

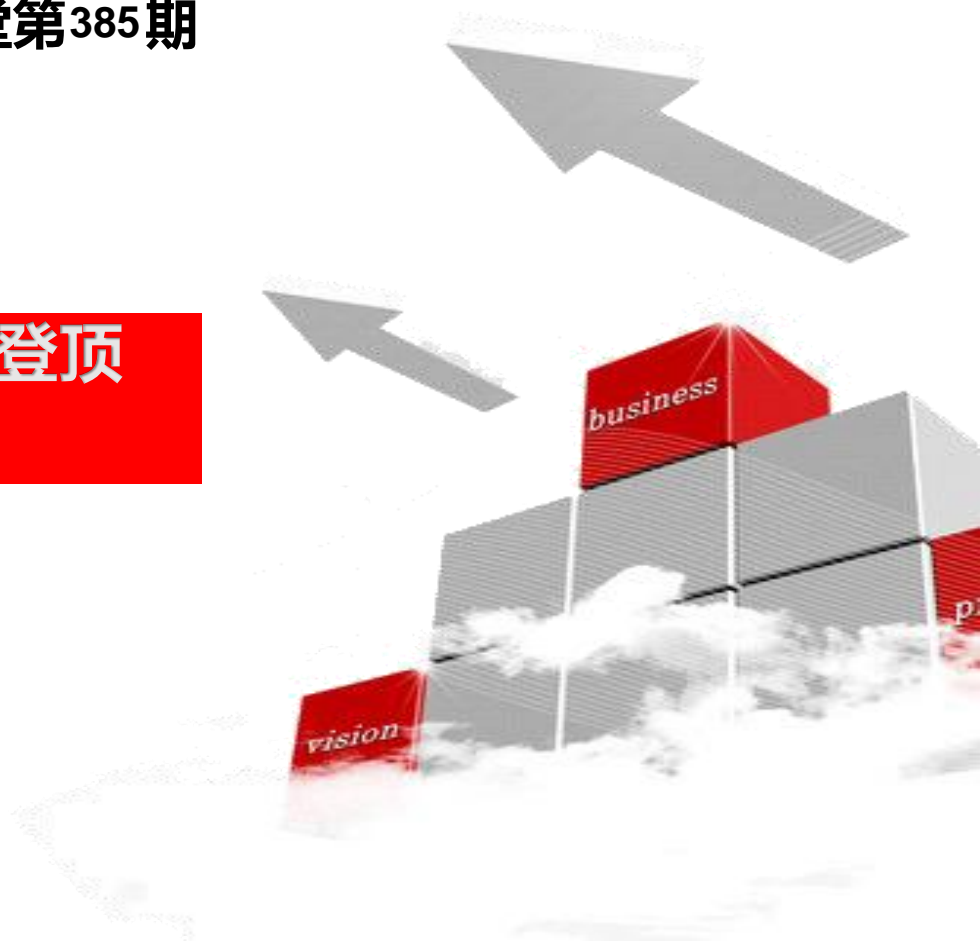


**2020**

**ITIL先锋论坛专家公益微课堂第385期**

**iTop二次开发之华山登顶  
接口服务开发**

**主讲嘉宾：冀利斌**



- 开发经理
- 个人介绍：北航工程硕士，14年IT开发及运维从业经验。曾服务过神华信息、久其软件、文思海辉等企业，具有丰富的政府信息化、企业信息化、金融行业项目的需求分析、设计、研发及运维经验。
- 认证证书：ScrumMaster、DevOps Professional、PRINCE2 Practitioner、ISO20000 Auditor、ITIL V3 Foundation等
- 热衷参与社区活动，包括：ITIL先锋论坛、敏捷之旅、中国DevOps社区、IDCF社区、IT达人圈等。
- 参与编写《开源IT运维管理软件-iTop实施指南》、组织iTop Wiki汉化以及建立itophub.cn中文wiki站点



## 关于 iTop

iTop代表IT运维门户（IT Operational Portal）。iTop是一个开源的网页应用程序，用于IT环境的日常操作。iTop是根据ITIL最佳实践思想设计，但是并不没有规定任何具体的流程，这个应用足够的灵活去适配你的流程，无论你是想要非正式的、务实的流程还是严格的ITIL对齐行为。

iTop的核心是CMDB（配置管理数据库），这是iTop起初开发的第一部分。后续开发的工单和所有的衍生流程。

iTop背后的理念是，CMDB是必须的运维工具。保证准确和最新的CMDB唯一方法是由IT团队（支持代理、IT工程师等）经常使用。此外，CMDB要与其他IT工具（监控系统、报告工具、自动化库存等）集成得越多越好。

