

# 平行盘点综合管理系统

## PDA 接口文档

平行自动化系统有限公司

刘峰

# 目 录

平行盘点综合管理系统.....	1
PDA 接口文档.....	1
1 概述.....	3
1.1 文档目的.....	3
1.2 功能介绍.....	3
1.3 接口调用示例.....	3
1.3.1 SOAP1.1.....	3
1.3.2 SOAP1.2.....	4
1.3.3 HTTP GET.....	5
1.3.4 HTTP POST.....	5
2 平行盘点综合管理系统 WebService 接口发布.....	6
2.1 用户登录接口.....	6
2.1.1 用户登录.....	6
2.2 档案记录接口.....	7
2.2.1 根据二维码标签获取该密集架/列/AB 面/组/格中的档案记录.....	7
2.3 位置信息接口.....	9
2.3.1 获取库房.....	9
2.3.2 获取区.....	9
2.3.3 根据库房、区获取密集架.....	11
2.3.4 根据密集架 ID 获取密集架.....	12
2.3.5 获取列.....	13
2.3.6 获取 AB 面.....	14
2.3.7 获取组.....	15
2.3.8 获取格.....	16

# 1 概述

## 1.1 文档目的

根据该文档可进行系统数据集成开发。

## 1.2 功能介绍

该接口是依赖 Web Service 技术将平行盘点综合管理系统的数据相集成的开放式接口，目前提供库房数据、密集架数据、档案信息数据等。

## 1.3 接口调用示例

下面是各种协议调用该接口的示例，包括SOAP1.1、SOAP1.2、HTTP GET、HTTP POST方式。以“`public string UserLogin(string userName, string userPwd)`”方法为例。

### 1.3.1 SOAP1.1

以下是 SOAP 1.1 请求和响应示例。所显示的占位符需替换为实际值。

```
POST /CSharpBackground/WebServer.asmx HTTP/1.1
Host: 192.168.0.222
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/UserLogin"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserLogin xmlns="http://tempuri.org/">
      <userName>string</userName>
      <userPwd>string</userPwd>
    </UserLogin>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserLoginResponse xmlns="http://tempuri.org/">
      <UserLoginResult>string</UserLoginResult>
    </UserLoginResponse>
  </soap:Body>
</soap:Envelope>
```

## 1.3.2 SOAP1.2

以下是 SOAP 1.2 请求和响应示例。所显示的占位符需替换为实际值。

```
POST /CSharpBackground/WebServer.asmx HTTP/1.1
Host: 192.168.0.222
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UserLogin xmlns="http://tempuri.org/">
      <userName>string</userName>
      <userPwd>string</userPwd>
    </UserLogin>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UserLoginResponse xmlns="http://tempuri.org/">
      <UserLoginResult>string</UserLoginResult>
    </UserLoginResponse>
  </soap12:Body>
</soap12:Envelope>
```

### 1.3.3 HTTP GET

以下是 HTTP GET 请求和响应示例。所显示的占位符需替换为实际值。

```
GET
/CSharpBackground/WebServer.aspx/UserLogin?userName=string&userPwd=st
ring HTTP/1.1
Host: 192.168.0.222
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

### 1.3.4 HTTP POST

以下是 HTTP POST 请求和响应示例。所显示的占位符需替换为实际值。

```
POST /CSharpBackground/WebServer.aspx/UserLogin HTTP/1.1
Host: 192.168.0.222
Content-Type: application/x-www-form-urlencoded
Content-Length: length

userName=string&userPwd=string
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

## 2 平行盘点综合管理系统 Webservice 接口发布

### 2.1 用户登录接口

#### 2.1.1 用户登录

函数名称	<code>public string UserLogin(string userName, string userPwd)</code>
描述	<p>用户登录。</p> <p>参数描述：</p> <p>userName: 用户名称。不能为空。</p> <p>userPwd: 用户密码</p>
返回	以json格式返回。包括登录是否成功标志、用户ID，用户所属局站ID等信息。
示例	<p>调用: <code>UserLogin("admin", "123456")</code></p> <p>返回:</p> <p>登录成功:</p> <pre>{   "success": "true",   "data": {     "ClerkID": "0", //用户ID     "ClerkStationID": "1" //用户所属局站ID   } }</pre> <p>登录失败:</p> <pre>{   "success": "false",   "data": "" }</pre>

