

# TAE 云服务

## 下载依赖包

### 摘要

TAE 面向所有业务和编程语言的开发者,提供常用的互联网服务。这些服务包括:关系型数据库服务(RDS)、开放存储服务(OSS)、开放缓存服务(OCS)、日志服务、图片服务等。本篇针对 Java 开发者,展示如何在 Java 工程中引入服务包依赖。PHP 和 Python 开发者无需任何依赖,可以直接对服务接口进行调用。

### 准备工作

[安装配置 TAE 本地 SDK](#)

### 云服务依赖包的引入

云服务包提供了对各种服务进行调用所需的 API。应用必须引入服务包依赖才可以对服务进行调用。

### 普通引入

对于普通开发者,请在应用的 ClassPath 中引入 `\api\services-api-{version}` 和 `\api\services-jae-api-{version}`。其中 version 代表依赖包的版本号,请及时更新本地 SDK 以获取对最新接口的支持。

### Maven 服务包引入

对使用 maven 作为构建工具的开发者来说,请在 pom.xml 中添加服务包的仓库地址

```
<repositories>
  <repository>
    <id>taerepository</id>
    <url>http://mvnrepo.tae.taobao.com/content/groups/public/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
  <repository>
    <id>taesnapshots</id>
    <url>http://mvnrepo.tae.taobao.com/content/repositories/snapshots/</url>
    <releases>
      <enabled>>false</enabled>
    </releases>
    <snapshots>
```

```
        <enabled>true</enabled>
    </snapshots>
</repository>
<repository>
    <id>taoreleases</id>
    <url>http://mvnrepo.tae.taobao.com/content/repositories/releases/</url>
    <releases>
        <enabled>true</enabled>
    </releases>
    <snapshots>
        <enabled>>false</enabled>
    </snapshots>
</repository>
</repositories>
```

和服务包依赖

```
<dependency>
    <groupId>com.alibaba.appengine</groupId>
    <artifactId>services-api</artifactId>
    <version>1.4.27-SNAPSHOT</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>com.alibaba.appengine</groupId>
    <artifactId>services-jae-api</artifactId>
    <version>1.0.8-SNAPSHOT</version>
</dependency>
```

请根据各服务具体章节调整服务依赖的版本号，以获得最新接口。

## 云服务的调用

读者可使用服务包提供的 API 调用云服务，具体细节参看各服务章节以及 [Java Doc](#)。

## 数据库服务

### 数据库服务简介

数据库服务基于阿里云关系型数据库服务（Relational Database Service，简称 RDS），为开发者提供了稳定可靠、可弹性伸缩的在线数据库服务。RDS 采用即开即用方式，兼容 MySQL、SQL Server 两种关系型数据库，并提供数据库在线扩容、备份回滚、性能监测及分析功能。

Database 服务[配置说明](#)

### 数据库服务 API

本节对数据库服务 API 的使用方式进行介绍。

#### 1、数据库连接

```
$pdo = new PDO();
```

新建 PDO 对象时无需传入配置信息，系统会自动关联您所开通的数据库服务。

## 2、创建表

强烈建议不要使用 PDO 的方式创建表，表的创建和结构的修改请在 Workstation 的数据库管理中操作。

[参考文档](#)

## 3、数据插入

一般的数据插入方法

```
$resultDataInsert = $pdo->exec("INSERT INTO `Person` (`name`,`address`,`age`)VALUES ('小二','阿里巴巴',24)");
```

使用预处理进行数据插入

```
$prepareDataInsert = $pdo->prepare("INSERT INTO `Person` (`name`,`address`,`age`)VALUES ('小小二','阿里巴巴',28)");
```

```
$prepareDataInsertResult = $prepareDataInsert->execute();
```

使用占位符进行数据插入

```
$dataInsertParam = array("大小二", "阿里巴巴", 29);
```

```
$placeholderDataInsert = $pdo->prepare("INSERT INTO Person (name, address, age) values (?, ?, ?)");
```

```
$placeholderDataInsertResult = $placeholderDataInsert->execute($dataInsertParam);
```

这里的占位符对应的值需要与数组中的值顺序保持一致。

使用命名占位符进行数据插入

```
$namedPlaceholderData = array(':name' => '二小', ':address' => '阿里巴巴', ':age' => 23);
```

```
$namedPlaceholderDataInsert = $pdo->prepare("INSERT INTO Person (address, name, age) value (:address, :name, :age)");
```

```
$namedPlaceholderDataInsertResult =
```

```
$namedPlaceholderDataInsert->execute($namedPlaceholderData);
```

注意，这里数组的 key 必须添加:，数组可以无序。

## 4、数据查询(FETCH\_ASSOC 方式)

```
$resultDataQueryAssoc = $pdo->query("select name , address as adr,age from Person;");
```

```
$resultDataQueryAssoc->setFetchMode(PDO::FETCH_ASSOC);
```

```
while ($row = $resultDataQueryAssoc->fetch()) {
```

```
    echo $row['name'];
```

```
    echo $row['adr'];
```

```
    echo $row['age'];
```

```
}
```

除此之外还可以使用 FETCH\_OBJ 、 FETCH\_CLASS 等方式。

## 5、数据更新(FETCH\_ASSOC 方式)

```
$resultDataUpdate = $pdo->exec("UPDATE `Person` set age=100 where name='小二;");
```

# 分库分表数据库服务

## 分库分表数据库服务简介

为了解决一个表中数据库量比较大的情况下的性能问题，我们引入阿里云的 drds 服务，它可以像使用传统关系型数据库一样来使用它用于数据处理、计算和存储，但是在处理海量数据时又具有良好的扩展性和性能，系统的升级、扩容成本非常低廉，并且能确保在扩容、升级期间对生产系统几乎没有影响，这个服务的使用方法和一般的数据库一致，尤其适合高并发、海量数据读写、对跨库一致性要求相对较低的应用系统。

## 数据库服务 API

你需要从 [workstation 的控制台](#) 获取到你的 drds 数据库服务的名字，然后通过下面的代码获取数据源。比如你的 drds 数据库服务的名字是 drds。获取数据源后的增删改查使用方式和普通的数据源一致

( <http://tae.taobao.com/support/article/57> )，为了获得更好的性能，增删改查的时候最好带上分表字段。问题反馈请找旺旺：drdsoncall

```
$pdo = new PDO("rdsAlias=drds");
```

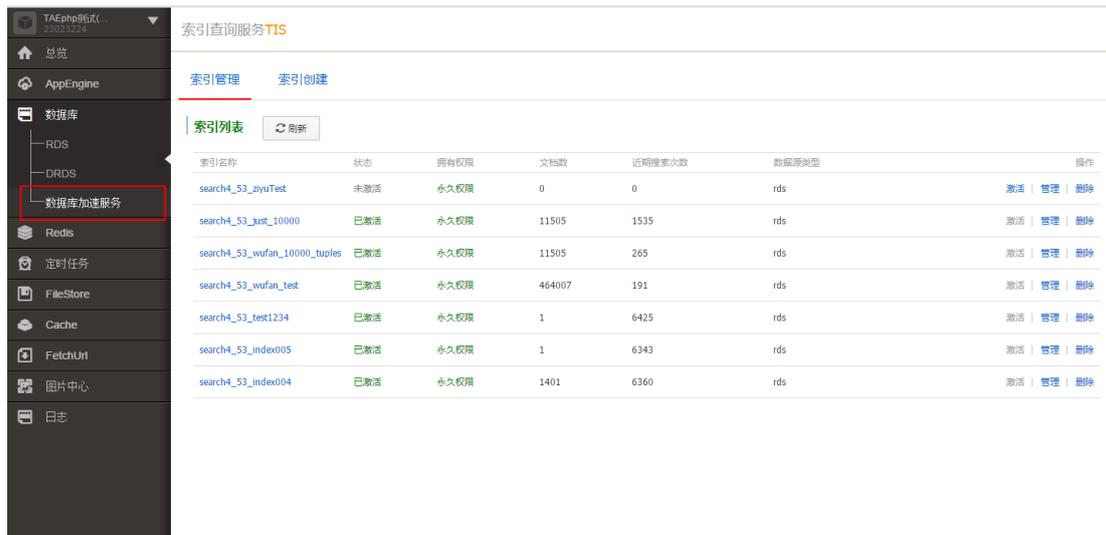
# 数据库加速服务

## 简介

索引查询服务 ( terminator index searcher ，简称 TIS ) 是一套为广大 ISV ，商家量身定制的，高效、稳定的云端查询加速服务。TIS 可轻松解决大数据背景下数据库的模糊、聚合、多表合并查询存在的性能瓶颈。使用 TIS 可以有效缓解复杂查询对数据库的压力，从而保证企业 IT 系统整体的稳定性。

### 1、进入数据库加速服务页：

进入 workstation 的页面，选择左边的数据库下面的数据库加速服务，点击进入：



进入 TIS 索引查询服务控制台页面后，会有索引管理和索引创建两个主要选项卡。点击索引管理，会出现对应的索引列表页面：

- (1) 如果索引列表为空或需要创建新的索引，则需要首先点击索引创建按钮，进行[索引创建](#)；
  - (2) 如果当前已有索引存在，但尚未激活，可以点击激活按钮，进行[索引激活](#)；
  - (3) 如果当前已有索引存在，并且已激活，可以点击管理按钮或者索引名称，直接进行[索引管理](#)；
- 如果索引不再使用，可以直接点击删除按钮，进行删除。

索引查询服务TIS

索引名称	状态	拥有权限	文档数	近期搜索次数	数据源类型	操作
search4_53_just_10000	已激活	永久权限	11505	1378	rds	激活   管理   删除
search4_53_wufan_10000_tuples	已激活	永久权限	11505	114	rds	激活   管理   删除