iTop特性，使用它可以做到：

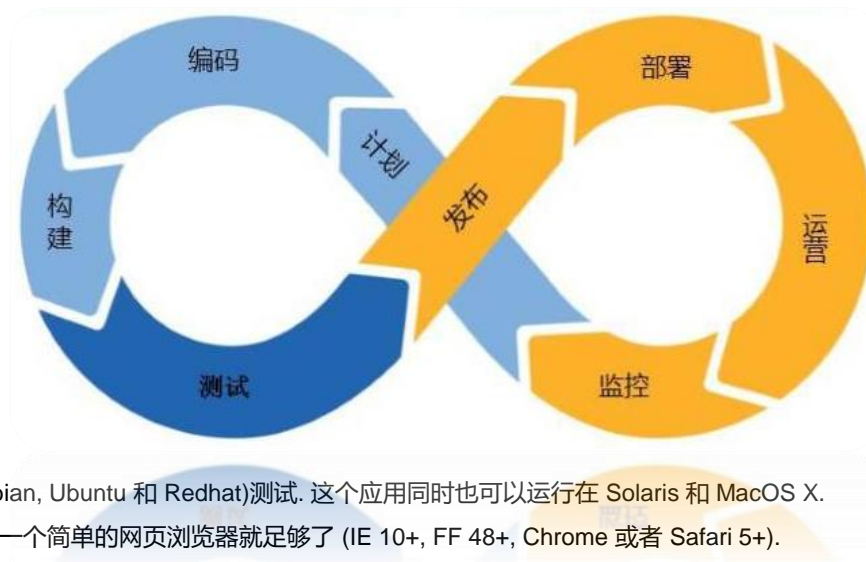
- 记录IT基础设施以及基础设施的各个部分和涉众之间的所有关系（服务器，应用，网络设备，虚拟机，联络人，位置）
- 管理事件，用户请求，计划停机
- 记录IT服务和包括服务级别协议的外部供应商合同
- 通过手工或脚本的方式导出所有信息
- 批量导入（手工和脚本）或者同步/集成来自外部系统的任何数据

iTop能够被不同类型的人员使用：

- 服务台代理
- 支持工程师（一线，二线）
- 服务经理
- IT经理
- 最终用户：可以使用简化的“门户”界面直接提交请求。

iTop依赖 Apache/IIS, MySQL 和 PHP:

- 能够运行在支持这些应用的任何操作系统。已经在Windows, Linux (Debian, Ubuntu 和 Redhat)测试. 这个应用同时也可以运行在 Solaris 和 MacOS X.
- iTop 基于网页的应用，不需要在用户的个人电脑部署任何客户端软件。一个简单的网页浏览器就足够了 (IE 10+, FF 48+, Chrome 或者 Safari 5+).