	}
备注	

## 2.2 档案记录接口

### 2.2.1 根据二维码标签获取该密集架/列/AB 面/组/格中的档案记录

函数名称	<code>public string GetArchivesByTDC(string twoDCLabelID, string equType, string startRecord, string sumRecord)</code>
描述	<p>根据二维码标签获取该密集架/列/AB面/组/格中的档案记录。</p> <p>参数描述：（以下参数均不能为空）</p> <p>twoDCLabelID：二维码标签号</p> <p>equType：设备类型。0-不区分类型；1-密集架；2-列；3-AB面；4-组；5-格</p> <p>startRecord：显示记录的开始条数，从0开始。</p> <p>sumRecord：一共显示多少条</p>
返回	<p>以json格式返回。包括当前二维码标签所指示的位置、档案的相关信息。</p> <p>返回参数说明：</p> <p>FMaterial：介质。1-纸质（盒）；2-光盘（盒）；3-实物</p> <p>DurationOfStorage：保管期限。1-10年；2-30年；3-永久</p> <p>SecurityLevel：密级。1-绝密；2-机密；3-密级；4-内部资料；5-公开资料</p> <p>ABSide：所在面。1-A面；2-B面</p> <p>ArchivesState：档案状态。1-在架；2-出库；3-借阅；4-注销</p>
示例	<p>调用：GetArchivesByTDC（“TDC601A6B001”，“1”，“0”，“20”）</p> <p>返回：</p> <pre>{   "Position": "601 库房 A 区 6 列 B 面", //位置   "TotalOf": "500", //应有   "OnShelf": "436", //在库   "Borrow": "64", //借阅   "ArchivesData": [     {       "ArchivesNum": "601A6B001", //档号       "RFIDLabelID": "RFID601A6B001", //RFID 标签号       "TwoDCLabelID": "TDC601A6B001", //二维码标签号     }   ] }</pre>

	<pre> "FMaterial": "1", // 介质 "ArchivesRoomNum": "601", // 档案馆（室）号 "MiniatureNum": "wsh601A6B001", // 微缩号 "FileTitle": "案卷题名", // 案卷题名 "PreparationUnit": "平行", // 编制单位 "PreparationTime": "2016-07-01", // 编制日期 "DurationOfStorage": "1", // 保管期限 "SecurityLevel": "3", // 密级 "StoreroomID": "601", // 所属库房 ID "StoreroomName": "601 库房", // 所属库房名称 "SAreaID": "6011", // 所属区 ID "SAreaName": "A 区", // 所属区名称 "CompactShelfID": "60111", // 密集架 ID "CompactShelfName": "密集架 60111", // 密集架名称 "ColNum": "6", // 所在列 "ABSide": "2", // 所在面 "GroupNum": "3", // 所在组 "CaseNum": "2", // 所在格 "ArchivesState": "1" // 档案状态 }, { "ArchivesNum": "601A6B002", // 档号 "RFIDLabelID": "RFID601A6B002", // RFID 标签号 "TwoDCLabelID": "TDC601A6B002", // 二维码标签号 "FMaterial": "1", // 介质 "ArchivesRoomNum": "602", // 档案馆（室）号 "MiniatureNum": "wsh601A6B001", // 微缩号 "FileTitle": "案卷题名", // 案卷题名 "PreparationUnit": "平行", // 编制单位 "PreparationTime": "2016-07-01", // 编制日期 "DurationOfStorage": "1", // 保管期限 "SecurityLevel": "3", // 密级 "StoreroomID": "601", // 所属库房 ID "StoreroomName": "601 库房", // 所属库房名称 "SAreaID": "6011", // 所属区 ID "SAreaName": "A 区", // 所属区名称 "CompactShelfID": "60111", // 密集架 ID "CompactShelfName": "密集架 60111", // 密集架名称 "ColNum": "6", // 所在列 "ABSide": "2", // 所在面 "GroupNum": "3", // 所在组 "CaseNum": "2", // 所在格 "ArchivesState": "1" // 档案状态 }, </pre>
--	--



	<pre> .....//更多记录     ] }</pre>
备注	

## 2.3 位置信息接口

### 2.3.1 获取库房

函数名称	<code>public string GetStoreroom(string storeroomID)</code>
描述	<p>根据库房ID获取库房信息。</p> <p>参数描述：（以下参数均不能为空）  storeroomID：库房ID。当为0时，获取所有库房信息。</p>
返回	以json格式返回。包括库房ID、库房名称等信息。
示例	<p>调用：GetStoreroom("0")</p> <p>返回：</p> <pre> [     {         "StoreroomID":"601",//库房1ID         "StoreroomName":"601库房"//库房1名称     },     {         "StoreroomID":"602",//库房2ID         "StoreroomName":"602库房"//库房2名称     },     .....//更多库房 ]</pre>
备注	

