API











































**Client and Server Communication**

Client / Mobile application will not have direct access to the Post GIS database, all will be done through a middle layer JSON or XML(<http://en.wikipedia.org/wiki/JSON>). This middle layer will also be called API Functions.

Client are going to make requests to the Server side, and in return will receive the requested response.

Base URI : [ROOT URI]/API

## All communications between mobile and server will be based on an encrypted protocol: HTTPS

## GZIP response

## For large tables, the API can send back gzipped content.

## The tag: “gzipped”:true will send back the ‘result’ array compressed and base64 encoded.

## 

## To decode the content, decode base64 and decompress the gzip content.

## 

## As a matter of reference, we are using the following functions to prepare the string:

## GZIP:<http://ca1.php.net/manual/en/function.gzencode.php>

## Base64:<http://ca1.php.net/manual/en/function.base64-encode.php>

## 

## Response

All communications between client and server will be JSON/XML based.One query to the server can contain one or more requests.

Send a POST request to the server ([ROOT URI]/API), content of this POST request must be a /XMLstring.

ROOT URI =

**Available request fields**

|  |  |
| --- | --- |
| **action** | **Controller to be called.**  **On the server, this file will be located inside ControllerUserManagementController** |
| **method** | **Controller’s function to be called.**  **Inside the controller, this function will be called: getUsers()** |
| **username** | **Username** |
| **password** | **Password** |
| **data** | **Array of parameters to be processed by the function. any number of parameters or sub-parameters can be sent here.** |
| **type** | **Type of the query, only “rpc” is supported for now.** |
| **tid** | **Id of the request, this ID will be reflected into the answer. This is useful if multiple queries are made into one single request.** |

**All those fields are mandatory, even if empty.**

**Available response field**

|  |  |
| --- | --- |
| **type** | **Type of the query, only “rpc” is supported for now.** |
| **tid** | **Id of the request. This is useful if multiple queries are made into one single request.** |
| **action** | **Controller to be called.**  **On the server, this file will be located inside ControllerUserManagementController** |
| **method** | **Controller’s function to be called.**  **Inside the controller, this function will be called: geUsers()** |
| **result** | **An array containing any results sent back by the controller’s function.** |

**Fiddler will be used to test the calls to the API’s functions:**

<http://www.telerik.com/fiddler>

**Firefox plugin called HttpRequest can be used to test the calls to the api functions:**

[**https://addons.mozilla.org/en-us/firefox/addon/httprequester/**](https://addons.mozilla.org/en-us/firefox/addon/httprequester/)

## For calls with a large output, ‘curl’ might be more appropriate, curl for Windows can be downloaded here:<http://www.paehl.com/open_source/?CURL_7.34.0>

For json verfication use <http://www.jsonlint.com>

## UserAuthentication API Response codes

**The following “system-wide” response codes are currently used within the application:**

|  |  |  |
| --- | --- | --- |
| **CODE** | **MESSAGE** | **DESCRIPTION** |
| **SUCCESS** | **Request Successful** | **Default code if no errors have been encountered** |
| **AUTHENTICATIONFAILED** | **Authentication Failed (incorrect username or password)** | **Will happen if username does not exist, or if password is incorrect for this username** |
| **AUTHORIZATIONFAILED** | **Insufficient privileges to access this module.** | **Privileges are incorrectly set for this user** |
|  |  |  |
|  |  |  |
|  |  |  |

**Those response code apply to any queries sent to the Client API. On top of those, each function can have its own response code, documented below.**

**OPENM4S API Functions (summary)**

Some methods, performing simple actions, have been created for internal usage but are also available through the API with limited documentation.

This section of the document contains a list of all methods available through the API.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Action** | **Method** | **INPUT** | **Status** | **Availability Server side** | **Client side** |
| System | getMobileUser | see details | Available |  |  |
|  | getTableDdl | see details | Available |  |  |
|  | getTableData | see details | Available |  |  |
| ApplicantRequest | getApplicantRequest | see details | Pending | - |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  | - |  |

**OPENM4S API Functions (details)**

**These are the available methods on the server:**

|  |  |
| --- | --- |
| **Method Name** | **Status** |
| System.getMobileUser | Available |

This method will be responsible for authenticating users, it will be invoked when the tablet is online and user does the first login to get to the main menu.