1

**iTop接口服务-简介**

2

**iTop接口服务-使用演示**

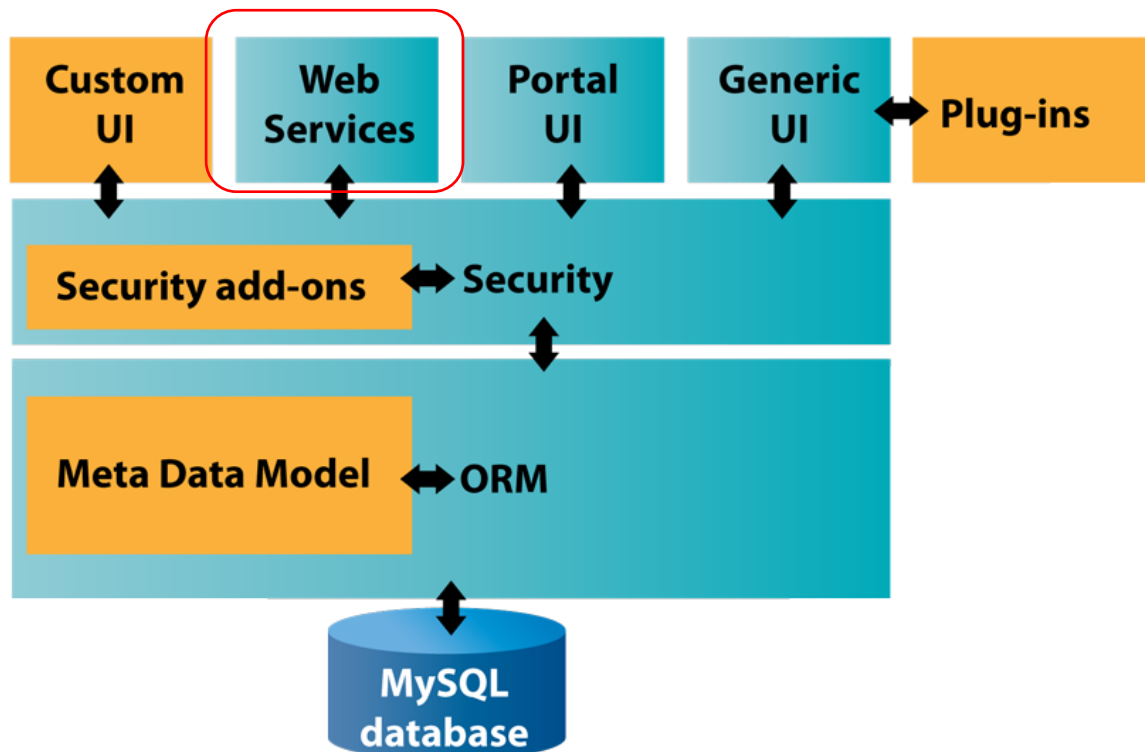
3

**iTop接口服务-WIKI说明**

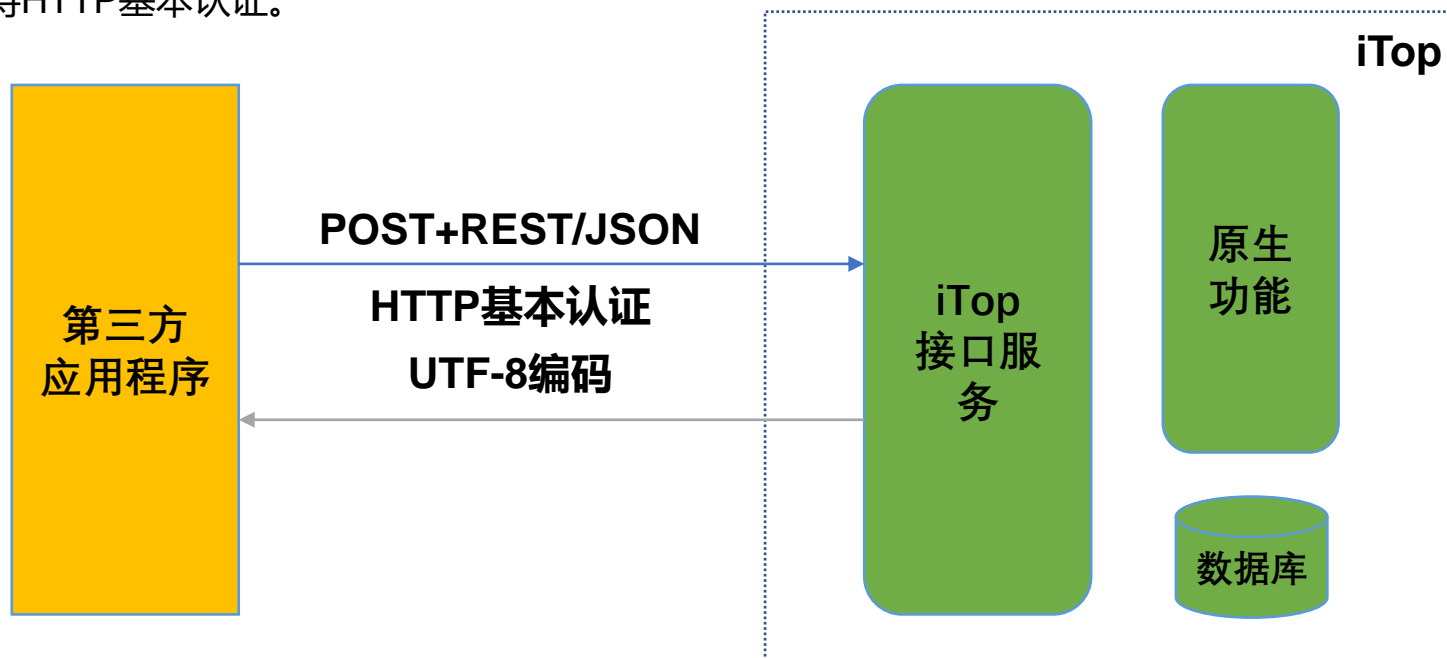
4

**iTop接口服务-案例探究**

## iTop 2.0 architecture



- REST/JSON接口，允许第三方应用程序与iTop进行远程交互，用来检索、创建或更新iTop对象。
- 该接口基于一组简单的HTTP POST请求。
- 从iTop传递和检索的数据使用UTF-8字符集的JSON编码。
- 支持HTTP基本认证。





1

iTop接口服务-简介

2

iTop接口服务-使用演示

3

iTop接口服务-WIKI说明

4

iTop接口服务-案例探究

## 参数配置:

- 允许任何用户访问REST web服务: 配置参数secure\_rest\_services并将其设置为false。
- iTop配置文件设置allowed\_login\_types启用url。

## 角色配置:

- 从iTop 2.5.0开始, 拥有REST Services User角色的用户才能访问REST web服务。
  - ◆确保使用REST web服务的脚本具有REST Services User角色。
  - ◆如果没有此REST Services User附加角色, 只有管理员角色的用户是无法访问REST的。