### 2.3.2 获取区

函数名称	<code>public string GetArea(string storeroomID, string sAreaID)</code>
描述	根据库房ID、区ID获取库房区信息。

	<p>参数描述：（以下参数均不能为空）</p> <p>storeroomID：库房ID。当为0时，获取所有库房区信息，此时sAreaID可以是随意数值。</p> <p>sAreaID：库房区ID。当为0时，获取当前库房中所有库房区信息。</p>
返回	以json格式返回。包括库房区ID、库房区名称等信息。
示例	<p>调用：GetStoreroom("0")</p> <p>返回：</p> <pre>[   {     "StoreroomID": "601", // 库房1ID     "AreaInfor": [       {         "SAreaID": "60101", // 库房1区1ID         "SAreaName": "A区" // 库房1区1名称       },       {         "SAreaID": "60102", // 库房1区2ID         "SAreaName": "B区" // 库房1区2名称       },       ..... // 更多库房区     ]   },   {     "StoreroomID": "602", // 库房2ID     "AreaInfor": [       {         "SAreaID": "60201", // 库房2区1ID         "SAreaName": "A区" // 库房2区1名称       },       {         "SAreaID": "60202", // 库房2区2ID         "SAreaName": "B区" // 库房2区2名称       },       ..... // 更多库房区     ]   }   ..... // 更多库房 ]</pre>
备注	

2.3.3 根据库房、区获取密集架

函数名称	<code>public string GetCompactShelf(string storeroomID, string sAreaID, string compactShelfID)</code>
描述	<p>根据库房ID、区ID、密集架ID获取密集架信息。</p> <p>参数描述：（以下参数均不能为空）</p> <p>storeroomID：库房ID。当为0时，获取所有库房区信息，此时sAreaID、compactShelfID可以是随意数值。</p> <p>sAreaID：库房区ID。当为0时，compactShelfID可以是随意数值。</p> <p>compactShelfID：密集架ID。当为0时，获取当前库房当前区内的密集架信息。</p>
返回	以json格式返回。包括密集架ID、密集架名称、RFID标签号、二维码标签号、列总数、分组总数、分格总数、备注等信息（列总数、分组总数、分格总数、备注等字段可能为空）。
示例	<p>调用：GetCompactShelf("601", "60101", "0")</p> <p>返回：</p> <pre>[   {     "StoreroomID": "601", // 库房1ID     "SAreaID": "60101", // 库房1区1ID     "CompactShelfInfor": [       {         "CompactShelfID": "6010101", // 密集架ID         "CompactShelfName": "601库房A区01密集架", // 密集架名称         "RFIDLabelID": "RFID6010001", // RFID标签ID         "TwoDCLabelID": "TDC6010001", // 二维码标签ID         "TotalCol": "8", // 列总数         "TotalGroup": "12", // 分组总数         "TotalCase": "6", // 分格总数         "Remark": "" // 备注       },       {         "CompactShelfID": "6010102", // 密集架ID         "CompactShelfName": "601库房A区02密集架", // 密集架名称         "RFIDLabelID": "RFID6010001", // RFID标签ID         "TwoDCLabelID": "TDC6010001", // 二维码标签ID         "TotalCol": "8", // 列总数</pre>

	<pre>                 "TotalGroup": "12", // 分组总数                 "TotalCase": "6", // 分格总数                 "Remark": "" // 备注             },             ..... // 更多密集架         ]     },     ..... // 更多库房 ] </pre>
备注	

### 2.3.4 根据密集架 ID 获取密集架

函数名称	<code>public string GetCompactShelfByID(string compactShelfID)</code>
描述	<p>根据密集架ID获取密集架信息。</p> <p>参数描述：（以下参数均不能为空）</p> <p><code>compactShelfID</code>：密集架ID。当为0时，获取所有的密集架信息。</p>
返回	以json格式返回。包括密集架ID、密集架名称、RFID标签号、二维码标签号、列总数、分组总数、分格总数、备注等信息（列总数、分组总数、分格总数、备注等字段可能为空）。
示例	<p>调用：<code>GetCompactShelf("6010101")</code></p> <p>返回：</p> <pre> [     {         "StoreroomID": "601", // 库房1ID         "SAreaID": "60101", // 库房1区1ID         "CompactShelfInfor": [             {                 "CompactShelfID": "6010101", // 密集架ID                 "CompactShelfName": "601库房A区01密集架", // 密集架名称                 "RFIDLabelID": "RFID6010001", // RFID标签ID                 "TwoDCLabelID": "TDC6010001", // 二维码标签ID                 "TotalCol": "8", // 列总数                 "TotalGroup": "12", // 分组总数                 "TotalCase": "6", // 分格总数                 "Remark": "" // 备注             }         ]     } ] </pre>

