**培训总结**

在3月5号的那一天，我进入上海象融信息有限公司进行为期二周的培训，今天是培训完该做汇报的日子，下面是我的汇报总结。

培训内容：

1. 前端：使用aws平台进行有关业务流程方面对的开发，运维和实施
2. 后端：使用IDEA,postman等开发辅助工具进行后台开发。

我对培训完知识的总结：

1. 为开发者和非开发者构建下一代企业应用提供了完整、专业的基础设施工具， 包括BPM（Business Process Management，业务流程管理）的全部能力、应用建模、移动管理、开放API和复杂后端集成服务。
2. Aws Paas构建的应用包括流程类应用，数据类应用，协同类应用，移动类应用，集成类应用，服务类应用。

2.1**流程类应用**：

适用场合：适合普遍存在于企业/政府组织中的各种流程，合规连接组织内外的业务、人员和系统。如审批类流程、业务处理流程。

基本步骤：

1. 设计流程模型，测试流程合理性；
2. 设计BO模型；
3. 设计表单模型；
4. 将表单绑定到流程节点；
5. 设计启动权限，部署到客户端使用。

好处：**AWS PaaS为构建流程类应用提供了强大的建模工具和引擎，可以通过配置完成复杂的流程应用。**

2.2**数据类应用:**

适合场合：适合普遍存在于企业/政府组织中的各种数据管理，如业务台账、数据记录的增删改查、查询报表、统计报表。

基本步骤：

1. 设计DW模型(Data Windows)；
2. 部署到客户端使用。

好处：**AWS PaaS为构建数据类应用提供了强大的建模工具和引擎（DW，Data Windows），可以通过配置，完成复杂的数据交互类应用和报表，为操作者提供互联网级的用户体验。**

**2.3**协同类应用****

**适合场合：**适合不能用建模配置完成的****无模式应用****，比如一些提高个人和团队效率的办公协作应用（如日程、投票）。这类****无模式应用****并不局限在办公协作，任何不适合免代码构建的场景，都可以采用该方式开发实现。

基本步骤：

1.本地安装AWS开发环境和IDEA开发工具（ecplice开发工具）；

2.基于AWS MVC编程框架进行代码开发，调试；

3.测试，打包分发到生产环境；

4.部署到客户端使用。

好处：**AWS是一个支持全栈Java 8编程语言的Java PaaS，开发者可以使用自己熟悉的IDE，开发和调试运行在AWS 上的Java程序并且**开发者直接调用AWS的Jar资源和SDK API，也可以在自己app资源工程加入自己熟悉的Jar资源和前端JS框架。

2.4**移动类应用**：

适用场合：适合运行在手机和平板的移动场景。在AWS PaaS提供了完整的[企业移动管理](https://docs.awspaas.com/emm/aws-paas-emm-solution/index.html" \t "https://docs.awspaas.com/help/paas/_blank)服务（EMM，Enterprise Mobile Management），即支持集成第三方移动应用也支持开发新移动应用。（开发原生移动应用，开发H5移动应用）。

开发原生移动应用基本步骤：

1. 在相关应用里，为移动app封装ASLP服务；
2. 开发者使用Android/iOS开发工具，完成移动App的开发，调试；
3. 创建AWS Paas的移动类型应用，上传编译的移动应用程序包；
4. 通过MAM授权安装权限，用户在移动端安装，访问。

开发H5移动应用

好处：HTML5已经成为企业移动解决方案的主流技术路线，一次开发可运行在各种移动设备。此种情况下，AWS PaaS定位成一个H5应用快速生成工具和轻量级移动开发框架。

**H5自动生成功能：**

移动表单，适合于在手机，平板设备访问流程表单。

轻量级移动开发架构：

1. 基于WAS MVC编程框架；
2. 推荐使用HBuilder开发，调试H5代码；
3. 使用内置的H5开发框架（MUI，阿里巴巴Weex，百度Echarts图标），或者引入团队习惯的前端框架。

基本步骤：

1.本地安装AWS开发环境和HBuilder开发工具；

2.基于AWS MVC编程框架进行代码开发，调试；

3.配置应用入口，测试，打包分发到生产环境；

4.部署到客户端使用。

2.5**集成类应用**：

适用场合**：**企业级应用，集成几乎无处不在。集成类应用适用于为上层场景提供集成服务。

只要包括：

1. [组织、身份](https://docs.awspaas.com/apps/com.actionsoft.apps.addons.ldapsync/" \t "https://docs.awspaas.com/help/paas/_blank)集成，一次身份验证，[免登录访问各个系统](https://docs.awspaas.com/reference-guide/aws-paas-sso-reference-guide/index.html" \t "https://docs.awspaas.com/help/paas/_blank)；
2. 流程集成，通过[事件触发器](https://docs.awspaas.com/reference-guide/aws-paas-process-listener-reference-guide/index.html" \t "https://docs.awspaas.com/help/paas/_blank)与外部系统的数据、API进行交互；
3. 表单、数据集成，通过UI组件直接读取外部系统的数据；
4. [内容集成](https://docs.awspaas.com/solution-package/aws-paas-app-solution-package-portal/pc_portal/page.html" \t "https://docs.awspaas.com/help/paas/_blank)，抓取外部系统信息，展示到门户；
5. [移动集成](https://docs.awspaas.com/emm/aws-paas-emm-solution/architecture/mam.html" \t "https://docs.awspaas.com/help/paas/_blank)，将企业散落的移动App整合到一个访问入口；
6. [OPEN API](https://docs.awspaas.com/reference-guide/aws-paas-api-guide/index.html" \t "https://docs.awspaas.com/help/paas/_blank)，将AWS PaaS的操作、数据以API形式开放给外部系统。