## 通过接口创建工单

- 使用Bash代码创建工单
- 使用Python代码创建工单



1

iTop接口服务-简介

2

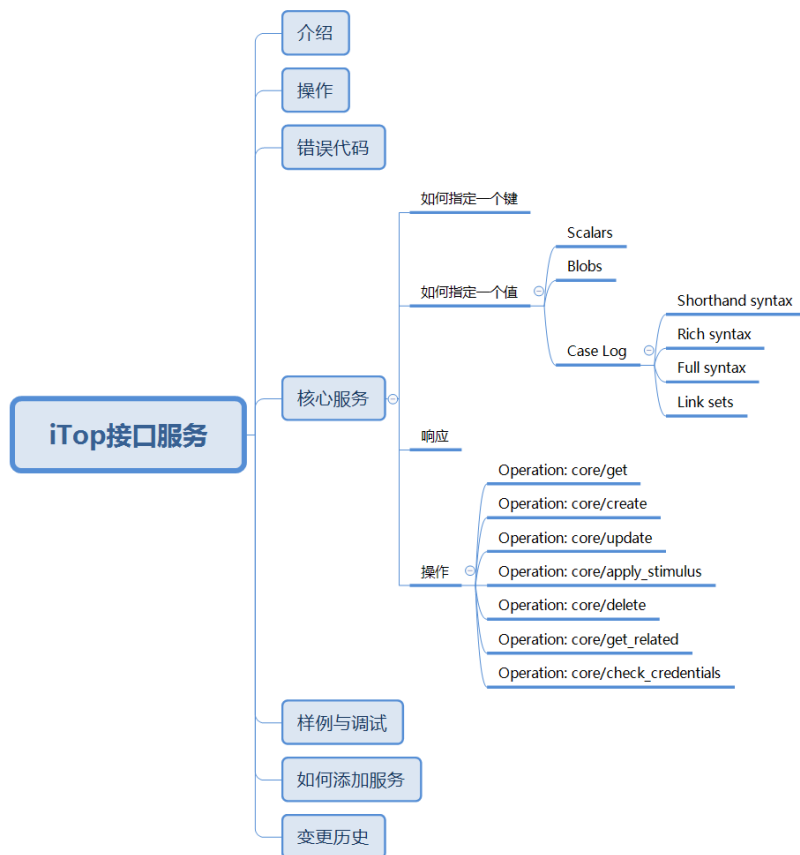
iTop接口服务-使用演示

3

iTop接口服务-WIKI说明

4

iTop接口服务-案例探究



详见: <https://www.itophub.cn/bin/view/4/4.5/>

iTop提供了一个REST/JSON接口，允许第三方应用程序与iTop进行远程交互，用来检索、创建或更新iTop对象。该接口基于一组简单的HTTP POST请求。从iTop传递和检索的数据使用UTF-8字符集的JSON编码。

这样的请求可以通过任何能够发出HTTP/POSTs和操作JSON编码数据的编程语言执行。通过跨域脚本(iTop支持CORS和JSON-P)，请求甚至可以直接在任何web页面内的javascript中运行。[REST JSON Playground](#)中给出了一个用javascript编写跨域脚本的例子。

POST请求的URL像这样：<itop-root>/webservices/rest.php?version=1.3

这些请求需要传递下列参数：auth\_user | auth\_pwd | json\_data

实际上，HTTP方法可以是POST或GET(需要JSON-P支持)。出于安全原因(为了避免以明文形式传递凭据)，以及GET对输入数据的大小施加了限制，建议使用POST和HTTPS。支持HTTP基本身份验证。

## 访问角色：

从iTop 2.5.0开始，拥有REST Services User角色的用户才能访问REST web服务。

- 由于可以很容易地将REST web服务放入循环中，因此用户必须对修改的类具有写访问和批量写权限。变更状态属于修改。
- 确保使用REST web服务的脚本具有**REST Services User**角色。
- 如果用户没有**REST Services User**附加角色，即使有**Administrator**角色的用户是无法访问REST的。

如果模拟前面的行为(即允许任何用户访问REST web服务)，请添加配置参数secure\_rest\_services并将其设置为false。

根据测试REST/JSON API的方式，需要在iTop配置文件中设置allowed\_login\_types中启用url。

## HTTP参数

参数	描述	默认值
version	API版本。它是一种确保目标iTop服务器可以提供若干功能的方法，并且确保了脚本的稳定：只要版本可用，操作将保持不变，除以下情况：bug修复，返回消息的修改，返回的JSON结构中的新元素。	-
auth_user	用户登录名	-
auth_pwd	用户密码	-
json_data	包含处理请求所需的所有信息的结构字符串。 特别注意，这里给出请求的操作。	
callback	使用JSONP时，设置。	

这些参数应用到所有类型的请求操作。

## 列出操作: list\_operations

应该熟悉的第一个操作是: list\_operations。此命令返回所有可能操作的列表。

json\_data输入的语法非常简单:

```
{ "operation": "list_operations" }
```

响应如下:

```
{
  "version": "1.2",
  "operations":
  [
    {
      "verb": "core/create",
      "description": "Create an object",
      "extension": "CoreServices"
    },
    {
      "verb": "core/update",
      "description": "Update an object",
      "extension": "CoreServices"
    },
    {
      "verb": "core/get",
      "description": "Search for objects",
      "extension": "CoreServices"
    }
  ],
  "code": 0,
  "message": "Operations: 3"
}
```

## 错误代码

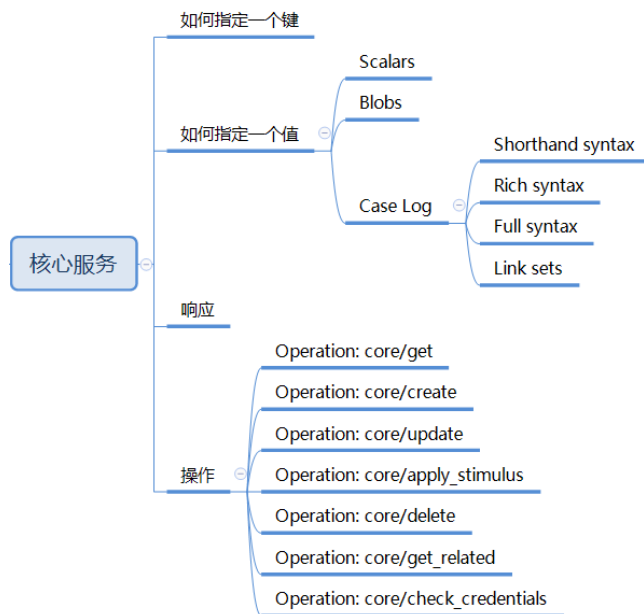
错误代码可以在applicationextension.inc.php中找到，作为类RestResult的常量。