## 2、索引创建

点击索引创建按钮，可以进入创建索引流程，主要有以下五个步骤，下面依次详细说明。

### 创建索引



### 2.1 索引命名

索引名称为 4-20 个字符，以大小写字母开头，由字母和下划线组成，索引名长度：0/20，建议输入业务相关的名称，最好简单易记。

备注框里可以填写业务相关的信息，方便以后的查看。

### 2.2 选择数据类型和规模

目前提供 RDS 数据源供选择，后期会有多种数据源支持，点击一下 RDS 图标，可以看到需要选择数据规模，这里的数据规模决定了服务需要的相应资源，每一个选项上都有相关的解释。

- (1) QPS 规模：即每秒查询峰值上限，超过此上限的查询请求将被丢弃。
- (2) 文档规模：索引文档数的上限，超过此上限，新增加的文档将无法构建索引。
- (3) 平均查询命中数：单次查询可能命中的文档数，命中文档数越多，应用的系统资源要求越高。



### 2.3 选择数据源

数据源选择页面 这里会自动展示出当前账号在TAE内的RDS实例ID,以及当前实例下的所有数据库。点击数据库名,会展示出该库下面的所有表名。选择需要创建索引的表名,点击添加。注意:用户可以选择其中一张或多张表,请注意,多张表,必须保证表的字段结构一模一样,否则无法通过验证。

创建索引

① 索引命名 ② 选择数据类型和规模 ③ 选择数据源 ④ 定义元数据 ⑤ 确认



选择多张字段结构不一致的表是无法通过验证的。



## 2.4 元数据管理

元数据管理页面会展示出当前已选择表的所有字段信息，对于每一个字段可以选择相关操作：

- (1) 字段类型：创建索引时需要设置的类型,可以与数据库中不一致；
  - (2) 设置主键：数据库主键字段；
  - (3) 设置默认值：数据库内容为空的字段必须设置默认值；
  - (4) 数据处理：数据构建索引前可以进行预处理；
  - (5) 可搜索：勾选表示该列可以被单独搜索；
- 注意：“可搜索”和“可展示”至少选择一个；
- (6) 可展示：勾选表示搜索结果集将包含此列；
- 注意：全部勾选，可能导致索引数据量增加，查询中不需要返回的字段建议取消勾选此项；
- (7) 区间查询：勾选后，此列支持区间查询，仅支持 int 或者 long 型字段的区间查询；
  - (8) 分词方法：字段类型为 string 的字段支持分词，可以自由选择需要的分词方法，常用的分词方法有包括：  
(a) 庖丁分词：采用庖丁分词器进行分词；  
(b) like 查询分词：类似于数据库的模糊查询；  
(c) 空格分词：采用空格分词，常用于英文或空格分开多字符的字段；  
(d) 正则分词：支持自定义的分词方法，例如：该列以标点符号进行分词，可以输入对应的标点符号，将会采用该标点进行分词。
  - (9) 删除：根据业务需求可以选择删除某个不需要构建索引的字段；
  - (10) 添加：添加新的自定义字段，用于新的业务需求；

数据表:trade\_order

字段名称	字段类型	主键	默认值	数据处理	可搜索 ?	可展示 ?	区间查询 ?
1 id	int	x	+	+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2 price	double	x	+	+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3 persum	int	x	+	+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4 companyid	int	x	+	+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5 totalsum	long	x	+	+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

数据分组键:  uniqueKey:  备注: 添加备注信息

页面左下方，需要选择数据分组键，这个分组键是当数据源很大时，需要进行索引切分，将索引分成多片，部署在不同机器上，而需要采用的字段，这里一般选择主键。后面的 uniqueKey 是表示索引的唯一键，避免索引中出现重复的记录数，一般默认为主键。

## 2.5 确认

编辑元数据后，点击下一步，会进入索引确认页面

### 创建索引



索引名称	ziyuTest
创建用户	
数据来源	RDS
数据库名	rds-testppp-307
数据库表	category
数据列数	7

上一步

确认创建索引

取消

页面上会展示索引名称、创建用户、数据来源、数据库名、数据库表、数据列数等信息。核对无误之后，可以点击“确认创建索引”，如果有问题，可以点击上一步，返回重新编辑。

点击“确认创建索引”后，索引任务已成功创建，但是尚未激活，需要激活，才能索引数据，并提供搜索服务。

### 创建索引



用户可以直接点击页面上的“激活索引”，进行索引激活，或者返回索引列表，在索引列表页面点击激活按钮。

## 3. 索引激活

用户在索引创建最后一步，或者索引列表点击激活按钮，会进入索引激活流程。

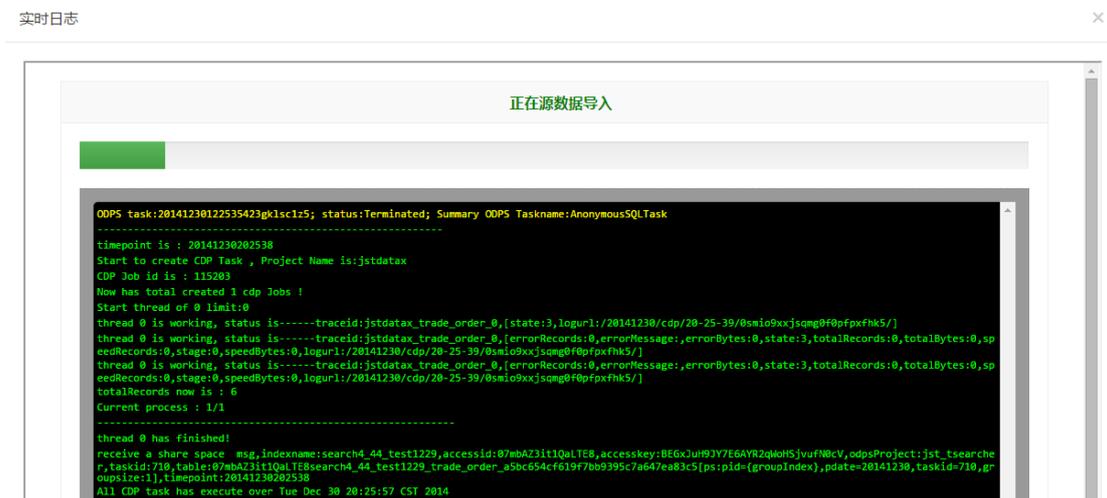
### 3.1 选择数据规模

点击激活，出现如下页面，用户需要选择导入全部数据还是最多导入一万条数据，一般情况下，如果是测试，建议选择最多导入一万条数据，保证资源的合理利用。



### 3.2 实时日志

点击 2.1 中的确认激活后,会出现当前索引激活的实时日志,这个过程主要包括四个部分:(1)源数据导入;(2)索引构建;(3)索引切换;(4)TIS 服务发布。页面上会展示出当前索引激活过程中的实时状态:



当页面上出现如下状态时,表示 TIS 服务发布成功,业务方可以开始调用 TIS 的 API 进行索引查询服务。



注意：数据源的数据量规模，决定了激活过程的时间长短，如果数据量很大的话，建议一段时间之后，再观察是否成功，或者直接关掉实时日志窗口。

关闭实时日志窗口后，页面将进入当前索引的索引管理页面。

#### 4. 索引管理

在索引列表页面点击索引名称，或点击了索引名称右方的管理按钮，或者索引激活完成后，都将会进入索引管理页面。

如果出现如下页面，表示索引服务已经创建成功。请直接进入索引基本信息页面。

索引名: **search4\_53\_just\_10000** [← 返回索引列表](#)

索引基本信息 | [索引查询](#) | [元数据管理](#) | [配额管理](#) | [日志历史信息](#) | [数据统计](#) | [同步数据](#)

**基本信息** [刷新](#)

总文档数	总请求数	秒请求数	错误数
11505	1385	0.01	0

**索引数据源信息**

数据源类型	数据库名称	表名称	应用创建者	应用创建时间	最近修改时间	操作
rds	rds-testppp-307	auction_auctions_0001	c测试账号150	2015/01/12 16:27:22	2015/01/12 16:27:22	<a href="#">修改数据源</a>

如果出现如下页面，表示索引尚未激活成功，或者出现相关故障。

索引基本信息 | [索引查询](#) | [元数据管理](#) | [配额管理](#) | [日志历史信息](#) | [数据统计](#) | [同步数据](#)

**当前索引不可用，请重新构建索引**

可能的原因

- 服务到期或未激活
- 构建索引数据失败
- 服务器异常

请尝试 [激活](#)

如有疑问请加入TIS技术支持群咨询：1205087413

**索引数据源信息**

数据源类型	数据库名称	表名称	应用创建者	应用创建时间	最近修改时间	操作
rds	rds-testppp-307	category	c测试账号150	2015/01/14 14:29:16	2015/01/14 14:29:16	<a href="#">修改数据源</a>

相关故障分析：点击日志历史信息选项卡，如果该索引构建的“状态”栏是正在进行中，表示当前索引正在创建，请耐心等待；如果是失败，可以返回索引基本信息页面，重新激活。如果是成功，请联系 TIS 技术支持，可以加入旺旺群：1205087413。

任务编号	任务类型	触发时间	状态	错误信息	日志详细信息	查看实时日志
707	索引构建	2014/12/30 19:53:28	正在进行中		<a href="#">查看详情信息</a>	<a href="#">实时日志</a>