2.6**服务类应用：**

适用场合：API是AWS PaaS开放给开发者的编程接口，用来与AWS PaaS各种引擎和程序服务资源进行跨网络、跨设备、跨系统的交互。AWS PaaS的基础架构，是一个原生的API Server，开发者可以将任何处理快速封装成互联网Web API，提供给第三方系统，利用AWS PaaS的API架构设施，可以将这些服务封装成自己的应用，统一向外部提供交互服务。（AWS Paas API）

常用场景：

1.以流程引擎为服务的方式，提供给第三方；

2.将Paas的后台处理逻辑，以HTTP+JSON方式提供给移动端应用；

3.与企业内部系统和外部SaaS/B2B服务，安全的进行API互操作。

AWS Paas 的**调度服务模块**：定时器

1. 常规Job：在指定的触发条件执行一次或周期重复执行指定的Java程序。

基本步骤：

1.1登录您的AWS PaaS实例控制台；

1.2访问“调度服务”页面；

1.3点击“新建”按钮；

1.4输入Java程序所在的应用，并选择“Job 常规定时任务”；

1.5点击“确定”按钮，完成新建；

1.6填写任务名称和任务实现类，实现类要在IDEA开发工具中去写，将写完过后，测试通过的代码打包分发到was上。

1. SOAP JOb：在指定的触发条件执行一次或周期重复执行指定的SOAP服务，该服务指向一个AWS CC的SOAP适配器。

基本步骤：

1. 在CC，注册SOAP适配器

1.1登录您的AWS PaaS实例控制台；

1.2访问“连接服务”页面；

1.3点击“新建”按钮；

1.4选择“访问SOAP Web服务”；

1.5点击“确定”按钮，完成新建。

1. 新建Job

2.1访问“调度服务”页面；

2.2点击“新建”按钮；

2.3选择“SOAP Job // 定时访问SOAP Web服务”；

2.4点击“确定”按钮，完成新建

1. 完成配置

3.1填写任务名称和CCSOAP的实现类。

3.HTTP Job：在指定的触发条件执行一次或周期重复执行指定的HTTP服务，该服务指向一个AWS CC的HTTP适配器。

基本步骤：

1.在CC，注册SOAP适配器

1.1登录您的AWS PaaS实例控制台；

1.2访问“连接服务”页面；

1.3点击“新建”按钮；

1.4选择“访问Http(s) Web服务”；

1.5点击“确定”按钮，完成新建。

2.新建Job

2.1访问“调度服务”页面；

2.2点击“新建”按钮；

2.3选择“HTTP Job // 定时访问Http(s) Web服务”；

2.4点击“确定”按钮，完成新建

3.完成配置

3.1填写任务名称和CCSOAP的实现类。

3.SQL Job：在指定的触发条件执行一次或周期重复执行指定的数据库SQL。

基本步骤：

1.1登录您的AWS PaaS实例控制台；

1.2访问“调度服务”页面；

1.3点击“新建”按钮；

1.4输入Java程序所在的应用，并选择“SQL Job // 定时执行数据库SQL”；

1.5点击“确定”按钮，完成新建；

1.6填写任务名称和任务实现类，实现类要在IDEA开发工具中去写，将写完过后，测试通 过的代码打包分发到was上。

4.NativeCall Job：在指定的触发条件执行一次或周期重复执行指定的外部Shell命令，该服务指向一个AWS CC的NativeCall适配器。

基本步骤：

1.在CC，注册SOAP适配器

1.1登录您的AWS PaaS实例控制台；

1.2访问“连接服务”页面；

1.3点击“新建”按钮；

1.4选择“执行外部Shell命令”；

1.5点击“确定”按钮，完成新建。

2.新建Job

2.1访问“调度服务”页面；

2.2点击“新建”按钮；

2.3选择“NativeCall Job // 定时执行外部Shell命令”；

2.4点击“确定”按钮，完成新建

3.完成配置

3.1填写任务名称和CC NativeCall的实现类。

AWS Paas 的**连接服务模块：**CC（Connect Center，连接中心），是AWS PaaS对外提供服务接口、对内提供外部资源连接API的组件。

CC好处：

1. 轻量简单。CC对开发友好，提供了多种类型的常用API，能快速上手并使用统一的方式解决开发中遇到的常见问题，规避类库依赖和一些容易忽略的问题。
2. 系统中部分业务模块通过CC功能，能和业务实施协同，提高交付效率。

CC的使用：以管理员账户登录AWS PaaS控制台，进入连接服务,点击左上角新建按钮，创建相应适配器服务连接，然后在AWS PaaS平台提供的相应调用入口进行数据读写操作。如列表、网络数据字典、数据视图等数据源，API调用等。