值	常量	含义
0	OK	没有遇到任何问题
1	UNAUTHORIZED	缺失/错误的凭据或用户没有足够的权限执行请求操作
2	MISSING_VERSION	缺失 'version' 参数
3	MISSING_JSON	缺失'json_data' 参数
4	INVALID_JSON	输入结构不是一个有效的JSON字符串
5	MISSING_AUTH_USER	缺失 'auth_user' 参数
6	MISSING_AUTH_PWD	缺失 'auth_pwd' 参数或者正在使用 <a href="#">URL登录方式</a> 但在iTop配置文件没有允许URL登录方式
10	UNSUPPORTED_VERSION	指定版本没有操作可用
11	UNKNOWN_OPERATION	指定版本请求的操作无效
12	UNSAFE	因为可能引起数据（完整性）丢失，请求的操作不能执行
100	INTERNAL_ERROR	无法执行该操作，请查看有关疑难解答的信息

## 核心服务

核心服务是一些**通用**的服务。它们等价于 iTop Core PHP APIs: DBOBJECT, DBOBJECTSEARCH 和 DBOBJECTSET。

使用这些服务，假如你知道足够多的类和它们的属性的话，你能够操作所有种类的数据。





## 如何指定一个键

一个键可以识别一个对象。此类规范可用于确定操作的目标对象，并用于确定 external key（或 “foreign key” ）的值。

三种允许的格式：

- 指定对象的id（数值型）：

... "key": 123

- 指定一个查询语句 (OQL):

... "key": "SELECT UserRequest WHERE caller\_name LIKE \"monnet\""

- 指定查询条件（所有的查询条件通过AND操作符组合）：

... "key": { "name": "Monnet", "first\_name": "Claude" }

## 如何指定一个值

给定属性值的格式依赖于属性类型。

下面的例子阐述了这些情况：

```
"name": "Monnet",  
"first_name": "Claude",  
"age": 80,  
"picture": {  
    "data": "iVBORw0KGgoAAAAN.....AAAAEIFTkSuQmCC",  
    "filename": "smiley.png",  
    "mimetype": "image/png"  
}
```