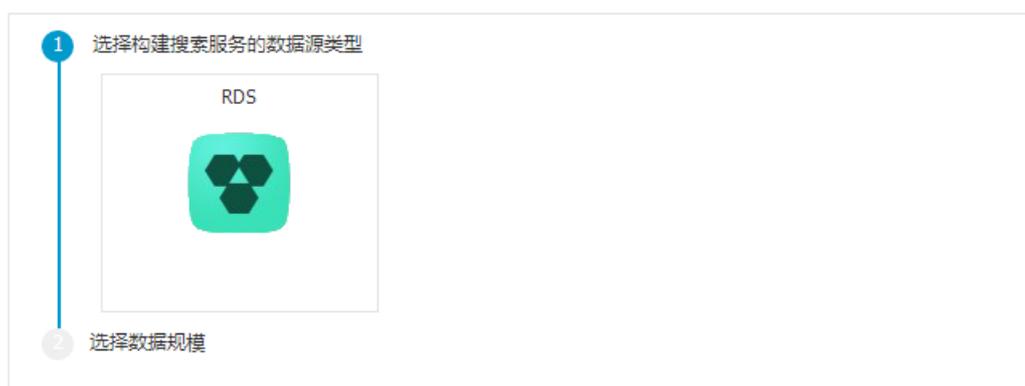
## 4.1 索引基本信息

索引基本信息页面主要包含了索引的基本信息和索引数据源信息。

如果该应用的数据源地址发生了变更，可以点击“修改数据源”，更换数据源，将会重新进行索引创建，具体流程可以参考索引创建流程（从 1.2 开始）。

| 当前索引：just\_10000

① 索引命名 → ② 选择数据类型和规模 → ③ 选择数据源 → ④ 定义元数据 → ⑤ 确认



## 4.2 索引查询

点击索引查询选项卡，进入索引查询页面：

索引名: search4\_53\_just\_10000

返回索引列表

索引基本信息

索引查询

元数据管理

配额管理

日志历史信息

数据统计

同步数据



索引查询允许用户自定义查询语句，对索引进行查询操作

点击索引查询页面按钮，打开查询页面

此外支持3种高级查询功能:统计查询(facet) 分组查询(group) 高亮查询(hl)，同时支持选列功能

索引查询

打开索引查询页面

点击“打开索引查询页面”按钮，可以进入索引查询页面，用户可以使用索引查询页面进行测试查询。查询页面支持任意组合的查询语句查询，还提供统计查询(facet)、分组查询(group)和高亮查询(highlight)三种高级查询，详细的查询语法，可以参考《TIS 查询手册》。

TIS查询语句 [了解TIS查询语法](#)

\*:\*

查询

高级查询：

facet查询 [?](#) group查询 [?](#) 高亮查询 [?](#) debug查询:  选择目标列: [选择](#)

rows 3 sort eg: "create\_time desc" fq eg: "id: [5 TO 12]"

结果条数:11505

```
auction_id:8192000 auction_point:0 auction_status:0 auction_type:b category:50008438 city:上海 ends:20140101000001 feature_cc:5 features;tags:41922; gmt_modified:20141030122312 incrementnum:0.0 inventory_id:110000123456781000 item_dim:0 old_quantity:0 old_starts:20140101000001 options:536870912 ordercost:0 outer_id:8192000 point_price:1 postage_id:0 promoted_status:1 property:0 prov:上海 quantity:20020 reserve_price:110.0 secure_trade_fast_post_fee:0.0 secure_trade_ordinary_post_fee:0.0 sold_quantity:0 spu_id:0 starts:20140101000001 stuff_status:0 sync_version:1 title:测试1万条数据 user_id:2065211312 vertical_market:0
```

### 4.3 元数据管理

元数据管理页面主要是在需要对元数据进行相关修改，或数据源字段发生变更，需要同步时使用。

同步数据源列信息									
字段名称	字段类型	主键	默认值	可搜索	可展示	区间查询	分词方法	操作	
1 id	int	x	+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	选择分词方法	<a href="#">删除</a>	
2 price	double	x	+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	选择分词方法	<a href="#">删除</a>	
3 perzun	int	x	+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	选择分词方法	<a href="#">删除</a>	
4 companyid	int	x	+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	选择分词方法	<a href="#">删除</a>	
5 totalsun	long	x	+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	选择分词方法	<a href="#">删除</a>	

添加

数据分词键: id uniqueKey: id 修改备注: 修改索引备注

[修改schema并重新构建索引](#)

元数据管理页面允许管理本对应的元数据信息，可以在本页面执行以下三种操作：

(1) 修改任何一个字段的任意属性；

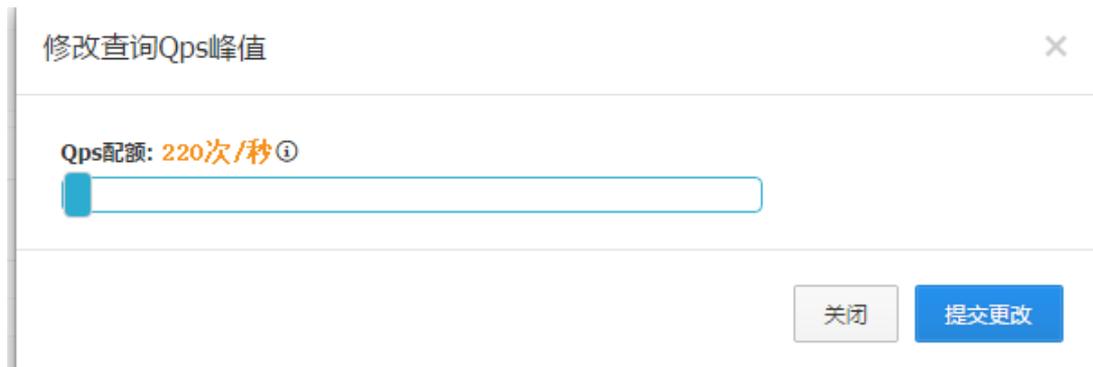
(2) 点击同步数据源列信息，将同步数据源新增或删除的列信息；

(3) 点击添加按钮增加一个自定义的列。

以上三种操作，只有当你点击修改元数据并重新构建索引后才会生效。生效意味着：本次修改将替换之前的索引内容，同时服务端将重新构建索引,新索引会覆盖老的索引。

#### 4.4 配额管理

该页面主要包含了索引创建过程中相关配额参数设置，如果业务场景有变更或数据规模以及访问量增加，当前的配额已经不能满足实际需求，可以通过这个页面进行相关修改。点击修改，进行相关操作。



#### 4.5 日志历史信息

日志历史信息页面主要是进行日志信息的管理，当 2.1 中点击激活按钮，弹出实时日志窗口后，关掉实时日志窗口，可以通过这个页面查看相关日志信息。

##### 日志历史信息

日志列表						
任务编号	任务类型	触发时间	状态	错误信息	日志详细信息	查看实时日志
713	索引构建	2014/12/30 23:29:58	成功		<a href="#">查看详细信息</a>	<a href="#">实时日志</a>

状态栏如果是正在进行，表示索引正在创建，可以点击“查看详细信息”或者“实时日志”，查看实时日志：

##### 任务编号为 713 的日志信息

时间	来源	等级	内容摘要
2014/12/30 23:30:01	tgrsv (10.176.126.67)	INFO	indexname:search4_44_TIS_testget json from tis taskid:713 {"appName":"search4_4
2014/12/30 23:30:01	tgrsv (10.176.126.67)	INFO	start execute,type:create
2014/12/30 23:30:01	share (10.143.32.153)	INFO	Schema analysis Success !
2014/12/30 23:30:01	share (10.143.32.153)	INFO	get sharedKey Success, sharedKey is id
2014/12/30 23:30:01	share (10.143.32.153)	WARN	start Create ODPS Table...
2014/12/30 23:30:01	share (10.143.32.153)	WARN	ODPS statement is create table if not exists search4_44_TIS_test_trade_order_a5b

状态栏如果是失败，将会有错误信息抛出，也可以通过“查看详细信息”定位具体错误原因。

状态栏如果是成功，表示此次索引构建成功，可以进行相关查询操作，构建过程的日志，也可以通过“查看详细信息”进行查看。

#### 4.6 数据统计页面

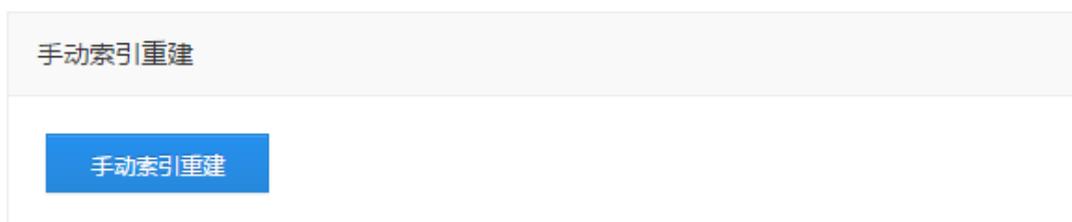
数据统计页面主要是进行索引数据的统计信息，提供多个时间维度的统计：



主要统计索引的搜索次数(PV)和文档数，可以根据自己的需要进行相关的统计。

#### 4.7 同步数据页面

当源数据每天有大量增量数据时，索引同步会导致出现索引碎片，影响查询性能；或者源数据发生了字段或者其它变更时，需要时同步数据页面。页面主要包括：定时索引重建，手动索引重建和索引构建历史。源数据发生了字段或者其它变更时，需要选择手动创建，直接点击“手动索引重建”，将会重新创建一次索引，弹出实时日志窗口。



当源数据每天有大量增量数据时，索引同步会导致出现索引碎片，影响查询性能，这种情况可以选择定时重建索引，设置定时任务，比如每天 0 点 50 分构建一次全量索引：



页面上将会生成定时任务，在每天 00:50 执行一次全量任务。

## 定时索引重建

设置任务计划： 每日 00:50 执行索引重建

定时任务以及手动重建任务的日志历史都可以在索引构建历史下查看：

任务编号	任务类型	触发时间	状态	错误信息	日志详细信息	查看实时日志
717	索引构建	2014/12/31 00:50:00	成功		<a href="#">查看详情信息</a>	<a href="#">实时日志</a>
716	索引构建	2014/12/31 00:50:00	正在进行中		<a href="#">查看详情信息</a>	<a href="#">实时日志</a>
713	索引构建	2014/12/30 23:29:58	成功		<a href="#">查看详情信息</a>	<a href="#">实时日志</a>