**In/Out parameters**

|  |  |
| --- | --- |
| **IN (data object)** | **OUT (result object)** |
|  | total |
|  | results |
|  | code |
|  | message |

**Response**

|  |  |  |
| --- | --- | --- |
| **Response field** | **If successful** | **If failure** |
| Total | 1 | 0 |
| Results | array with users record | null |
| Code | SUCCESS | *see response codes* |
| Message |  | *see response codes* |

**Function-specific response codes**

|  |  |  |
| --- | --- | --- |
| **Code** | **Message** | **Description** |
| n/a | n/a | n/a |

**Example**

|  |
| --- |
| **URL:** <http://precisetech.dyndns.org:9002/Api/UserManagement/getOUser>  **Header:**  User-Agent: Fiddler  Host: precisetech.dyndns.org:9002  Content-Type: application/json  accept: application/json  **Content:**  {  "data": {},  "action": "System",  "method": "getMobileUser",  "username": "4-master",  "password": "4-master",  "type": "rpc",  "tid": "144"  } |

|  |  |
| --- | --- |
| **Method Name** | **Status** |
| System.getTableDdl | Available |

This method returns the table / view structure. The structure will be based on ORACLE data dictionary views. This method is generic and returns the structure based on the table name received. Use table names with capital letters.

**In/Out parameters**

|  |  |
| --- | --- |
| **IN (data object)** | **OUT (result object)** |
| tableName | total |
|  | results |
|  | code |
|  | message |

**Response**

|  |  |  |
| --- | --- | --- |
| **Response field** | **If successful** | **If failure** |
| Total | 1 | 0 |
| Results | array with table structure | null |
| Code | SUCCESS | *see response codes* |
| Message |  | *see response codes* |

**Function-specific response codes**

|  |  |  |
| --- | --- | --- |
| **Code** | **Message** | **Description** |
| TABLE\_NOT\_FOUND | Table not found | If table requested by user does not exist |

**Example**

|  |
| --- |
| **URL:** [http://precisetech.dyndns.org:9002/Api/TableOperations/](http://precisetech.dyndns.org:9002/Api/UserManagement/getMobileUser)GetTableDdl  **Header:**  User-Agent: Fiddler  Host: precisetech.dyndns.org:9002  Content-Type: application/json  accept: application/json  **Content:**  {  "data":  {  "tableName": "YM\_USERS"  }  ,  "action": "System",  "method": "getTableDdl",  "username": "4-master",  "password": "4-master",  "type": "rpc",  "tid": "144"  } |

|  |  |
| --- | --- |
| **Method Name** | **Status** |
| System.getTableData | Available |

This method is a generic method that will return records from a table based on the value of the filtering column received.

|  |  |
| --- | --- |
| **IN (data object)** | **Description** |
| ViewName |  |
| columnName | columnName must have a value, either “\*” or a comma separated list of valid columns “column1, column2, column3”. |
| filterColumnName |  |
| filterValue |  |
| pagingStart |  |
| pagingLimit | 0 if no limit |
| **OUT (result object)** |  |
| total |  |
| results |  |
| code |  |
| message |  |

**Response**

|  |  |  |
| --- | --- | --- |
| **Response field** | **If successful** | **If failure** |
| Total | 1 | 0 |
| Results | array with requested records | null |
| Code | SUCCESS | *see response codes* |
| Message |  | *see response codes* |

**Function-specific response codes**

|  |  |  |
| --- | --- | --- |
| **Code** | **Message** | **Description** |
| See API Exception |  |  |

**Example**

|  |
| --- |
| **URL:** http://precisetech.dyndns.org:9002/TCSCloud.Services.API/Api/TableData/GetTableData  **Header:**  User-Agent: Fiddler  Host: precisetech.dyndns.org:9002  Content-Type: application/json  accept: application/json  **Content:**  {  "action": "System",  "method": "getTableData",  "username": "4-master",  "password": "4-master",  "gzipped": true,  "data":  {  "tableName": "YM\_USERS",  "columnName": "\*",  "filterColumnName": "",  "filterValue": "",  "pagingStart": "0",  "pagingLimit": "0"  }  ,  "type": "rpc",  "tid": 144  } |

|  |  |
| --- | --- |
| **Method Name** | **Status** |
| PostInventory | Available |