## 二进制大对象 Blobs

对于Blob（即二进制数据），该值以三个项目的数组形式给出：数据，文件名和mimetype。

“数据”是已被base 64编码的二进制数据（请参阅PHP / base64\_encode）

“文件名”是原始文件的名称。其目的是提供默认文件名，以防进一步下载该文件。

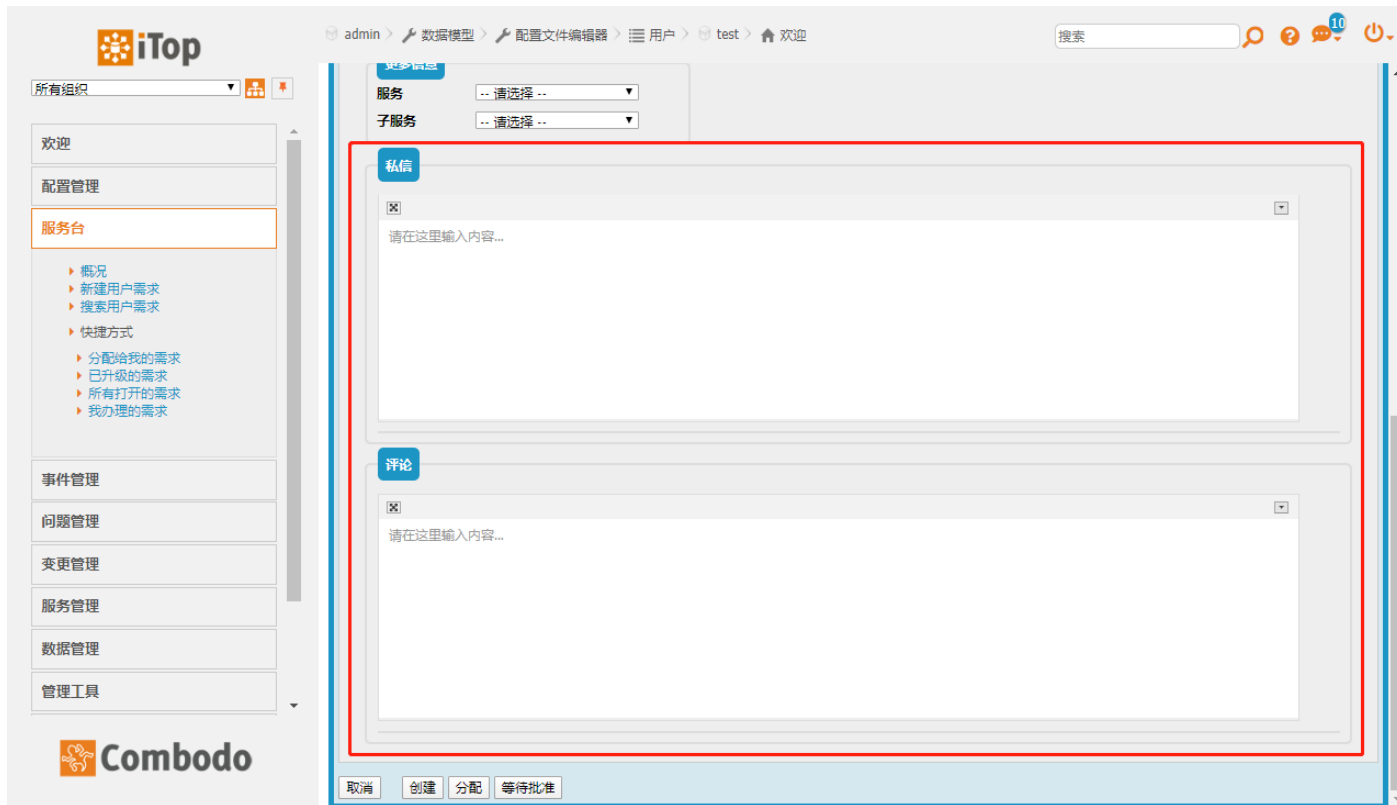
“mimetype”必须与文件格式匹配，以便iTop可以正确显示它。

例：

```
"picture": {  
    "data": "iVBORw0K...QU1BAAC3ao+MkWfFnFj5IAAAAAEIFTkSuQmCC",  
    "filename": "smiley.png",  
    "mimetype": "image/png"  
}
```

## 工单日志 Case Log

从iTop 2.0.3开始, “CaseLog”字段可以使用3种语法: 简写语法, 一条日志或整个日志。



The screenshot shows the iTop web interface for creating a Case Log. The interface is in Chinese and includes a sidebar with navigation options like '服务台' (Service Desk) and '事件管理' (Event Management). The main form area is titled 'Case Log' and contains two text input fields: '私信' (Private Message) and '评论' (Comment). Both fields have a placeholder text '请在这里输入内容...' (Please enter content here...). The form also includes dropdown menus for '服务' (Service) and '子服务' (Sub-service), and buttons for '取消' (Cancel), '创建' (Create), '分配' (Assign), and '等待批准' (Waiting for approval).

## 简写语法 Shorthand syntax

如果提供了简单的文本字符串，则该文本将作为具有当前日期和时间以及当前用户的工单日志中的新条目添加，并且该文本将被隐式地视为HTML格式。

例：

```
"public_log": "blah blah blah",
```

## 富语法 Rich syntax

如果提供了带有名为add\_item的条目的结构，则可以指定其他信息：

- 用户名（默认为REST/JSON API使用的凭据）
- 输入的日期/时间（默认为“现在”）
- 注释格式（默认为“html”，可设置为“文本”）

例：

```
"public_log": {  
  "add_item":  
  {  
    "date": "yyyy-mm-dd hh:mm:ss",  
    "user_login": "jfoo",  
    "message": "The first line\nAnother line",  
    "format": "text"  
  }  
},
```

## 全部语法 Full syntax

如果提供了一个名为items的数组，那么工单日志的整个内容将被提供的条目替换。

例：

```
"public_log":  
{  
  "items":  
  [  
    {  
      "date": "yyyy-mm-dd hh:mm:ss",  
      "user_login": "jfoo",  
      "message": "<p>The first line</p><p>A second line</p>",  
      "format": "html"  
    },  
    {  
      "date": "yyyy-mm-dd hh:mm:ss",  
      "user_login": "jfoo",  
      "message": "blah blah blah",  
      "format": "text"  
    }  
  ]  
},
```

## 链接集合 Link sets

对于链接集，值是对象定义的数组。

例：

```
"contacts_list":  
[  
  {  
    "role": "artist",  
    "contact_id":  
    {  
      "finalclass": "Person",  
      "name": "monet",  
      "first_name": "claudesimonet"  
    }  
  },  
  {  
    "role": "Basket-ball",  
    "contact_id": "SELECT Team WHERE name = 'San Antonio Spurs'"  
  },  
  {  
    "contact_id": 1234  
  }  
]
```



## 响应

每个对象的位置是这个格式：

core/...操作响应的格式如下：

```
{
  "objects":
  {
    "objectclass::objectkey":
    {
      ...
    }
  },
  "code": 0,
  "message": "Found: 1"
}
```

```
{
  "code": 0,
  "message": "",
  "class": "Person",
  "key": 1234,
  "fields": {
    "id": 1234,
    "name": "My last name",
    "status": "Active",
    "org_id": 123,
    ....
    ....
    ....
  }
}
```

## 响应

给定的属性列表可以通过参数输出字段（如果可用）进行控制。参数output\_fields可以有以下形式：

- 由逗号分隔的属性代码列表（例如：“name, status, org\_id”）。
- 这里只能给出被查询类的属性。
- \* 表示查询类的所有属性（子类可能比查询的类具有更多的属性）