## 5. 查询客户端

### 5.1 TOP 调用权限申请

如果你是百川的应用，默认给你开通了 tis 的 top 调用权限，不需要去申请，如果你属于优站、爱淘宝、特色中国、店铺模块后台、互动应用后台等业务需要在 <http://my.open.taobao.com/> 登陆账号，选择你的应用，在证书权限管理页面，申请" TIS 终端索引查询服务"的增值权限。如下图所示：

### 申请增值包

注意：在申请之前，请仔细阅读应用权限审核规则

增值包名称： [查看该增值包详情](#)

应用功能描述：  
请详细描述应用对增值权限的业务需求与使用场景，这将涉及到开放平台是否发放给您所需要的权限

申请理由：

[申请开通](#)

## 无线开放百川接口

申请使用

### 申请条件前置规则 ?

规则名称	是否必须
top运营审核	是

### API列表

基础	taobao.itempropvalues.get	收费	获取标准类目属性值
基础	taobao.sellercats.list.get	收费	获取前台展示的店铺内卖家自定义商品类目
基础	taobao.shop.get	收费	获取卖家店铺的基本信息
基础	taobao.itemprops.get	收费	获取标准商品类目属性
基础	taobao.itemcats.get	收费	获取后台供卖家发布商品的标准商品类目
增值	taobao.tae.item.detail.get	免费	商品详情数据接口
增值	taobao.tae.items.select	免费	商品选品服务
增值	taobao.tae.items.list	免费	商品列表服务
增值	taobao.tis.query	免费	TIS终端索引查询服务
增值	taobao.tae.exist.shop.coupon.get	免费	判断是否存在店铺优惠券

## 5.2 SDK 下载

进入 my.open.taobao.com，在应用里面，点击 SDK 下载，找到 Taobao App Engine SDK，根据你的对应的编程语言选择相应的版本下载：



请首先在 tae 平台下载 SDK，然后编写查询代码：

## 5.3 Java 调用示例

示例：

```
import com.taobao.api.ApiException;
import com.taobao.api.DefaultTaobaoClient;
import com.taobao.api.TaobaoClient;
import com.taobao.api.request.TisQueryRequest;
import com.taobao.api.response.TisQueryResponse;
```

```

public class TaeTest {
    private final static String url="http://gw.api.taobao.com/router/rest?";
    private final static String appkey="****"; //当前账号申请的 Appkey,请填写
    private final static String secret="****"; //当前账号申请的 AppSecret , 请填写
    public static void main(String[] args) {
        TaobaoClient client = new DefaultTaobaoClient(url, appkey, secret);
        TisQueryRequest req = new TisQueryRequest ();
        req.setServiceName("search4_53_just_10000"); //索引名称, 在 TIS 里面的创建的名
称进行访问

        req.setQueryStr("q=*.*"); // 业务查询参数,根据业务场景可自行拼装
        try {
            TisQueryResponse response = client.execute(req);
            System.out.println(response.getQueryResult());
        } catch (ApiException e) {
            e.printStackTrace();
        }
    }
}

```

#### 5.4 PHP 调用示例 :

```

$c = new TopClient;
$c->appkey = appkey;
$c->secretKey = secret;
$req = new TisQueryRequest;
$req->setServiceName("search4wjb"); //索引名称
$req->setQueryStr("q=title:xxx&routeValue=seller_id:4&start=0&rows=20"); //业务查
询参数,根据商家业务场景自行拼装
$resp = $c->execute($req);

```

## 缓存服务

### 缓存服务简介

TAE 缓存服务采用阿里云开放缓存服务 ( Open Cache Service , 简称 OCS ) , 提供基于内存的缓存服务 , 支持海量小数据的高速访问。OCS 可以极大缓解对后端存储的压力, 提高网站或应用的响应速度。OCS 支持 Key-Value 的数据结构, 兼容 Memcached 协议的客户端都可与 OCS 通信。

### 缓存服务 API

JAE 的 PHP 引擎中内嵌了文件存储对象 \$cacheService , 可直接调用。

#### 1、set(key,value,expireTime) 方法

设置缓存, 参数含义分别为 : key , value , key 的过期时间

/\*\* key 可以为 Map 类型数组 \*\*/

```
$array = array("name" => "心远", "address" => "阿里巴巴");  
$cacheService->set("心远", $array, 10);
```

## 2、get(key) 方法

根据 key 获取缓存值

```
$getResultExpire = $cacheService->get("心远");
```

## 3、mget(keys) 方法

根据 key 批量获取缓存

```
$keys= array('123456', '234567', '345678');  
$cacheResult= $cacheService->mget($keys);//一次批量获取
```

## 4、delete(key) 方法

根据 key 删除缓存值

```
$getResultExpire = $cacheService->delete("心远");
```

# 文件存储服务

## 文件存储服务简介

TAE 文件存储服务采用阿里云开放存储服务 ( Open Storage Service , 简称 OSS ) 来支持任意数据类型的存储, 支持任意时间、地点的数据上传和下载。OSS 中每个存储对象 ( object ) 由名称、内容、描述三部分组成。通过 OSS 可以获得更稳定更安全的存储服务, 使开发大容量和高并发系统变得更容易。

## 文件存储服务 API

PHP 引擎中内嵌了文件存储对象 \$fileStoreService, 可直接调用。

1. saveTextFile(fileContent, fileName) 方法, 存储文本文件

```
$saveTextFileResult = $fileStoreService->saveTextFile("xinyuan", "/services/tfs/xinyuan.txt");
```

2. getFileText(fileName) 方法, 获取文件内容

```
$getFileTextResult = $fileStoreService->getFileText("/services/tfs/xinyuan.txt");
```

3. isFileExist(fileName) 方法, 判断文件是否存在

```
$isFileExistResult = $fileStoreService->isFileExist("/services/tfs/xinyuan.txt");
```

4. deleteFile(fileName) 方法, 删除文件

```
$deleteFileTextResult = $fileStoreService->deleteFile("/services/tfs/xinyuan.txt");
```

## 文件存储服务使用示例

```
getFileTextResult = $fileStoreService->getFileText("/services/tfs/filename.txt");  
echo "getFileText:" . $getFileTextResult;
```

```
$fileStoreService->saveTextFile("filecontent", "/services/tfs/filename.txt");
```

```
$getFileTextResult = $fileStoreService->getFileText("/services/tfs/filename.txt");  
echo "getFileText:". $getFileTextResult;
```

```
$isFileExistResult = $fileStoreService->isFileExist("/services/tfs/filename.txt");  
echo "isFileExist:". $isFileExistResult;  
  
$fileStoreService->deleteFile("/services/tfs/filename.txt");  
  
$getFileTextAfterDeleteResult = $fileStoreService->getFileText("/services/tfs/filename.txt");  
echo "getFileText:". $getFileTextAfterDeleteResult;  
  
$fileStoreService->createDir("/services/tfs/newdir");  
  
$fileStoreService->deleteDir("/services/tfs/newdir");
```

## FetchUrl 服务

### FetchUrl 服务简介

TAE 中 FetchUrl 服务提供基于 http 协议的网页抓取。目前 TAE 系统中有全局 FetchUrl 的白名单。如果期望的 URL 不在全局白名单中,可自主申请 App 粒度的 URL 白名单。值得注意的是,HTTP 请求必须使用 FetchUrl,之外的类似 CURL 之类的 API 不被允许。

### FetchUrl 服务 API

PHP 引擎中内嵌了 fetchurl 服务对象 \$fetchService,可直接调用。

1.fetchByGet(url) 方法,通过 GET 方式获取网页内容。

```
$fetchGetTBResult = $fetchService->fetchByGet("http://www.taobao.com");
```

2.post(url,param) 方法,通过 POST 方式获取网页内容

```
$parameter = array("username" => "jae", "password" => "jae");
```

```
$fetchPostResult=$fetchService->post("https://login.taobao.com/member/login.jhtml",  
$parameter);
```

### FetchUrl 服务域名限制

在默认情况下,Fetchurl 服务没有开启对任何域名的访问。如果读者需要使用 FetchUrl 服务访问外部域名,需要通过旺旺联系梓晗,反馈哪个应用访问了哪个域名,只要是合理的就会配置到对应应用的白名单里面。

### FetchUrl 服务使用示例

```
echo '$fetchService->fetchByGet 方法实例: ';  
$url = "http://www.taobao.com";  
$content=$fetchService->fetchByGet($url);  
if($content) {  
    echo '调用$url: '$url . '的远程内容为: '$content;  
}
```

## 日志服务

## 日志服务简介

TAE 日志服务提供了应用标准日志记录功能，接入该功能后，可以通过 workstation 控制台查看应用自己输出的日志，便于开发调试和问题定位。

## 实时日志使用说明

内置 \$appLog 对象的三个方法，通过 info、warn、error 三个方法指定不同的日志输出等级，这里输出的日志属于应用日志，可以在 workstation 中查看。

```
1 $appLog->info("info-log-emssage");
2 $appLog->warn("warn-log-emssage");
3 $appLog->error("error-log-emssage");
```

## 实时日志日志查看

本地日志

本地 SDK 环境的日志在目录 logs/applogs 下。

线上日志

线上日志，请在 WorkStation 中进行查看 [参考链接](#)

## 打点监控埋点服务 API 方法

```
$response=$monitorService->invoke("Student","getName",2132141,true);
```

```
$response=$monitorService->invoke("Student","getSex",500,false);
```

应用名称	调用次数	失败次数	响应时间(RT)	成功率	QPS	超耗时
Student	2	1	1066320.5 ms	50.00 %	0	选择
getSex	1	0	500	100 %	0	选择
getName	1	1	2132141	0 %	0	选择
qinning	0	0	0 ms	0.00 %	0	选择

