### 各项目目录及根目录下的文件是什么作用(看着项目说)

不知道的：annotation（注解） aspect（切面<拦截器处理订单中心推送的订单状态、数据权限、业务日志>） enums（枚举） mq（消息队列） web/intercepter（拦截器<controller层的跳转url、返回到页面的通用配置和权限>） logback.properties（设置日志的属性） twoDistributionFee.properties（#配置二次分拨默认费用） setting.xml（settings.xml里面是Maven的基本配置，有maven代理、中央库、server认证、镜像等）、pom.xml（Maven坐标<像jar包和war包的构件>、传递性依赖、排除依赖。Project Object Model，项目对象模型。通过xml格式保存的pom.xml文件。作用类似ant的build.xml文件，功能更强大。该文件用于管理：源代码、配置文件、开发者的信息和角色、问题追踪系统、组织信息、项目授权、项目的url、项目的依赖关系等等。）、spring-config-aop.**xml**（切面配置文件，由AopConfig类文件和注解注入到springBean中）

说的不清楚的：feign（java http 客户端库） wrap（包装类，用于接口和controller返回值类型。把每个被选元素放置在指定的 HTML 内容或元素中） XescmAcApplication（是一个简单的可以独立运行的 Web 应用） application.properties（springboot主配置文件，设置开发环境） externalSysUrl.properties（外部系统url配置文件）、edas.provider（服务提供包）、exception(业务异常<BusinessException **extends** RuntimeException>)

Web->controller包下有五个包，分别是base（BaseController和ErrorController<转向错误页面>）、jump、rest（功能实现controller）、test(测试，我用它测试了接口)、worker

Util包下有10个工具类：BigDecimalUtils（四则运算）、CodeGenUtils（redis生成序列号）、DateUtils（日期操作）、JsonUtil（joson与对象、字符串之间的转换）、MQUtil（）、MyMapper(继承自己的MyMapper）、PrimaryGenerater（编号）、PubUtils（正则表达式）、RedisLock、RedisUtils（判断Key是否存在）、Response（返回到页面数据的设置）、StringUtil （字符串操作public class StringUtil extends StringUtils）、UuidUtil（得到没有“-”的uuid字符串）

主要使用的是：BigDecimalUtils（四则运算）、DateUtils（日期操作）、JsonUtil（joson与对象、字符串之间的转换）、PubUtils（正则表达式）、Response（返回到页面数据的设置）、StringUtil （字符串操作public class StringUtil extends StringUtils）

1、@Service用于标注业务层组件   
2、@Controller用于标注控制层组件(如struts中的action)   
3、@Repository用于标注数据访问组件，即DAO组件.   
4、@Component泛指组件，当组件不好归类的时候，我们可以使用这个注解进行标注。      
     @Service public class UserServiceImpl implements UserService { }   
     @Repository public class UserDaoImpl implements UserDao { } getBean的默认名称是类名（头字母小写），如果想自定义，可以@Service(“\*\*\*”)               这样来指定，这种bean默认是单例的，如果想改变，可以使用@Service(“beanName”)   
     @Scope(“prototype”)来改变。可以使用以下方式指定初始化方法和销毁方法（方法名任意）： @PostConstruct public void init() { }

枚举类：  
**public enum** SettleTypeEnum {  
 ***BY\_ORDER***(10,**"按单计费"**),  
 ***BY\_CAR***(20,**"按车计费"**),  
 ***ORDER\_SHARE***(30,**"按车计费分摊到订单"**);//调用下面的构涵SettleTypeEnum（）  
  
 SettleTypeEnum(**int** code, String desc) {  
 **this**.**code** = code;  
 **this**.**desc** = desc;  
 }

调用枚举类  
acSettle.setSettleType(SettleTypeEnum.***BY\_ORDER***.getCode());

### 系统的数据权限是如何控制的

不知道什么是数据权限

不知道怎么实现数据权限的

一般来说，权限有许多种，我们经常用到的一般有操作权限和数据权限两种。

1）所谓操作权限就是有或者没有做某种操作的权限，具体表现形式就是你看不到某个菜单或按钮，当然也有的是把菜单或按钮灰掉的形式。

2）所谓数据权限，就是有或者没有对某些数据的访问权限，具体表现形式就是当某用户有操作权限的时候，但不代表其对所有的数据都有查看或者管理权限。数据权限有两种表现形式：一种是行权限、另外一种是列权限。所谓行权限，就是限制用户对某些行的访问权限。