## 操作-获取 core/get

查询对象列表。例如：

通过下方的json\_data：

```
{  
  "operation": "core/get",  
  "class": "Person",  
  "key": "SELECT Person WHERE email LIKE '%.com'",  
  "output_fields": "friendlyname, email"  
}
```

或者使用 “key” 的另外格式：

```
{  
  "operation": "core/get",  
  "class": "Person",  
  "key": 1,  
  "output_fields": "**"  
}
```

## 操作-获取 core/get

从2.6.1开始，使用两个新参数处理分页：

- **limit (int): 返回结果的数量 (default: 0 = 不限制)**
- **page (int): 返回的页码数 (不能< 1)**

可以使用classes/class/properties/order 模型XML节点控制记录的排序规则。

例如：

```
{  
  "operation": "core/get",  
  "class": "Person",  
  "key": "SELECT Person",  
  "output_fields": "friendlyname, email"  
  "limit": "5",  
  "page": "2"  
}
```

## 操作-新建 core/create

创建指定类的新对象，通过下方的json\_data：

```
{
  "operation": "core/create",
  "comment": "Synchronization from blah...",
  "class": "UserRequest",
  "output_fields": "id, friendlyname",
  "fields":
  {
    "org_id": "SELECT Organization WHERE name = \"Demo\"",
    "caller_id":
    {
      "name": "monet",
      "first_name": "claud",
    }
    "title": "Houston, got a problem!",
    "description": "The fridge is empty"
  }
}
```

...最终结果创建一个新的用户请求。

## 操作-新建 core/create

某些属性具有特殊格式：

- 链接集 Link sets：目前只支持间接链接集，请参阅上面给出的示例
- 二进制大对象 Blobs：文档内容（例如DocumentFile/file）的格式必须是：  
{data: base64-encoded-data, mimetype: ..., filename: ...}
- 工单日志：允许三种形式。
  - ◆ 传递一个字符串相当于从GUI中添加一个新消息：消息由代表当前用户（用于调用web服务的凭证）记录。
  - ◆ 以{add\_item:{message:'blah', user\_id:123, date:'2012-02-28 10:30'}}形式传递一个结构也会添加一条消息。user\_id和date是可选的，分别默认为当前用户和当前日期和时间。指定用户标识要求用于调用服务的凭据具有管理员权限，否则操作将失败并显示未经授权的错误代码。
  - ◆ 以{items:[{message:'blah'}]}形式传递一个结构来设置整个日志。

## 操作-新建 core/create

响应示例:

```
{
  "code": 0,
  "message": "",
  "objects":
  {
    "UserRequest::123":
    {
      "code": 0,
      "message": "created",
      "class": "UserRequest",
      "key": 29,
      "fields":
      {
        "id": 29,
        "friendlyname": "R-000029"
      }
    }
  }
}
```

## 操作-更新 core/update

更新一个对象。传递一下json\_data:

```
{
  "operation": "core/update",
  "comment": "Synchronization from blah...",
  "class": "UserRequest",
  "key":{"description": "The fridge is empty"},
  "output_fields": "friendlyname, title, contact_list",
  "fields":
  {
    "contacts_list":
    [
      {
        "role": "pizza delivery",
        "contact_id":
        {
          "finalclass": "Person", "name": "monet", "first_name": "claudio"
        }
      }
    ]
  }
}
```

... 将更新用户请求（通过描述“The fridge is empty”识别的），设置联系人列表为一个联系人（Claude Monet），角色是‘pizza delivery’。



## 操作-变更状态 core/apply\_stimulus

更新对象并应用一个刺激去改变对象的状态，传递以下的json\_data：

```
{
  "operation": "core/apply_stimulus",
  "comment": "Synchronization from blah...",
  "class": "UserRequest",
  "key": 15,
  "stimulus": "ev_assign",
  "output_fields": "friendlyname, title, status, contact_list",
  "fields":
  {
    "team_id": 18,
    "agent_id": 57
  }
}
```

... 将更新用户请求（通过键“15”识别的），设置agent\_id和team\_id字段（对于已分配状态是必填的），并且然后将应用刺激ev\_assign使工单状态从新建到已分配状态。

## 操作-删除 core/delete

删除对象集合，传递下面的json\_data：

```
{
  "operation": "core/delete",
  "comment": "Cleanup for customer Demo",
  "class": "UserRequest",
  "key":
  {
    "org_id": 2
  },
  "simulate": false
}
```

... 将删除2号组织的所有用户请求。

删除可能意味着删除和/或更新链接到已删除对象的其他对象。所有对象都列在报告中。

报告的每个对象都有一个删除状态代码和一条提供附加信息的信息。状态代码在core/restservices.class.inc.php中定义，作为RestDelete类的常量。

## 操作-删除 core/delete

值	常量	含义
0	OK	已根据初始请求删除对象
1	ISSUE	一般问题 (用户权限 or ... ?)
2	AUTO_DELETE	必须删除才能保持数据库的完整性
3	AUTO_DELETE_ISSUE	必须删除以保持数据库的完整性, 但这是不可能的
4	REQUEST_EXPLICITLY	必须删除以保持数据库的完整性, 但必须明确请求
5	AUTO_UPDATE	必须更新以保持数据库的完整性
6	AUTO_UPDATE_ISSUE	必须更新以保持数据库的完整性, 但这是不可能的