## 追踪日志服务 API

### 1.开始代码段

```
$monitorService->startCodeBlock("testNameBlock01", "Enter code block01!");
```

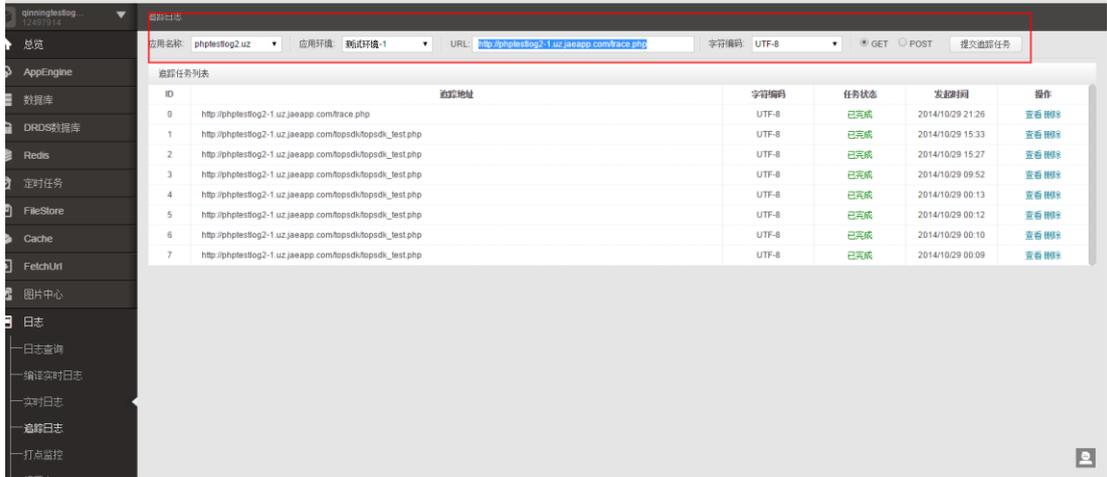
### 2.中间打信息代码段

```
$monitorService->log("testNameBlock01Inside", "Oh! I'm in code block01!");
```

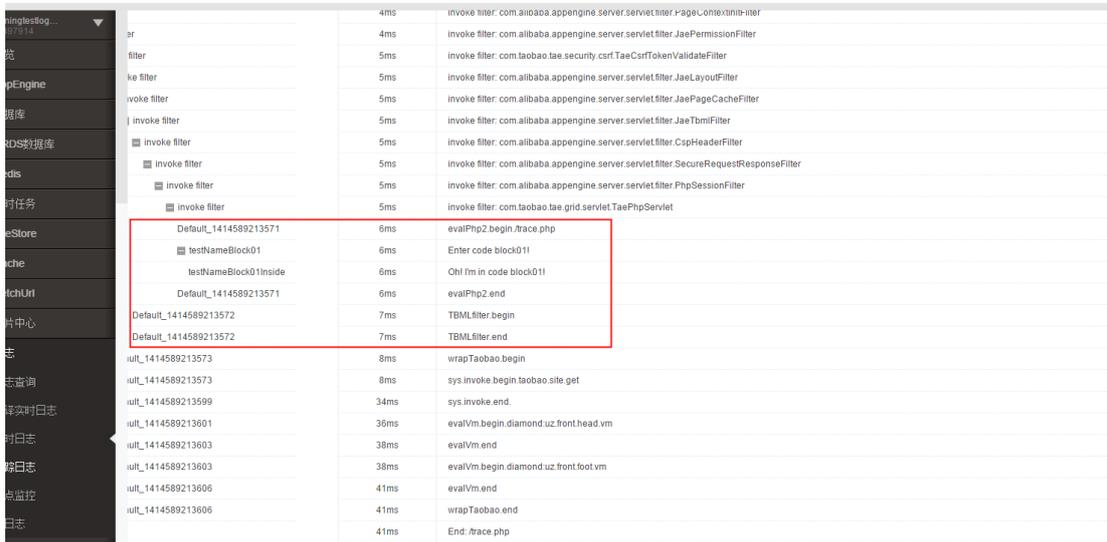
### 3.结束代码段

```
$monitorService->endCodeBlock();
```

## 创建追踪日志



效果图：



## 定时任务服务

### 定时任务介绍

定时任务是指我们定时触发用户写的 php 代码的执行，来完成需要周期性处理的后台任务

### 调度任务开发指南

接入 TAE 的任务模型，ISV 需要把任务的业务逻辑拆解为两个阶段：数据准备阶段及数据处理阶段  
TAE 任务框架负责回调这两阶段的代码

### 定时服务使用方法

开发者需要在 TAE 控制台配置任务服务，入口如下：



## 如何设置一个定时任务

### 1. 创建新任务

通过定时任务入口进入创建页面



### 创建新任务

## 新建任务

任务名称：

调度表达式：  
 [关于表达式？](#)

任务描述：

新建一个任务需要提供三个信息，任务名称、任务表达式及任务描述，任务名称与任务描述由开发者自由定义；

任务表达式必须满足的 Quartz 的规则，完整的的表达式说明可以参考：[Quartz 说明](#)

以每天 2 点定时执行会员增量同步的案例来看，表达式应该如下设置：

```
0 2 * * ?
```

上述表达式就说明该任务是在每个月每天的 2 点 0 分开始执行该任务。

Quartz 表达式说明：

第 1 列表示分，取值 0~59 每分钟用\*或者 \*/1 表示

第 2 列表示小时，取值 0~23 (0 表示 0 点)

第 3 列表示日期，取值 0~31

第 4 列表示月份，取值 0~11 或 JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV

第 5 列表示星期几，取值 1~7(1=Sunday)或 SUN, MON, TUE, WED, THU, FRI,SAT

符号简介：

"\*"代表取值范围内的数字

"/"代表"每"

"-"代表从某个数字到某个数字

","分开几个离散的数字

"?"用来表示没有意义的值，日期列和星期列中，必须有一个为?

## 2. 上传调度任务

应用创建完毕后，将已打好包的调度任务程序包上传到对应的任务上；



**需要注意:**调用任务程序包必须打包为 ZIP 格式，ZIP 包包括 app.yml 文件和 php 文件

**app.yml 文件 demo：**

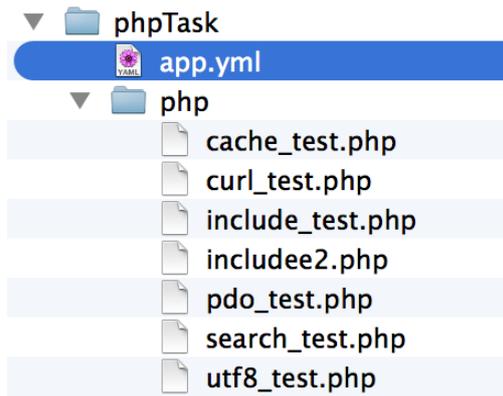
```
!!com.alibaba.appengine.jae.model.TaskConfig!!
```

```
taskFile: php/task.php
```

```
fileEncoding: UTF-8
```

其中 taskFile 表示要定时执行的 php 文件，php?文件的路径使?用 app.yml 的相对路径

以下面的目录结构为例，打成 zip 包的时候不要包含 phpTask 这个文件夹



### 网络访问：

在定时任务中可以使用同一个 appkey 的所有服务，使用方法同 web 应用。可靠的网络访问方法只有 Fetchurl，不能直接访问。

### 本地开发环境

#### 下载

下载后修改 service-conf/service-config.properties 文件，服务配置格式和 web 应用的 ace4j-private.properties 格式一样

执行 run.bat(linux 下面是执行 run.sh)，可以启动一次任务执行，会提示你让你输入任务所在的目录。

如果定时任务的代码所在目录是不变，不想每次输入可以修改 run.bat，把你的定时任务所在的位置写在这里这个双引号中：set WORKER\_BASE=""

### 定时任务 demo

#### 定时任务 demo 下载

## 图片服务

### 界面方式

在控制台选中图片中心，打开如下界面，包括：图片目录、上传图片、新建文件夹、图片查看模式和图片管理等功能，图片中心容量：500G。



### 图片上传功能



点击“修改位置”，可以修改放置图片的文件夹。



双击图片，可以查看图片详情，左侧支持多尺寸图片。



新建文件夹



文件查看，可以选择只查看图片；图片展示模式可以选择列表模式或者大图模式。



代码方式

PicService

无需显示引入 picservice，容器会自动引入，开发者直接使用即可

/\*\*

- \* 上传图片,指定图片全路径,会自动创建相应目录,如果存在同名图片会报错
- \* 图片大小限制是 3M
- \* @param dirpath 目录路径,例如: "/a/b/" 可以为空,如果为空表示根目录

```

* @param filename 文件名称，例如："c.jpg" 可以为空，如果为空系统生成随机文件名
* @param content 文件内容
* @return
*/
$response=$picService->savePic($dirpath,$filename,$content);
//调用 response 对象的 isSuccess()判定上传是否成功。
//调用 response 对象的 getStatus()得到上传的状态码。
//调用 response 对象的 getResult()得到 Img 类型的图片信息，包含图片的名字,图片路径，URL 等。
/**
* 根据图片全路径获取图片信息，如果目录不存在或图片不存在，返回 response 的 success 为 false
* @param $dirpath 图片全路径
* @param filename 文件名称
* @return
*/

$response=$picService->getPic($dirpath,$filename);
/**
* 根据图片全路径删除图片,如果目录不存在或图片不存在，返回 response 的 success 为 false
* @param $dirpath 图片全路径
* @param filename 文件名称
* @return
*/
$response=$picService->deletePic($dirpath,$filename);
/**
* 获取某个目录路径获取图片列表，如果目录不存在、返回 response 的 success 为 false
*
* @param $dirpath 目录路径
* @return
*/

$response=$picService->getPicList($dirpath)
/**
* 翻页查询：根据目录 id 获取图片列表，如果目录不存在、返回 response 的 success 为 false
* @param $dirpath 目录路径 例如 "/a/b/" ,为空表示根目录
* @param $currentPage 页码 从 1 开始
* @param $pageSize 每页记录数 最小 1 最大 100
* @return
*/
$response=$picService->getPicList($dirpath, $currentPage,$pageSize)
/**
* 删除目录，非空目录不能删除，不能删除根目录
*
* @param $dirpath 图片路径
* @return