	<pre>         },         {             "CompactShelfID": "6010102", // 密集架ID             "CompactShelfName": "601库房A区02密集架" // 密集架名称             "RFIDLabelID": "RFID6010001", // RFID标签ID             "TwoDCLabelID": "TDC6010001", // 二维码标签ID             "TotalCol": "8", // 列总数             "TotalGroup": "12", // 分组总数             "TotalCase": "6", // 分格总数             "Remark": "" // 备注         },         ..... // 更多密集架     ] }, ..... // 更多库房 ] </pre>
备注	

### 2.3.5 获取列

函数名称	<code>public string GetCompactShelfCol(string storeroomID, string sAreaID, string compactShelfID, string colNum)</code>
描述	<p>获取密集架列信息。</p> <p>参数描述：（以下参数均不能为空）</p> <p>storeroomID：库房ID。不能为0。</p> <p>sAreaID：库房区ID。不能为0。</p> <p>compactShelfID：密集架ID。不能为0。</p> <p>colNum：列编号。当为0时，获取当前密集架的所有列信息。</p>
返回	以json格式返回。包括库房编号、区编号、密集架编号、密集架名称、列编号、列二维码标签号、备注等信息。
示例	<p>调用：GetCompactShelf("601", "60101", "6010101", "0")</p> <p>返回：</p> <pre> [     {         "StoreroomID": "601", // 库房1ID         "SAreaID": "60101", // 库房1区1ID </pre>

	<pre> "CompactShelfID":"6010101",//密集架ID "CompactShelfName":"601库房A区01密集架",//密集架名称 "ColInfor":[     {         "ColNum":"1",//列编号         "TwoDCLabelID":"TDC60100011",//二维码标签ID         "Remark":"","//备注     },     {         "ColNum":"2",//列编号         "TwoDCLabelID":"TDC60100012",//二维码标签ID         "Remark":"","//备注     },     .....//更多列 ] }, .....//更多信息 ] </pre>
备注	

### 2.3.6 获取 AB 面

函数名称	<code>public string GetColABSide(string storeroomID, string sAreaID, string compactShelfID, string colNum, string aBSide)</code>
描述	<p>获取密集架列AB面信息。</p> <p>参数描述：（以下参数均不能为空）</p> <p>storeroomID：库房ID。不能为0。</p> <p>sAreaID：库房区ID。不能为0。</p> <p>compactShelfID：密集架ID。不能为0。</p> <p>colNum：列编号。不能为0。</p> <p>aBSide：AB面编号。当为0时，获取当前面的所有列信息。</p>
返回	以json格式返回。包括库房编号、区编号、密集架编号、密集架名称、列编号、列二维码标签号、备注等信息。
示例	<p>调用：GetColABSide("601", "60101", "6010101", "1", "0")</p> <p>返回：</p> <pre> [     { </pre>

	<pre> "StoreroomID": "601", //库房1ID "SAreaID": "60101", //库房1区1ID "CompactShelfID": "6010101", //密集架ID "CompactShelfName": "601库房A区01密集架", //密集架名称 "ColNum": "1", //列编号 "ABSideInfor": [     {         "ABSide": "1", //AB面编号         "TwoDCLabelID": "TDC601000111", //二维码标签ID         "Remark": "" //备注     },     {         "ABSide": "2", //AB面编号         "TwoDCLabelID": "TDC601000122", //二维码标签ID         "Remark": "" //备注     } ] }, .....//更多信息 ] </pre>
备注	

### 2.3.7 获取组

函数名称	<pre>public string GetColGroup(string storeroomID, string sAreaID, string compactShelfID, string colNum, string aBSide, string groupNum)</pre>
描述	<p>获取密集架列AB面信息。</p> <p>参数描述：（以下参数均不能为空）</p> <p>storeroomID：库房ID。不能为0。</p> <p>sAreaID：库房区ID。不能为0。</p> <p>compactShelfID：密集架ID。不能为0。</p> <p>colNum：列编号。不能为0。</p> <p>aBSide：AB面编号。不能为0。</p> <p>groupNum：AB面编号。当为0时，获取当前密集架的所有列信息</p>
返回	以json格式返回。包括库房编号、区编号、密集架编号、密集架名称、列编号、列二维码标签号、备注等信息。

示例	
备注	

2.3.8 获取格

函数名称	
描述	
返回	
示例	
备注	