在controller获取用户信息以参数传到service实现类，使用注解让拦截器拦截到，然后拦截器根据参数值判断是否有数据权限。

项目中，海洋写的注解便是数据权限控制。

@Permission  
@Override  
**public** List<PayableReconciliationVo> queryData(@ValidParam(authType = ValidParam.AuthType.***ADD\_PARAM***) SearchForm form, @ValidParam(isUserInfo = **true**) AuthResDto authResDto) {  
 List<PayableReconciliationVo> dataList = **acAccountPayMapper**.queryData(form);

### 什么是跨域问题，项目是如何解的

不知道

什么是跨域？

跨域，指的是浏览器不能执行其他网站的脚本。它是由浏览器的同源策略造成的，是浏览器对[JavaScript](http://lib.csdn.net/base/javascript" \o "JavaScript知识库" \t "http://blog.csdn.net/lambert310/article/details/_blank)施加的安全限制。两个URL的协议，域名（主机名、IP）和端口都相同则认为是同源，否则有任一个不同则认为跨域。

解决方法：JSONP、CORS、document.domain、WebSocket、ifream、服务器代理、flash socket。

本项目是先在externalSysUrl.properties属性文件配置路径，再在ExternalSysUrlConfig类给此类成员变量赋值，之后在URL拦截器中用request.setAttribute（）放到请求中，最后在视图（ftl文件）中用window.location.href转向此路径指定的页面。

**var** html = **window**.**location**.**href**;  
**var** index = html.indexOf(**"/index#"**);  
url = html.substring(0, index) + **"/index#"** + url;  
**window**.open(url, **"\_blank"**);

### Swagger是什么，项目中是如何使用的

知道Swagger是什么，但不知道项目中是如何使用

1、官方说法：Swagger是一个规范和完整的框架，用于生成、描述、调用和可视化 RESTful 风格的 Web 服务。总体目标是使客户端和文件系统作为服务器以同样的速度来更新。文件的方法，参数和模型紧密集成到服务器端的代码，允许API来始终保持同步。

个人觉得，swagger的一个最大的优点是能实时同步api与文档。在项目开发过程中，发生过多次：修改代码但是没有更新文档，前端还是按照老旧的文档进行开发，在联调过程中才发现问题的情况（当然依据开闭原则，对接口的修改是不允许的，但是在项目不稳定阶段，这种情况很难避免）。

API的英文全称(Application Programming Interface)

2、[先在SwaggerConfiguration类中配置（类注解为@Configuration）并打开swagger功能（类注解为@EnableSwagger2），然后](mailto:先在SwaggerConfiguration类中配置（类注解为@Configuration和@EnableSwagger2）.zhihou) 对实例化service对象后便可以直接调用。

设置类方法为@Bean  
**public** Docket createRestApi() {  
 **return new** Docket(DocumentationType.***SWAGGER\_2***).  
 apiInfo(apiInfo()).*//用来创建该Api的基本信息* select().*//暴露哪些接口来访问，*// 选择那些路径和api会生成documentapis(RequestHandlerSelectors.*basePackage*(**"com.xescm.ac"**)).*//此处暴露整个controller包下类，*// 对所有api进行监控paths(PathSelectors.*any*()).// 对所有路径进行监控  
 build();  
}

调用：Wrapper<List<RmcCarDriverRespDto>> rmcCarDriverRespDtoWrapper = **rmcCarDriverEdasService**.queryDriverListByCarNum(acAccountPayable.getShipmentBus().trim());

### 什么是EDAS，英文、中文全名是什么

不知道

企业级分布式应用服务（EDAS，Enterprise Distributed Application Service）是企业级互联网架构解决方案的核心产品，充分利用阿里云现有资源管理和服务体系，引入中间件成熟的整套分布式计算框架（包括分布式服务化框架、服务治理、运维管控、链路追踪和稳定性组件等），以应用为中心，帮助企业级客户轻松构建并托管分布式应用服务体系。