```

```

*/

$response=$picService->deleteDir($dirpath);

/**
 * 创建目录
 *
 * @param $dirpath 图片全路径，例如："/a/b/"，不能为空
 * @return
 */

$response=$picService->createDir($dirpath);
/**
 * 获取某个目录的子目录列表，如果目录不存在，返回 response 的 success 为 false
 *
 * @param dirpath 目录路径 例如 "/a/b/"，为空表示根目录
 * @return
 */

$response=$picService->getDirList($dirpath)
/**
 * 判断文件是否存在
 *
 * @param $dirpath 目录，为空表示根目录
 * @param $filename 文件名称
 * @return
 */

$response=$picService->isFileExist($dirpath,$filename);
/**
 * 判断目录是否存在
 *
 * @param $path 文件全路径
 * @return
 */

$response=$picService->isDirExist($dirpath);

```

#### 返回值类型：

```

public class Response {
    private Integer status = 200; //状态码
    private boolean success = false; //方法调用是否成功、是否按预期执行
    private Img result; //返回值，调用成功才有返回值

```

```
private String errorMsg; //调用失败有错误信息  
}
```

状态码列表:

200:请求成功

400:参数错误

404:图片不存在

405:目录不存在

408:图片大小超限

406:目录非空不能删除

406:存在同名目录或文件

408:请求超时

500:系统错误

### Pic 类型

```
public class Pic{  
    /**  
    * 文件名  
    */  
  
    public String filename;  
    /**  
    * 所属目录  
    */  
  
    public String dirpath;  
    /**  
    * 文件大小  
    */  
  
    public Long size;  
    /**  
    * 文件类型,0:jpg,1:gif,2:png,3:bmp  
    */  
  
    public Integer type;  
    /**  
    * 文件内容修改时间  
    */  
  
    public Date fileModified;  
    /**  
    * 删除标识, 0 代表正常, 1 代表已经删除, 3 代表在回收站中  
    */  
}
```

```

public Integer deleted;
/**
 * md5 加密
 */

public String md5;
/**
 * 图片绝对地址, 使用这个地址, 可直接访问图片.
 * 例
如:http://img03.taobaocdn.com/imgextra/i3/436168931/T2Q1UWXFpaXXXXXXXXX_!436168931-
2-sitemanager.png
 */

public String fullUrl;
/**
 * 获取不同尺寸的图片绝对地址
 * 例如 60x60 :
http://img03.taobaocdn.com/imgextra/i3/436168931/T2Q1UWXFpaXXXXXXXXX_!436168931-2-s
itemanager.png_60x60.jpg
 */

public String getCDNFullUrl($width,$height);
}

```

## demo 下载

[图片上传 demo 下载](#)

## 消息通知

### 消息通知简介

为了保证用户的体验闭环, 我们实现了针对百川渠道产生的订单列表的消息推送服务, 消息推送服务可以让交易数据通过 TOP-TMC 实时通知到 ISV, 帮助引导成交应用的 ISV 做售后服务, 个性化营销推荐等, 同时也可以让消费者在对应 APP 渠道查看自己的订单列表。

在使用上, 百川项目的第三方 app 可以直接通过 top 端提供的消息服务进行消息订阅, 从而获取订单基本信息 ( 比如主订单 Id、订单状态、购买者混淆 nick、宝贝标题、宝贝 id、宝贝图片以及实际付款等 )。

### 前置条件

- 1、需要客户端 SDK 支持 App 引导成交订单的识别, 所以请将 SDK 升级到某个版本以上;
- 2、通过 TOP 获取的 tae 的 H5 官方下单页面

### 订单相关的消息

#### 1、taobao\_tae\_BaichuanTradeCreated

创建订单消息(下单未付款), 订单状态(order\_status)是 7

## 2、taobao\_tae\_BaichuanTradeSuccess

交易成功消息(确认收货后), 订单状态(order\_status)是 6

## 3、taobao\_tae\_BaichuanTradeRefundSuccess

退款成功, 没有订单状态

## 4、taobao\_tae\_BaichuanTradePaidDone

付款成功(下单已付款), 订单状态(order\_status)是 2

## 5、taobao\_tae\_BaichuanTradeClosed

交易关闭(包括退款后交易关闭和创建订单后交易关闭), order\_status 为 4 是退款后交易关闭, order\_status 为 8 是创建订单后交易关闭

## 6、taobao\_tae\_BaichuanAuctionChange

商品更新通知服务 [详细](#)

### 使用说明

消息服务使用完整介绍: <http://open.taobao.com/doc/detail.htm?id=101663>

### 首先申请权限

目前跟订单推送相关的权限包都位于“无线开放百川消息包”增值权限包内,要订阅订单推送相关的消息, 首先请在[开放平台应用管理后台](#)申请该增值权限包。

应用 / 无线开放百川消息包

订单推送服务产品试用  
appkey: 23025814  
应用概况  
应用设置  
**证书权限管理**  
SDK下载  
安全中心  
报表中心  
消息服务

### APP 证书

App Key: ⓘ	
App Secret: ⓘ	<a href="#">查看 / 重置</a>
应用安全等级: ⓘ	2级
证书流量: ⓘ	5000次 / 天
基础开放权限:	<a href="#">物流宝</a> <a href="#">卖家基础开放</a> <a href="#">系统工具</a> <a href="#">酒店</a> <a href="#">分销</a> <a href="#">消息服务</a>
增值权限: ⓘ	<a href="#">聚石塔数据推送</a> <a href="#">淘点外卖业务</a> <a href="#">TAE订单推送服务</a>



### 订阅淘宝消息

在[开放平台应用管理后台](#)如下页面订阅消息：  
可以在分类中选择“导购平台”快速筛选消息



点击消息名称，就能查看具体的返回信息字段



### 代码实现接收消息

正式环境请求地址：[ws://mc.api.taobao.com/](http://ws://mc.api.taobao.com/)

沙箱环境请求地址：[ws://mc.api.tbsandbox.com/](http://ws://mc.api.tbsandbox.com/)

消息服务相关 api：

[http://open.taobao.com/api/api\\_cat\\_detail.htm?spm=0.0.0.0.V0oi2c&cat\\_id=20439&category\\_id=102](http://open.taobao.com/api/api_cat_detail.htm?spm=0.0.0.0.V0oi2c&cat_id=20439&category_id=102)

1、为用户开通消息服务

百川无线应用类型可以不需要此步骤；

2、接收消息，实现方式有两种：[通过 SDK 接收消息](#)、[通过 API 接收消息](#)

## 通过 SDK 接收消息

目前支持 JAVA 语言，其它语言建议采用 API 接收消息。通过 SDK 接收消息只需要关注业务的处理，不需要操心消息重发、确认、长连接的重连等操作，SDK 会自动处理好一切。

### Java 接口说明

```
public interface MessageHandler {  
/**  
*消息服务客户端收到消息后，会回调该方法处理具体的业务，处理结果可以通过以下两种方式来表述：  
<li>抛出异常或设置 status.fail()表明消息处理失败，需要消息服务端重发  
<li>不抛出异常，也没有设置 status 信息，则表明消息处理成功，消息服务端不会再投递此消息  
@param message 消息内容  
@param status 处理结果，如果调用 status.fail()，消息服务将会择机重发消息；否则，消息服务认为消息处理成功  
@throws Exception 消息处理失败，消息服务将会择机重发消息  
*/  
public void onMessage(Message message, MessageStatus status) throws Exception;  
}
```

### Java 代码：

```
TmcClient client=new TmcClient("app_key","app_secret","default");  
client.setMessageHandler(new MessageHandler(){  
public void onMessage(Message message, MessageStatus status){  
try{  
System.out.println(message.getContent());  
System.out.println(message.getTopic());  
//默认不抛出异常则认为消息处理成功  
}catch(Exception e){  
e.printStackTrace();  
status.fail();//消息处理失败回滚，服务端需要重发  
}  
}  
});  
client.connect();
```

**注：采用 Java main 方法在 Eclipse 里面运行上面的代码测试时，请在 client.connect()后面加上 Thread.sleep 让 main 线程等待一段时间结束，以便观察消息的实时接收情况，否则 main 线程结束后，TMC 长连接也会跟着断开。如果是在 web 服务器上运行上面的代码，则不用在 client.connect()后面加任何 Thread.sleep 代码，也不需要在外面包一层 while(true)循环，因为 web 服务器上的主线程只要服务器不关闭都是不会结束的，TMC 的长连接会一直保持。**

## 通过 API 接收消息

提供 API 接收消息的目的是那种对多线程和长连接处理不方便的语言使用的，比如 PHP、Python，这些语言官方暂时没有提供 SDK，可以通过下面两个 API 配合使用也能达到准实时接收和确认消息的目的。

首先：消费消息

API 名称：[taobao.tmc.messages.consume](#)

API 入参：

应用级输入参数					
名称	类型	是否必须	示例值	默认值	描述
group_name	String	可选	vip_user		用户分组名称，不传表示消费默认分组，如果应用没有设置用户分组，传入分组名称将会返回错误
quantity	Number	可选	100	100	每次批量消费消息的条数

说明：消息消费后，指针自动后移，下次调用自动获取到未消费过的消息，但是消费确认后的消息无法再次获取。

然后：确认消息

API 名称：[taobao.tmc.messages.confirm](#)