## 操作-获取连接项 core/get\_related

查询依赖对象，给出一个对象或者对象列表，搜索正在影响或受这些对象影响的其他对象。

传递一下json\_data:

```
{  
  "operation": "core/get_related",  
  "class": "Server",  
  "key": 1,  
  "relation": "impacts",  
  "depth": 4,  
  "redundancy": true,  
  "direction": "down"  
}
```

... 将查询id为1的服务器影响的所有配置项。查询深度限制在4级（默认20级）。

## 操作-获取连接项 core/get\_related

这样的格式显示结果：

```
{
  "objects":
  {
    "objectclass::objectkey":
    {
      ...
    }
  },
  "relations":
  {
    "origin-class::origin-key":
    [
      {
        "key": "destination-class::destination-key"
      }
    ]
  },
  "code": 0,
  "message": "Scope: 1; Found: Server=4, VirtualMachine=3, Farm=1"
}
```

## 操作-认证 core/check\_credentials

检查用户login + password, 传递如下json\_data:

```
{  
  "operation": "core/check_credentials",  
  "user": "john",  
  "password": "abc123",  
}
```

...将相应 (如果一切正常的话) ...

```
{  
  "code": 0,  
  "message": "",  
  "authorized": true,  
}
```



JSRUN测试: <http://jsrun.net/fh6Kp>

集成REST/JSON调用的方法示例: [使用bash和wget](#)

可以通过开发一个模块（或扩展）来扩展服务，该模块声明一个实现iRestServiceProvider接口的类（包含在自定义模块中）。

安装模块后，您的自定义服务将可用，并用operation=list\_operations列出。



## OQL对象查询语言

iTop使用对象查询语言（简称：OQL）去定义检索持久化对象的查询。

OQL查询在iTop中定义或配置对象集合的很多地方被使用（审计规则，通知等）。

OQL语言的目标是隐藏数据模型中实际SQL模式的复杂性，利用OQL定义提供了一个安全(查询时进行访问权限控制)，强大的(OQL可以表达复杂的条件)，高效的(OQL查询解析成一个SQL查询)，对象清晰的（尊重类继承）并且简单的语法。

OQL语句类似于SQL语句，OQL语法是基于SQL语言的子集。

到目前为止，只有SELECT语句已经实现。

这样的语句只能返回两种类型的数据集：

- 要么是指定类的对象列表。
- 或者是对象和关联对象的列表（每行有对个类对象）。

详见：<https://www.itophub.cn/bin/view/6/备用内容/2%20系统管理/2.4.01-日常管理/02-对象查询语言OQL/>

iTop 版本	JSON REST 版本	变更
2.0.1	1.0	创建
2.0.2	1.1	在返回的对象信息中添加了'key' (在1.0中需要一些解析)，将对象搜索条件转换为严格的搜索（与松散的搜索相反：contains...），允许重置外部键（设置为0，表示“未定义”）
2.0.3	1.2	完全处理工单日志（可以完全读取或写入）、枚举/获取提供原始值而不是本地化标签、允许HTTP基本身份验证方法、添加动词core/check_凭据、改进错误报告（缺少身份验证参数、写入链接集的类错误）、添加选项*+以输出所有字段对象（不仅是查询类的字段），修复了对象删除报告
2.2.0	1.3	动作 get_related：增加了选项redundancy和direction，以便在影响分析中考虑冗余。出于安全原因，用户帐户对profile portal用户使用REST/JSON webservice被禁用。如果只使用其中一个webservice检查用户的凭据，请调整代码以使用core/check_credentials操作。出于安全原因，core/get和core/get_related的操作现在只允许对指定的对象类具有批量读取权限的用户使用。
2.5.2, 2.6.1, 2.7.0	1.4	动作 core/get : 增加了分页参数 limit 和 page



1

iTop接口服务-简介

2

iTop接口服务-使用演示

3

iTop接口服务-WIKI说明

4

iTop接口服务-案例探究

## iTop小程序





iTop 微信公众号



iTop 小程序



iTop 案例



# 感谢聆听！

愿我们一同成长，奔向更好的未来！



- 最新预告&PPT更新，请关注论坛微信公众号
- ITIL先锋论坛专家直播讲堂，期待各行各业大咖分享，欢迎联系Karen。
- 提供ITIL /DEVOPS/PRINCE2/ITSS/ISO27001/TOGAF/COBIT等高质量课程培训
- 电话：4008060230，扫一扫微信



ITIL先锋论坛专家直播讲堂  
请加Karen老师微信



看预告&PPT更新  
请关注微信公众平台二维码



咨询电话  
4008060230



咨询Karen老师  
QQ:80049496

# 选择艾拓先锋培训机构的理由

○ 名优讲师 精心指导

○ 专家讲堂 公益分享

○ 火爆社区 求职推荐

○ 学员口碑 成就梦想



**50000人社区! 100000+帖子! 超2500名学员!**

PEOPLECERT/EXIN权威授权! 国内名副其实的IT服务管理培训第一品牌!

咨询电话: 400-806-0230 QQ群: 119205977

