攻击，导致网页被篡改或破坏：通过利用SQL命令插入到Web表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令，获得系统或数据库管理员权限，从而达到网页篡改或破坏网页的目的。

DOS、DDOS攻击，导致业务瘫痪：攻击在线业务，造成服务瘫痪，严重影响系统的可用性，DDOS攻击主要就是用大量的计算机攻击一个系统。很多的计算机联合起来就可以发送很多的虚假的请求，以至于被攻击的系统超负荷，而不能向其他的用户提供服务。 蓄意工具者为了发动DDOS，就必须获取足够多的机器。恶意的用户设计在别人的电脑上注入木马和病毒，获取机器的控制权，"借"别人的电脑发送攻击。被控制的电脑就是所谓的"僵尸"。DDOS攻击一般来攻击服务器，而且攻击的方式也是防不胜防，很多的防护软件和防火墙不能区分正确的请求和虚假的请求。

跨站攻击（XSS攻击），导致信息被盗：攻击者利用网站程序对用户输入过滤不足，输入可以显示在页面上对其他用户造成影响的HTML代码，如在用户留言中，用户插入"<script>alert('Error')</script>"，那么我们的留言的页面就中是弹出提示。 跨站脚本的攻击一般表现为一个在URL参数中带有客户端的脚本。这些脚本用来盗取用户的cookie信息等从而盗取用户资料、利用用户身份进行某种动作或者对访问者进行病毒侵害的一种攻击方式；

网站挂马，导致用户形象被破坏：攻击者通过在正常的页面中(通常是网站的主页)插入一段代码，上网者在打开该页面的时候，这段代码被执行，然后下载并运行某木马的服务器端程序，进而控制上网者的主机。

因此程序在开放中要注意web安全策略，使用决策来保护可能最弱的帐户，使用存储过程或者至少是参数化的命令。后台管理密码要用复杂策略，上传模块要限制EXE文件上传。页面要防止xss攻击，服务器修改数据库等远程端口号，加强安全策略。

**建议六、系统后台**

监控实时活动来解决问题和瓶颈

开发和发布高性能应用的关键，在于密切和实时地关注应用程序在现实情况下的性能。你必须能够监控特定设备的活动和网站的基础设施。

大多数监控网站的活动都很被动 —— 它只告诉你将会发生什么，让你自己去发现问题和解决问题。

监控可以抓到不同类型的问题。它们包含：

服务器宕机。

服务器不稳定，容易掉线。

服务器大概率出现缓冲失效。

服务器发送的内容不正确。

你可以使用 New Relic 或 Dynatrace 这种全球性的应用性能监控工具，来监控遥远地方加载页面的时间，也可以利用 NGINX 监控应用程序的发布。当你考虑是否需要给基础设施扩容来维持流量时，应用性能数据可以告诉你这些优化是否真能给用户带来很大的改善。

NGINX Plus 增加了[检查应用程序健康的功能](https://www.nginx.com/products/application-health-checks/) —— 综合一些定期重复性的操作，以及在问题发生时报警，这样可以快速地定位和解决问题。NGINX Plus 还有[会话耗尽功能](https://www.nginx.com/products/session-persistence/" \l "session-draining) —— 当任务完成后终止新的连接，以及慢启动的能力 —— 允许负载均衡集群中的一台服务器，从刚修复的状态慢慢赶上来。如果使用得当，健康检查可以在发生影响用户体验的重大问题前，就定位出问题。会话耗尽和慢启动，允许更换服务器，并保证在过程中不会对性能和正常的运行时间造成不好的影响。下图是一个在 NGINX Plus 中集成了实时活动监控的 dashboard，上面显示了Web 基础设施和服务器、TCP 连接以及缓存等相关信息。

监控实时活动来解决问题和瓶颈

开发和发布高性能应用的关键，在于密切和实时地关注应用程序在现实情况下的性能。你必须能够监控特定设备的活动和网站的基础设施。

大多数监控网站的活动都很被动 —— 它只告诉你将会发生什么，让你自己去发现问题和解决问题。

监控可以抓到不同类型的问题。它们包含：

服务器宕机。

服务器不稳定，容易掉线。

服务器大概率出现缓冲失效。

服务器发送的内容不正确。

你可以使用 New Relic 或 Dynatrace 这种全球性的应用性能监控工具，来监控遥远地方加载页面的时间，也可以利用 NGINX 监控应用程序的发布。当你考虑是否需要给基础设施扩容来维持流量时，应用性能数据可以告诉你这些优化是否真能给用户带来很大的改善。