1. **RDS**：RDS（Relational Database Service，关系型数据库服务） Adapter是一种可让开发者在程序和建模时直接访问PaaS外部关系型数据库服务的技术适配器。****数据库连接超时时间为6秒。****
2. SOAP：SOAP（Simple Object Access Protocol，简单对象访问协议） Adapter是一种可让开发者在程序和建模时直接访问PaaS外部Web Service服务的技术适配器。
3. HTTP：HTTP（HyperText Transfer Protocol，超文本传输协议） Adapter是一种可让开发者在程序和建模时直接访问PaaS外部Web URL服务的技术适配器。****该类适配器支持直接通过CC API使用。****

****AWS组织服务模块：****组织是工作流路由寻找参与者的主要资源之一，组织模型建模是对现实企业组织管理架构的映射过程。

AWS PaaS提供了一组工具来维护、管理这些模型，以适应组织架构的动态变化。

在AWS PaaS中，组织模型被存储在本地BPM数据库中，并提供了集成企业[LDAP/AD的组织同步工具](https://docs.awspaas.com/user-manual/aws-pass-app-user-manual-addons-ldapsync/index.html" \t "https://docs.awspaas.com/user-manual/aws-pass-console-user-manual-org/organization/_blank)。其身份验证支持的方式多种多样，即允许在本地验证，也可以连接到LDAP或指定的DB数据库或三方的Portal SSO令牌验证。

AWS的组织模型结构层次清晰，按照组织对象管理属性被分为以下主要模型：

* ****1.**[单位](https://docs.awspaas.com/user-manual/aws-pass-console-user-manual-org/organization/unit.html)** 对单一企业或集团多架构的单位属性进行描述
* ****2.**[部门](https://docs.awspaas.com/user-manual/aws-pass-console-user-manual-org/organization/department.html)** 部门必须在一个单位或部门下，部门下允许定义人员和子部门
* ****3.**[人员](https://docs.awspaas.com/user-manual/aws-pass-console-user-manual-org/organization/people.html)** 人员必须定义在部门下，一个人有一个主部门和主角色，可定义多个兼任部门的兼任角色
* ****4.**[角色](https://docs.awspaas.com/user-manual/aws-pass-console-user-manual-org/organization/role.html)** 一组抽象功能的权限集合，一个角色也是一组人员的动态集合
* ****5.**[团队](https://docs.awspaas.com/user-manual/aws-pass-console-user-manual-org/organization/team.html)** 一组人员的动态集合
* ****6.**[虚拟组织](https://docs.awspaas.com/user-manual/aws-pass-console-user-manual-org/organization/virtual_orgnization.html)** 它是实体组织的影子，其人员、岗位来自实体组织的引用，但结构是虚拟的

**AWS权限服务模块**

**AWS导航服务模块：**用于维护AWS PaaS平台功能模块的菜单结构，通过权限组"菜单权限"授权给客户端访问。导航菜单从层次上最多支持三级，被分为子系统、目录和功能，在运行时刻，各种主题风格门户有不同表现风格。

**AWS实例运行管理模块：**统一对AWS PaaS平台的流程实例和任务实例进行运维管理，可通过跟踪相关的流程和任务，查看它们对应的状态，并且对无执行人的任务分配执行人。

**AWS服务质量监控模块：**[SLA](http://baike.baidu.com/view/163802.htm" \t "https://docs.awspaas.com/reference-guide/aws-paas-sla-reference-guide/introduction/_blank)(Service-Level Agreement，服务等级协议）在AWS PaaS中被定义为一组保持各项服务质量和连续性的指标，其目的在于通过对PaaS资源/服务指标的连续监控和分析，帮助客户的技术团队及时发现服务质量隐患，并为解决或改善问题提供诊断线索。

无论AWS平台服务运行在私有部署环境还是云中，企业都依赖一种可靠的7X24小时服务监控系统，对应用服务质量进行评估和改进，并期望在第一时间获得告警和应对措施。

**AWS日志审计查询：**主要为AWS PaaS内部的原生应用提供日志服务。

1.系统日志：系统日志是排除出错故障和调试信息打印的运行记录。日志数据存放在独立的文件中。

AWS服务的日志文件(WARN/ERR)存放在%AWS-HOME%/logs目录下，Web服务日志(INFO/WARN/ERR)存放在%AWS-HOME%/logs目录下。

2.审计日志：AWS的审计日志数据涵盖了应用开发、部署、运行和维护全周期，并对安全、风险等特定场景提供专项记录。所有这些日志的收集默认开启。审计日志数据存放在数据库中。

通过二周的培训，我学习到了以上的知识，其中肯定会有许多没提到的部分，还学要我进一步的去学习。

基于这个AWS Paas 平台的开发，通过做的一些案例，我觉得自己在流程类应用，数据类应用方面的开发比较熟练，对于服务类应用很欠缺，因为里面涉及道路很多AWS Paas API，有很多API的使用功能知道的不是很多，这个要在以后的项目中更加的去熟练使用它们。