API 入参：

应用级输入参数					
名称	类型	是否必须	示例值	默认值	描述
group_name	String	可选	vip_user		分组名称，不传代表默认分组
s_message_ids	Number []	必须	123,456		处理成功的消息ID列表
f_message_ids	Number []	可选	234,567		处理失败的消息ID列表

说明：获取消息后，如果不确认，消息服务会选择时机重发，重发次数由消息服务控制，如果消息 7 天内都没有被确认将会被删除。

Java API 方式代码：

```
TaobaoClient client=new
DefaultTaobaoClient("http://gw.api.taobao.com/router/rest","app_key","app_secret","json");
do{
    longquantity=100L;
    TmcMessagesConsumeResponse rsp=null;
    do{
        TmcMessagesConsumeRequest req=new TmcMessagesConsumeRequest();
        req.setQuantity(quantity);
        req.setGroupName("default");
        rsp=client.execute(req);
        if(rsp.isSuccess()&&rsp.getMessages()!=null){
            for(TmcMessagemsg:rsp.getMessages()){
                //handlemessage
                System.out.println(msg.getContent());
                System.out.println(msg.getTopic());
                //confirmmessage
                TmcMessagesConfirmRequest cReq=newTmcMessagesConfirmRequest();
                cReq.setGroupName("default");
                cReq.setsMessageIds(String.valueOf(msg.getId()));
                TmcMessagesConfirmResponse cRsp=client.execute(cReq);
                System.out.println(cRsp.getBody());
            }
        }
    }
}
```

```
    }
    System.out.println(rsp.getBody());

    }while(rsp!=null&&rsp.isSuccess())&&rsp.getMessages()!=null&&rsp.getMessages().size()>=quantity);
    Thread.sleep(3000L);
    }while(true);
更多请查看消息服务完整介绍：http://open.taobao.com/doc/detail.htm?id=101663
```

## PHP API 方式

```
<?

include("tmcmessagesconsumerequest.php");

include("topclient.php");

include("TmcMessagesConfirmRequest.php");

//实例化 TopClient 类

$c=new TopClient;

$c->appkey="21749834";

$c->secretKey="83d21fd8f1ba9f150c774a568deef058";

$c->gatewayUrl="http://gw.api.taobao.com/router/rest";

$c->format="json";

//实例化具体 API 对应的 Request 类

$req=new TmcMessagesConsumeRequest;

$req->setQuantity(10);

$resp=$c->execute($req);

print_r($resp);

for($i=0;$i<count($resp->messages->tmc_message);$i++)

{
```

```

    $messages_ids.=$resp->messages->tmc_message[$i]->id.,"";

}

$messages_ids=substr($messages_ids,0,strlen($messages_ids)-1);

/*

*dosomething

*

**/

echo $messages_ids;

//确认消息

$req2=new TmcMessagesConfirmRequest;

$req2->setSMessagesIds($messages_ids);

$resp2=$c->execute($req2);

?>

```

### ISV 返回数据实例

付款成功：

```

{"buyer_id":"AAE8g3trABpL86adzOb7d948","extre":"","paid_fee":"0.02","order_id":"887450175487518","order_status":"2","auction_infos":[{"detail_order_id":"887450175497518","auction_id":"AAH3g3toABpL86adzOzgP5-G","auction_pict_url":"i1/TB1YZ_QGVXXXXcmXVXXXXXXXXXXXXX!!0-item_pic.jpg","auction_title":"0s- 女装 - 发货合约 8hrs- 测试宝贝 - 拍下无效","auction_amount":"2"};{"detail_order_id":"887450175507518","auction_id":"AAH3g3toABpL86adzOzgP5-G","auction_pict_url":"i1/TB1YZ_QGVXXXXcmXVXXXXXXXXXXXXX!!0-item_pic.jpg","auction_title":"0s-女装-发货合约 8hrs-测试宝贝-拍下无效","auction_amount":"2"}]}

```

创建订单：

```

{"buyer_id":"AAFDg3trABpL86adzOb7GpFt","extre":"","paid_fee":"1333332.00","order_id":"887450175256640","order_status":"7","auction_infos":[{"detail_order_id":"887450175256640","auction_id":"AAFAg3toABpL86adzO6Hh49u","auction_pict_url":"i1/263660307/TB2dM.OapXXXXaoXXXXX XXXXXX!!263660307.jpg","auction_title":"自动化测试宝贝不要改不要买","auction_amount":"2"}]}

```

交易成功：

```

{"buyer_id":"AAGwg3trABpL86adzObh0qil","extre":"","paid_fee":"503.88","order_id":"814255897503474","order_status":"6","auction_infos":[{"detail_order_id":"814255897513474","auction_id":"A

```

AGag3toABpL86adzO43mH-X","auction\_pict\_url":"i3/TB1bG\_fGpXXXXbwXpXXXXXXXXXXXXX!!0-item\_pic.jpg","auction\_title":" 软 妹 子 纯 白 钉 珠 毛 绒 上 衣 ", "auction\_amount":"2"}, {"detail\_order\_id":"814255897523474", "auction\_id":"AAEJg3toABpL86adzO9yUatl", "auction\_pict\_url":"i4/TB1S9kQGXXXXXUXXXXXXXXXXXXXX!!0-item\_pic.jpg", "auction\_title":"睫毛蕾丝衫", "auction\_amount":"2"}]}

退款成功：

{ "buyer\_id": "AAE8g3trABpL86adzOb7d948", "refund\_id": "39347310141875", "refund\_fee": "0.01", "extre": "{}", "oid": "891474117497518", "tid": "891474117517518", "seller\_nick": "c 测试账号 85" }

交易关闭：

{ "buyer\_id": "189753268", "extre": "", "paid\_fee": "1817.00", "order\_id": "876888661036832", "is\_eticket": false, "order\_status": "4", "auction\_infos": [{"auction\_id": "1", "detail\_order\_id": "876888661046832", "auction\_pict\_url": "i3/T1q2uAXftIXXXXXXXXX-50-50.png", "auction\_title": " 配 送 服 务 ", "auction\_amount": "13"}, {"auction\_id": "2", "detail\_order\_id": "876888661056832", "auction\_pict\_url": "i2/T1lvuAXghIXXXXXXXXX-50-50.png", "auction\_title": " 搬 楼 服 务 ", "auction\_amount": "13"}, {"auction\_id": "3", "detail\_order\_id": "876888661066832", "auction\_pict\_url": "i4/T1OLuAXc4IXXXXXXXXX-50-50.png", "auction\_title": " 安 装 服 务 ", "auction\_amount": "13"}, {"auction\_id": "36391354501", "detail\_order\_id": "876888661076832", "auction\_pict\_url": "i1/TB1.xFGFVXXXaLaXXXXXXXXXXXXX!!0-item\_pic.jpg", "auction\_title": "翡戈家具客厅简约现代全橡木实木功夫茶几大理石茶几配电视柜包邮", "auction\_amount": "13"}]}

**返回值：** buyer\_id、paid\_fee、order\_id、order\_status、auction\_id(商品混淆 ID)、auction\_title、auction\_pict\_url、topic、detail\_order\_id(子订单号)、auction\_amount(商品数量)

**注意：** 其中商品图片需要拼接上 CDN 地址：<http://img02.taobaocdn.com/bao/uploaded/>

### 商品更新推送服务简介

商品信息更新通知服务主要用于当商品信息发生变更时，通过 TMC 消息通道通知给订阅此消息的 ISV。由于商品每个字段的变更源于不同的消息源头和时间，所以商品信息变更的消息内，即可能包含所有相关的变更字段，也可能只包含某一个字段。

例如商品价格和状态同时变更，接收到的消息，可能是一条：

```
content:{"auction_change_info":{"itemStatus":"-1","itemPrice":"99900"},"open_iid":"AAG2vgyDV4MXr_m2Lly9mzGD"}
```

也可能是两条分别包含状态和价格的消息：

```
content:{"auction_change_info":{"itemStatus":"-1"},"open_iid":"AAG2vgyDV4MXr_m2Lly9mzGD"};content:{"auction_change_info":{"itemPrice":"99900"},"open_iid":"AAG2vgyDV4MXr_m2Lly9mzGD"};
```

### 前置条件

- 1、一期通过 TOP 接口 ( taobao.iselect.tae.itempool.change ) 添加了相关商品变更的监控；
- 2、直接在 TAE 选品中心通过商品池和标签进行勾选；

### 订单相关的消息

- 1、taobao\_tae\_BaichuanAuctionChange 商品变更消息

### 使用说明

消息服务使用完整介绍：<http://open.taobao.com/doc/detail.htm?id=101663>

## 首先申请权限

目前跟商品变更消息推送相关的权限包都位于“无线开放百川消息包”增值权限包内，要订阅商品变更相关的消息，首先请在[开放平台应用管理后台](#)申请该增值权限包。



## 订阅 From 淘宝消息

在[开放平台应用管理后台](#)如下页面订阅消息：  
可以在分类中选择“导购平台”快速筛选消息



点击消息名称，就能查看具体的返回信息字段。



代码实现详见消息推送的代码实现

### ISV 返回数据实例

```
content:{"auction_change_info":{"itemStatus":"-1"},"open_iid":"AAG2vgyDV4MXr_m2Lly9mzGD"}
```

```
content:{"auction_change_info":{"itemPrice":"12000"},"open_iid":"AAG2vgyDV4MXr_m2Lly9mzGD"}
```

topic:taobao\_tae\_BaichuanAuctionChange

其中, auction\_change\_info 对应的是一个由 Map 转换为的 json 串, key:itemStatus 表示 商品状态, 对应的 value 有两个: -1 (商品状态不可用) 和 1 (商品状态可用)

key:itemPrice 表示商品的最优惠价格(不同身份的买家看到的价格可能不同, 这里是最优惠的一种价格),value 是变更后的价格,单位分

目前 key 包含 itemStatus 和 itemPrice, 后续会继续添加

## Redis 服务

### 简介

redis 是一个 key-value 存储系统。和 Memcached 类似,它支持存储的 value 类型相对更多,包括 string(字符串)、list(链表)、set(集合)、zset(sorted set --有序集合)和 hash (哈希类型)。这些数据类型都支持 push/pop、add/remove 及取交集并集和差集及更丰富的操作,而且这些操作都是原子性的。在此基础上,redis 支持各种不同方式的排序。与 memcached 一样,为了保证效率,数据都是缓存在内存中。区