NGINX Plus 增加了[检查应用程序健康的功能](https://www.nginx.com/products/application-health-checks/) —— 综合一些定期重复性的操作，以及在问题发生时报警，这样可以快速地定位和解决问题。NGINX Plus 还有[会话耗尽功能](https://www.nginx.com/products/session-persistence/" \l "session-draining) —— 当任务完成后终止新的连接，以及慢启动的能力 —— 允许负载均衡集群中的一台服务器，从刚修复的状态慢慢赶上来。如果使用得当，健康检查可以在发生影响用户体验的重大问题前，就定位出问题。会话耗尽和慢启动，允许更换服务器，并保证在过程中不会对性能和正常的运行时间造成不好的影响。下图是一个在 NGINX Plus 中集成了实时活动监控的 dashboard，上面显示了Web 基础设施和服务器、TCP 连接以及缓存等相关信息。

监控实时活动来解决问题和瓶颈

开发和发布高性能应用的关键，在于密切和实时地关注应用程序在现实情况下的性能。你必须能够监控特定设备的活动和网站的基础设施。

大多数监控网站的活动都很被动 —— 它只告诉你将会发生什么，让你自己去发现问题和解决问题。

监控可以抓到不同类型的问题。它们包含：

服务器宕机。

服务器不稳定，容易掉线。

服务器大概率出现缓冲失效。

服务器发送的内容不正确。

你可以使用 New Relic 或 Dynatrace 这种全球性的应用性能监控工具，来监控遥远地方加载页面的时间，也可以利用 NGINX 监控应用程序的发布。当你考虑是否需要给基础设施扩容来维持流量时，应用性能数据可以告诉你这些优化是否真能给用户带来很大的改善。

NGINX Plus 增加了[检查应用程序健康的功能](https://www.nginx.com/products/application-health-checks/) —— 综合一些定期重复性的操作，以及在问题发生时报警，这样可以快速地定位和解决问题。NGINX Plus 还有[会话耗尽功能](https://www.nginx.com/products/session-persistence/" \l "session-draining) —— 当任务完成后终止新的连接，以及慢启动的能力 —— 允许负载均衡集群中的一台服务器，从刚修复的状态慢慢赶上来。如果使用得当，健康检查可以在发生影响用户体验的重大问题前，就定位出问题。会话耗尽和慢启动，允许更换服务器，并保证在过程中不会对性能和正常的运行时间造成不好的影响。下图是一个在 NGINX Plus 中集成了实时活动监控的 dashboard，上面显示了Web 基础设施和服务器、TCP 连接以及缓存等相关信息。

监控实时活动来解决问题和瓶颈

开发和发布高性能应用的关键，在于密切和实时地关注应用程序在现实情况下的性能。你必须能够监控特定设备的活动和网站的基础设施。

大多数监控网站的活动都很被动 —— 它只告诉你将会发生什么，让你自己去发现问题和解决问题。

监控可以抓到不同类型的问题。它们包含：

服务器宕机。

服务器不稳定，容易掉线。

服务器大概率出现缓冲失效。

服务器发送的内容不正确。

你可以使用 New Relic 或 Dynatrace 这种全球性的应用性能监控工具，来监控遥远地方加载页面的时间，也可以利用 NGINX 监控应用程序的发布。当你考虑是否需要给基础设施扩容来维持流量时，应用性能数据可以告诉你这些优化是否真能给用户带来很大的改善。

NGINX Plus 增加了[检查应用程序健康的功能](https://www.nginx.com/products/application-health-checks/) —— 综合一些定期重复性的操作，以及在问题发生时报警，这样可以快速地定位和解决问题。NGINX Plus 还有[会话耗尽功能](https://www.nginx.com/products/session-persistence/#session-draining) —— 当任务完成后终止新的连接，以及慢启动的能力 —— 允许负载均衡集群中的一台服务器，从刚修复的状态慢慢赶上来。如果使用得当，健康检查可以在发生影响用户体验的重大问题前，就定位出问题。会话耗尽和慢启动，允许更换服务器，并保证在过程中不会对性能和正常的运行时间造成不好的影响。下图是一个在 NGINX Plus 中集成了实时活动监控的 dashboard，上面显示了Web 基础设施和服务器、TCP 连接以及缓存等相关信息。