EDAS是一个以阿里巴巴中间件团队的多款久经沙场的分布式产品作为核心基础组件构建的企业云计算解决方案。

### EDAS的基本原理是什么？

不知道

把一些服务接口单独放在一个项目中，在其他项目中实现这个服务接口，通过调用这些接口实现类调用服务，这样便把服务集成了起来。

### Feign是什么，常用注解是什么

不知道

java http 客户端库，是worker项目与实现worker功能的项目建立联系被worker调用的作用。

常用注解：AcFeignAPI类中的@RequestLine和@Headers

### Paas平台都有哪些中心，对应的英文缩写都是什么

不知道

PAAS平台即(Platform-as-a-Service：平台即服务)，把应用服务的运行和开发环境作为一种服务提供的商业模式。

是指将一个完整的[计算机平台](https://www.baidu.com/s?wd=%E8%AE%A1%E7%AE%97%E6%9C%BA%E5%B9%B3%E5%8F%B0&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkryP-uAcLPHfzrHF9PH6Y0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjnzPjnvrj6dPjDdnj6znW0Y" \t "https://zhidao.baidu.com/question/_blank)，包括应用设计、[应用开发](https://www.baidu.com/s?wd=%E5%BA%94%E7%94%A8%E5%BC%80%E5%8F%91&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkryP-uAcLPHfzrHF9PH6Y0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjnzPjnvrj6dPjDdnj6znW0Y" \t "https://zhidao.baidu.com/question/_blank)、应用测试和应用托管，都作为一种服务提供给客户。在这种服务模式中，客户不需要购买硬件和软件，只需要利用 PaaS 平台，就能够创建、测试和部署应用和服务，与基于数据中心的平台进行软件开发相比，费用要低得多，这是 PaaS 的最大价值所在。

中心有：订单中心（ofc）、*用户中心（*uam*）*、*资源中心（*rmc*）*、*客户中心（*csc*）*

### 什么是Tbschedule

不知道

TBSchedule是一个支持分布式的调度框架，能让一种批量任务或者不断变化的任务，被动态的分配到多个主机的JVM中，不同的线程组中并行执行。基于ZooKeeper的纯[Java](http://lib.csdn.net/base/java" \o "Java 知识库" \t "http://blog.csdn.net/taosir_zhang/article/details/_blank)实现，由Alibaba开源。

### Tbschedule框架的基本原理是什么

不知道

Schedule集群和worker集群注册到zookeeper集群，并且schedule更新到zookeeper集群，之后zookeeper事件推送到worker执行任务。

### 什么是VUE

说不太清楚

是实现了MVVM的框架，也是实现了API响应数据绑定和组合的视图组件。不再使用DOM。

Vue.js（读音 /vjuː/, 类似于 view）是一个构建数据驱动的 web 界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。另一方面，Vue 完全有能力驱动采用[单文件组件](http://cn.vuejs.org/v2/guide/single-file-components.html)和 [Vue 生态系统支持的库](http://github.com/vuejs/awesome-vue" \l "libraries--plugins)开发的复杂单页应用。

Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。

MVVM是Model-View-ViewModel的简写。使开发人员可以将View和逻辑分离出来,这种数据绑定技术非常简单实用

[IMG_256](http://baike.baidu.com/pic/MVVM/96310/0/e61190ef76c6a7efe4baffc3fdfaaf51f2de66b2?fr=lemma&ct=single" \o "MVVM 功能图" \t "http://baike.baidu.com/_blank)MVVM 功能图

### 什么是前后端分离

不知道

简单来说,就是让前端不用再依赖后台进度做开发(当然后台亦然)。

前端人员负责页面展示和交互，然后通过 ajax 调用后端人员写好的数据接口。  
所以开发网站的核心现在基本上就是：

1、html + css ：页面展示

2、javascript + ajax（这个算是浏览器的一部分，可以通过 javascript 调用） : 页面交互、逻辑处理

### 什么是VUE的路由

不知道

Vue.js 路由允许我们通过不同的 URL 访问不同的内容。

<script src="https://unpkg.com/vue/dist/vue.js"></script> <script src="https://unpkg.com/vue-router/dist/vue-router.js"></script>

<router-link to="/bar">Go to Bar</router-link>

### <router-view></router-view> 什么是MQ

不知道

MQ（消息队列）是一种应用程序对应用程序通过发送和检索出入不需要同时执行的队列消息来通信的方法。

Redis是一个KV对nosql数据库，支持MQ功能，完全可以当做一个轻量级队列服务来使用。

消息队列（MQ）是一种应用程序对应用程序的通信方法。应用程序通过写和检索出入列队的针对应用程序的数据（消息）来通信，而无需专用连接来链接它们。消息传递指的是程序之间通过在消息中发送数据进行通信，而不是通过直接调用彼此来通信，直接调用通常是用于诸如[远程过程调用](http://baike.baidu.com/item/%E8%BF%9C%E7%A8%8B%E8%BF%87%E7%A8%8B%E8%B0%83%E7%94%A8" \t "http://baike.baidu.com/_blank)的技术。排队指的是应用程序通过队列来通信。队列的使用除去了接收和发送应用程序同时执行的要求。

