1. 项目中如何处理安全问题（阿里）？

**输入检查**

对于web应用来说，检查的位置主要是前端和后端。前端检查只能防止正常状况，没法阻止通过工具、程序绕开检查直接把数据发送给后端。因此后端也要检查所有输入的数据。

依据产品文档的要求进行合法性的检查，数值型，手机号，时间，邮箱等。

检查文本型是否含控制字符，过滤掉控制字符。

queryString中参数值字符串转义,使用urlencode函数处理

**文件上传**

防止文件注入型攻击

1. 文件上传目录设置为不可执行
2. 判断文件类型。结合MIME type与文件扩展名，设置文件类型白名单。对于图片文件，可以利用图片库函数深层次检查是否真是图片
3. 重命名文件名
4. 文件服务器使用独立的域名

**输出转义**

防止XSS（跨站脚本攻击）。向页面输出的文本型数据进行htmlencode（转义），如将“&”转义成“&amp”，“<”转移为“&lt”，对应的函数：javascriptencode，php的htmlentities。

同理xml数据包里的字符串值使用xmlencode，json数据包的字符串值使用jsonencode转义。

**XSS处理**

XSS本质是HTML注入，用户输入的数据被当成Html代码执行了。

1.cookie使用httpOnly限制

使得在客户端的js代码不能读取cookie值，但是不能防止从http的header里得到cookie值。

1. 输出文本HTML转义
2. 检查输入的URL
3. 对传入js函数的文本型参数值进行js转义

**SQL注入**

预防办法:数据与代码分离，即不用字符串拼凑SQL语句，使用SQL预处理方法（参数占用符?）

**XST处理**

XST(跨站追踪)攻击，关闭web服务器的TRACE方法。

**CSRF攻击的防御**

通常需要防御CSRF(Cross-Site Request Forgeries)攻击的页面是用户登录、修改/删除确认、订单确认等添加新数据、修改数据的页面，此外此类页面必须使用POST方法提交。

**HTTP头注入**

替换HTTP头字符值中的换行符

**Cookie安全**

Cookie仅保存SessionID值，设为HTTPONLY属性为On。

**用户密码安全**

使用安全的bcrypt替代MD5(MD5本来用于生成指纹，用来当加密算法是有误的）,bcrypt会产生随机盐salt

**权限控制**

1.SSO单点登录。OpenId系统

2.Spring security的实现-基于角色的访问控制（Role-Based Access Control）RBAC

3.基于数据的访问控制，比如将用户id、对象id修改成另一个存在的id，如果没有检查当前登录用户是否是参数对应的id用户，就会产生窜改另一个用户数据的问题，也即是越权访问。

4.OAuth授权。OAuth第三方开源库

5.SessionId。sessionId加密处理

2.居中为什么要使用transform(为什么不用marginLeft/Top)(阿里)

Transform属于合成属性（composite property），对合成属性进行transition/animation动画将会创建一个合成层（composite layer），这使得被动画元素在一个独立的层中进行动画。通常情况下，浏览器会将一个层的内容先绘制进一个位图中，然后再作为纹理（texture）上传到GPU，只要该层的内容不发生改变，就没必要进行重绘（repaint），浏览器会通过重新复合（recomposite）来形成一个新的帧。

Top/left属于布局属性，改属性的变化会导致重排（reflow/relayout）,所谓重排即指对这些节点以及受这些节点影响的其他节点，进行css计算=》布局=》重绘过程。浏览器需要为整个层进行重绘并重新上传到GPU，造成极大的性能开销。

3.移动端1px解决方案（ios端）

随着web项目越来越多，在高清屏下，移动端1px会很粗。产生原因跟DPR设备像素比有关,它是默认缩放为100%情况下，设备像素和CSS像素的比值。

Window.devicePixelRatio = 物理像素/ CSS像素

目前的主流的屏幕DPR=2（iphone8），或者3（iphone 8plus）。拿两倍屏来讲，设备的物理像素要实现1像素，而DPR=2，所以css像素只能是0.5.一般设计稿是按照750来设计的，它上面的1px是以750来参照的，而我们写CSS样式是以设备375为参照的，而我们写css样式是以设备375为参照的，所以应该写的0.5px就好了。兼容性IOS 8+系统支持，安卓不支持

1. promise.all并发限制

promise.all(promises:[]).then(fun:function);

Promise.all可以保证，promises数组中所有promise对象达到resolve状态，才执行then回调。

如果是以下这个场景：

如果promise数组中每个对象都是http请求，或者说每个对象包含了复杂的调用处理。而这样的对象有10几万个。

1. await,async的原理

**一句话，async 函数就是 Generator 函数的语法糖。**

1. vue双向绑定的原理以及实现方式
2. Vue采用的是数据劫持的形式
3. bind函数使用方法
4. 解决vue项目在IE中不显示的问题

一、ie11打开vue2.0项目空白,控制台报错 vuex requires a Promise polyfill in this browser

Promise为ES6语法，有的浏览器不支持ES6,所以需要安装babel-polyfill 来将ES6转换为浏览器可识别的ES5语法

①npm install --save-dev babel-polyfill

②在main.js中引入 import "babel-polyfill"

③在build文件夹中webpack.base.conf.js将

//entry: {

// app: ‘./src/main.js’

// },

替换为

entry: {

app: [‘babel-polyfill’, ‘./src/main.js’]

},

二、ie11打不开vue项目，报错 “对象不支持“addEventListener”属性 或方法”

在index.html中head标签中加上

< meta http-equiv=“X-UA-Compatible” content=“IE=EDGE”/>

这个属性主要是设置浏览器优先使用什么模式来渲染页面的。#代码IE=edge告诉IE使用最新的引擎渲染网页