别的是 redis 会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了 master-slave(主从)同步。

## 代码实现

如果你只有一个 redis 服务，在你调用 redis 的服务的时候我们会将此服务作为你的默认的 redis 实例调用，调用代码如下：

```
<?php
$redis = $redisService->getRedisClient();
$redis->set("k","v001_in_php");
echo $redis->get("k");
echo "<br>";
?>
```

如果你开通有多个 redis 服务，在调用的时候必须传入你创建时候的 redis 的服务的实例名字，比如，你创建的一个 redis 服务的名字为 test，调用代码如下：

```
<?php
$redisService = $RedisServiceFactory->getRedisService("test");
$redis2 = $redisService->getRedisClient();
$redis2->set("k2","v001_in_php");
echo $redis2->get("k2");
echo "<br>";
?>
```

## 异步任务服务

### 异步任务服务简介

异步任务允许 ISV 为耗时程序开辟一个新线程进行处理，处理完时返回。

### 异步任务服务 API

1、submit\_task 方法，提交一个 PHP 函数作为异步任务。

```
$taskId = submit_task(mytask);
```

2、lookup\_task 方法，查询任务执行结果。

```
$taskresult = lookup_task($taskId);
```

执行结果包括 status、errmsg 和 result：

**a."status": 执行状态，分为 success | fail | queued | started | unknow。**

success 时才会有 result；

fail 时才会有 errmsg；

queued 表示任务已提交到队列，尚未执行，排队中；

started 表示任务已开始执行，尚未执行完成；

unknow 可能有 3 种情况：

- 任务没提交过
- 任务执行完成，但是执行结果被挤出了 lru 内存
- 任务提交后，vm 重启够，状态数据丢失。任务是否执行、完成、出错未知。

**b."errmsg": 状态为 fail 时的错误信息。**

### c."result": 状态为 success 时的执行结果。

任务执行情况会输出到 log 中，可通过日志系统查看。

### 异步任务服务注意点

- 1、不支持持久化，APP 重启后异步任务信息丢失，开发者得自己实现持久化。
- 2、不支持多机协调。例如：一个 APP 部署在两个 vm，通过一个 PHP 提交任务，请求会随机路由到其中一台；同样性质/内容的任务，两次请求可能会分别路由到两个 vm，两个 vm 都会执行。开发者要自己保证对业务无影响。

### 异步任务服务使用示例

#### 提交任务

```
<?php
function mytask(){
    $result="do something";
    //return $result;
    return array("res:"=>$result, "name:"=>"myName", "inner"=> array("type"=>3,"id"=>"abc"));
}

$taskId = submit_task(mytask);
//save taskId to db
echo $taskId;
?>
```

#### 查询任务

//isv 可以完全不关心执行结果,或者通过自己的 db 状态判断结果；也可以调用 lookup\_task 查询结果：

```
<?php
$taskId = "730339294396284928";//query taskId from db;
$result = lookup_task($taskId);
if($result["status"] == "success"){
    $obj = $result["result"];
    //do something like displaying result
    echo "Task succeeded. result:". $obj;
}else if($result["status"] == "fail"){
    $obj = $result["errmsg"];
    //do something like displaying errmsg
    echo "Task executed failed. errmsg:". $obj;
}else if($result["queued"] == "fail"){
    //任务尚未调度
    echo "Task not scheduled yet.";
}else{
    //status 为 unknow
    //查自己数据库看是否任务已完成，或重新提交任务
    echo "Task status is unknow";
}
```

```
}  
?>
```

## RPC 服务

### RPC 服务简介

TAE 提供了 RPC ( Remote Procedure Call ) 调用服务，允许 Consumer 程序在远程机器上调用服务提供者 Provider 程序提供的服务。

### RPC 服务 API

本节对 RPC 服务 API 的使用方式进行介绍。关于详细的 API 说明，请参考 [Java Doc](#)。

### 引入服务包

请参考[引入服务包依赖](#)

### 获取 RPC 服务对象

服务提供者 ( Provider ) :

```
import com.alibaba.appengine.api.rpc.RpcFactory;  
import com.alibaba.appengine.api.rpc.ServiceConfig;  
ServiceConfig<HelloService> serviceConfig = new ServiceConfig<HelloService>();  
serviceConfig.setInterfaceClass(HelloService.class);  
RpcFactory.exportService(serviceConfig, new HelloServiceImpl());
```

服务消费者 ( Consumer ) :

```
import com.alibaba.appengine.api.rpc.ReferenceConfig;  
import com.alibaba.appengine.api.rpc.RpcFactory;  
ReferenceConfig<HelloService> referenceConfig = new ReferenceConfig<HelloService>();  
referenceConfig.setInterfaceClass(HelloService.class);  
reference = RpcFactory.getReference(referenceConfig);
```

### RPC 服务使用示例

本节将向读者展示如何通过 TAE SDK 开发并运行一个本地的 RPC 服务应用。

### 结果演示

访问 127.0.0.1:8080 启动 provider 服务，将返回：

```
server is OK!
```

重新打开一个浏览器窗口，访问 127.0.0.1:9080 启动 consumer 服务，将返回：

```
RPC Invocation Result:
```

```
HELLO
```

```
=====
```

```
Input: default0, output: default0
```

### 准备工作

- 1、 预留大约 15 分钟无干扰时间
- 2、 安装配置 TAE SDK，应用类型设置为云商店（即 app.biztype=32），开发语言设置为 JAVA（即 app.language=JAVA）[参考链接](#)
- 3、 将 SVN 导出的本地 sdk 环境拷贝一份到本地机器，创建两个 tomcat 容器，将 provider 程序部署到 8080 端口，consumer 程序部署到 9080 端口，详细步骤[参考链接](#)
- 4、 分别在两个 sdk 目录的 conf/custom 下的 ace4j-private.properties 文件修改配置  
rpc.registry.address=multicast://224.5.6.7:1234

rpc.port=20880

如何完成

读者可以选择从头开始工程搭建、开发和部署，或者略过这些步骤，选择直接下载源码、打包部署来完成本篇向导。

选择从头开始的读者，也请下载应用源码与 war 包，[rpc\\_provider.war 下载](#)，[rpc\\_consumer.war 下载](#)，应用源码 [rpc\\_consumer.zip 下载](#)，应用源码 [rpc\\_provider.zip 下载](#)，以便参考，并跳到下一节——工程创建。

选择直接部署的读者，请执行以下步骤：

- 1、 下载应用源码与 war 包，[rpc\\_provider.war 下载](#)，[rpc\\_consumer.war 下载](#)，应用源码 [rpc\\_consumer.zip 下载](#)，应用源码 [rpc\\_provider.zip 下载](#)
- 2、 分别将 rpc\_provider.war 文件和 rpc\_consumer.war 文件拷贝到本地解压的容器 webapps 目录和拷贝后的容器 webapps 目录下，并分别重命名为 ROOT.war；
- 3、 分别运行 bin/startup.sh(bat)启动两个容器；
- 4、 容器启动成功后，可以根据结果演示一节进行验证。

开发实战

- 1、 创建两个 maven 工程，分别命名为 rpc\_provider 和 rpc\_consumer
- 2、 在 rpc\_provider 中创建 DemoObject 类作为数据模型，HelloServer 类作为 provider 的服务提供接口，HelloServiceImpl 作为 provider 服务的接口实现类，完整示例参见：[rpc\\_server/src/main/java/com/alibaba/tae/demo/rpc](#)；并在 RpcProvider 这一 servlet 类中添加发布服务代码：

```
ServiceConfig<HelloService> serviceConfig = new ServiceConfig<HelloService>();
serviceConfig.setInterfaceClass(HelloService.class);
RpcFactory.exportService(serviceConfig, new HelloServiceImpl());
```

- 3、 在 rpc\_consumer 中创建同样的 DemoObject 类作为数据模型，HelloServer 类作为 provider 的服务提供接口，完成示例参见：[rpc\\_consumer/src/main/java/com/alibaba/tae/demo/rpc](#)；并在 RpcConsumer 这一 servlet 类中添加消费服务代码：

```
ReferenceConfig<HelloService> referenceConfig = new ReferenceConfig<HelloService>();
referenceConfig.setInterfaceClass(HelloService.class);
HelloService reference = RpcFactory.getReference(referenceConfig);
```

- 4、 初始化 servlet，分别在两个工程的 web.xml 中添加 servlet 的初始化代码：

服务提供者(Provider)：

```
<servlet>
<servlet-name>RpcProvider</servlet-name>
<servlet-class>com.alibaba.tae.demo.rpc.RpcProvider</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>RpcProvider</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
```

服务消费者(Consumer)：

```
<servlet>
<servlet-name>RpcConsumer</servlet-name>
<servlet-class>com.alibaba.tae.demo.rpc.RpcConsumer</servlet-class>
</servlet>
```

```
<servlet-mapping>  
<servlet-name>RpcConsumer</servlet-name>  
<url-pattern>/</url-pattern>  
</servlet-mapping>
```

#### 5、 编译打包

使用 maven 的开发在工程目录下执行下面的命令，编译打包

```
mvn clean install
```

其他开发者，利用自己开发工具的打包功能，编译打包

6、 将生成的 war 包分别拷贝到已准备好的对应的两个本地容器的 webapps 目录，分别重命名为 ROOT.war

7、 分别运行 bin/startup.sh(bat)启动两个容器；

8、 按照结果演示一节进行结果验证