MQ 发送普通消息有三种实现方式：可靠同步发送、可靠异步发送、单向(Oneway)发送。

本公司用的MQ产品是redis。

Redis是一个Key-Value的NoSQL[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "http://blog.csdn.net/sunxinhere/article/details/_blank)，开发维护很活跃，虽然它是一个Key-Value数据库存储系统，但它本身支持MQ功能，所以完全可以当做一个轻量级的队列服务来使用。

生产（者）和消费（者）如下：

http://www.cnblogs.com/sgor/p/6536709.html

### 什么是Redis

基本知道

Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、[Key-Value](https://www.baidu.com/s?wd=Key-Value&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkujTdPjnsPWw9nHb1uWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm4nj01n1Tvrjnkrj03P1RYr0" \t "https://zhidao.baidu.com/question/_blank)数据库，并提供多种语言的API。

redis是一个[key-value](https://www.baidu.com/s?wd=key-value&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkujTdPjnsPWw9nHb1uWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm4nj01n1Tvrjnkrj03P1RYr0" \t "https://zhidao.baidu.com/question/_blank)存储系统。

Redis 是一个高性能的[key-value](https://www.baidu.com/s?wd=key-value&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkujTdPjnsPWw9nHb1uWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm4nj01n1Tvrjnkrj03P1RYr0" \t "https://zhidao.baidu.com/question/_blank)数据库。

Redis支持主从同步。

### 分析型数据库是什么

不知道

分析型数据库（AnalyticDB），是阿里巴巴自主研发的海量数据实时高并发在线分析（Realtime OLAP）云计算服务，使得您可以在毫秒级针对千亿级数据进行即时的多维分析透视和业务探索。分析型数据库对海量数据的自由计算和极速响应能力，能让用户在瞬息之间进行灵活的数据探索，快速发现数据价值，并可直接嵌入业务系统为终端客户提供分析服务。

### 什么是SpringMVC

大概知道，但说不太明白

Spring Web MVC是一种基于Java的实现了Web MVC设计模式的请求驱动类型的轻量级Web框架。

即：1）使用了MVC架构模式的思想，将web层进行职责解耦；2）基于请求驱动指的就是使用请求-响应模型；3）轻量级指，帮助我们简化开发，Spring Web MVC也是要简化我们日常Web开发的。

### @RequestMapping里的produces和consumes两个参数含义是什么

没说清楚

Produces是根据请求头中的accept进行匹配，产生设置格式的数据；Consumes是指定处理请求的提交内容类型（media-Type）。

@RequestMapping(value = **"api/storage\_fee/"**, produces = {**"application/json;charset=UTF-8"**})

1）@RequestMapping(value = "/produces", produces = "application/json")：表示将功能处理方法将生产json格式的数据，此时根据请求头中的Accept进行匹配，如请求头“Accept:application/json”时即可匹配; @RequestMapping(value = "/produces", produces = "application/xml")：表示将功能处理方法将生产[xml格式](https://www.baidu.com/s?wd=xml%E6%A0%BC%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YzPvRznjRznjwbPW-bPhf10ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm1rHb4nWcYP1Rvnj0drHDsn0" \t "https://zhidao.baidu.com/question/_blank)的数据，此时根据请求头中的Accept进行匹配，如请求头“Accept:application/xml”时即可匹配。 此种方式相对使用@RequestMapping的“headers = "Accept=application/json"”更能表明你的目的。

2）ConsumesRequestCondition就是通过content type进行url过滤的.

consumes  指定处理请求的提交内容类型（media-Type），例如application/json, text/html.

### @RequestBody和@ResponseBody的含义是啥

说不清楚

@requestBody 读取Request请求的body部分数据。@responseBody用于将Controller的方法返回的对象，写入到Response对象的body数据区。

@requestBody作用：

      i) 该注解用于读取Request请求的body部分数据，使用系统默认配置的HttpMessageConverter进行解析，然后把相应的数据绑定到要返回的对象上；

      ii) 再把HttpMessageConverter返回的对象数据绑定到 controller中方法的参数上。

使用时机：

A) GET、POST方式提时， 根据request header Content-Type的值来判断:

* application/x-www-form-urlencoded， 可选（即非必须，因为这种情况的数据@RequestParam, @ModelAttribute也可以处理，当然@RequestBody也能处理）；
* multipart/form-data, 不能处理（即使用@RequestBody不能处理这种格式的数据）；
* 其他格式， 必须（其他格式包括application/json, application/xml等。这些格式的数据，必须使用@RequestBody来处理）；

B) PUT方式提交时， 根据request header Content-Type的值来判断:

* application/x-www-form-urlencoded， 必须；
* multipart/form-data, 不能处理；
* 其他格式， 必须；

说明：request的body部分的数据编码格式由header部分的Content-Type指定；

@Responsebody

作用：

      该注解用于将Controller的方法返回的对象，通过适当的HttpMessageConverter转换为指定格式后，写入到Response对象的body数据区。

使用时机：

      返回的数据不是html标签的页面，而是其他某种格式的数据时（如json、xml等）使用；

### @PathVariable注解是做什么的

不知道

以路径方式把前台数据传给controller，适用于简单数据类型，可以传多个。

@PathVariable:当使用@RequestMapping URI template 样式映射时， 即 someUrl/{paramId}, 这时的paramId可通过 @Pathvariable注解绑定它传过来的值到方法的参数上。

@RequestParam :  
A） 常用来处理简单类型的绑定，通过Request.getParameter() 获取的String可直接转换为简单类型的情况（ String--> 简单类型的转换操作由ConversionService配置的转换器来完成）；因为使用request.getParameter()方式获取参数，所以可以处理get 方式中queryString的值，也可以处理post方式中 body data的值；  
B）用来处理Content-Type: 为 application/x-www-form-urlencoded编码的内容，提交方式GET、POST；  
C) 该注解有两个属性： value、required； value用来指定要传入值的id名称，required用来指示参数是否必须绑定；

### 什么是Restful风格

不知道

REST即Representational State Transfer的缩写，可译为"表现层状态转化”。REST最大的几个特点为：资源、统一接口、URI和无状态。如果一个架构符合REST的约束条件和原则，我们就称它为RESTful架构。

Restful在项目中主要是设置其他项目路径的。先在application-dev.properties配置文件，后在配置文件把数值注入到springbean中，然后baseController设置传到前端的参数，供前端使用。

### 什么是Git

大概知道,知道分支合并，知道便于协作

Git是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目的项目版本管理。

Git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的[版本控制软件](https://www.baidu.com/s?wd=%E7%89%88%E6%9C%AC%E6%8E%A7%E5%88%B6%E8%BD%AF%E4%BB%B6&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkuyDvrAndnj7WmvnkuWcs0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPH6YPWDzrj6vrHm3P1bLn1md" \t "https://zhidao.baidu.com/question/_blank)。

### 还有哪些和Git类似的工具，他们之间有什么区别

Git是版本控制系统，类似的工具还有Git 、CVS、SVN。

区别为：

1. Git是分布式的，分支和合并更明显，可以多个分支一起工作。其他不是
2. Git不但是版本控制系统，而且有它自己的版本构架，所以也是内容管理系统和工作管理系统。Git关心的是文件而不像其他的关心行变化。
3. Git有一个本地版本库，并且还有互联网方式建立远程核心代码库，这样让开发和提交更安全和方便。cvs没有网络的情况下什么也做不了，文件都没法编辑。
4. Git在各个分支间切换更自由简单。并且像master有权限设置，我们就合并不了。提高了可靠性。
5. Git的版本号变更是针对某个分支的，而SVN是针对整个项目的。

Git是分布式的，

1. Git 不仅仅是个版本控制系统，它也是个内容管理系统(CMS)、工作管理系统等。
2. Git是分布式的，SVN不是，这是Git和其它非分布式的版本控制系统（SVN，CVS）最核心的区别。
3. Git跟SVN一样有自己的集中式版本库或服务器。但Git让每个开发人员从中心版本库的服务器上chect out代码后会在自己的机器上克隆一个自己的版本库。
4. Git把内容按元数据方式存储，而SVN是按文件。所有的资源控制系统都是把文件的元信息隐藏在一个类似.svn、.cvs等的文件夹里。
5. Git可以在分支间简单切换。SVN不可以。
6. Git的完整性要优于SVN
7. Git没有一个全局的版本号，而SVN有。

### Git特点是什么

直接记录快照，而非差异比较

1、分支可以理解为在项目开发过程中，对版本控制的一种模式，也可以说是项目版本升级的一种流程管理。

和cvs和svn相比，更简单直接（直接推送）和具有重复性（子分支）。

2、设置模式是围绕代码中心库的，即：origin。

3、分散版本控制系统，比任何其它版本控制工具都更支持分支和合并。

4、从主分支分下来的分支都是要再合并到主分支上的。