This method is a generic method that will return records from a table based on the value of the filtering column received.

|  |  |
| --- | --- |
| **IN (data object)** | **Description** |
| InventoryTransferMaster (Must) | **create\_date, create\_user, ten\_id, invmobtra\_id, cus\_id, mob\_id\_from, mob\_id\_to, transfer\_number, last\_insert\_date, last\_insert\_username, last\_update\_username** |
| InventoryTransferMaster (Optional) | Last\_update\_date, Last\_update\_user, transfer\_asof\_date, description, notes,use\_id\_prepared\_by, use\_id\_finalized\_by, prepared\_date,finalized\_date, invmobreq\_id, transfer\_status\_id, issue\_date, receive\_date, issue\_note, receive\_note, tra\_flag, deny\_reason, ref\_ro, tracking\_no, carrier |
| InventoryTransferDetail (Must) | Create\_date, Create\_user, TEN\_ID, Invmobtradet\_id, invmobtra\_id, seq\_number, Par\_id, quantity |
| (Optional) | **last\_update\_date, last\_udpate\_user, issue\_qty, receive\_qty** |
| UserName, Name | **e.g. 4-master, 4-master** |
| Action | System |
| method | PostInventory |
| type, tid | rpc, 144 |
|  |  |
| **OUT (result object)** |  |
| total |  |
| results |  |
| code |  |
| message |  |

**Response**

|  |  |  |
| --- | --- | --- |
| **Response field** | **If successful** | **If failure** |
| Total | 1 | 0 |
| Results | array with requested records | null |
| Code | SUCCESS | *see response codes* |
| Message |  | *see response codes* |

**Function-specific response codes**

|  |  |  |
| --- | --- | --- |
| **Code** | **Message** | **Description** |
| See API Exception |  |  |

**Example**

|  |
| --- |
| **URL:** http://precisetech.dyndns.org:9002/Api/InventoryTransfer/PostInventory  **Header:**  User-Agent: Fiddler  Host: precisetech.dyndns.org:9002  Content-Type: application/json  accept: application/json  **Content:**  {  "action": "System",  "method": "PostInventory",  "username": "4-master",  "password": "4-master",  "data": [  {  "InventoryTransferMaster": [  {  "CREATE\_DATE": "2014-03-29 12:03:49",  "CREATE\_USER": "2014-03-29 12:03:49",  "LAST\_UPDATE\_DATE": "2014-03-29 12:03:49",  "LAST\_UPDATE\_USER": "",  "TEN\_ID": "",  "INVMOBTRA\_ID": "14",  "CUS\_ID": "",  "MOB\_ID\_FROM": "4-master",  "MOB\_ID\_TO": "1",  "TRANSFER\_ASOF\_DATE": "2014-03-29 12:03:49",  "TRANSFER\_NUMBER": "14",  "DESCRIPTION": "",  "NOTES": "Reason",  "TRA\_STAT\_ID": "",  "USE\_ID\_PREPARED\_BY": "",  "USE\_ID\_FINALIZED\_BY": "",  "PREPARED\_DATE": "2014-03-29 12:03:49",  "FINALIZED\_DATE": "2014-03-29 12:03:49",  "LAST\_INSERT\_USERNAME": "",  "LAST\_UPDATE\_USERNAME": "",  "LAST\_INSERT\_DATE": "2014-03-29 12:03:49"  }  ],  "InventoryTransferDetail": [  {  "CREATE\_DATE": "2014-03-29 12:03:50",  "CREATE\_USER": "",  "LAST\_UPDATE\_DATE": "2014-03-29 12:03:50",  "LAST\_UPDATE\_USER": "",  "TEN\_ID": "",  "INVMOBTRA\_ID": "14",  "INVMOBTRADET\_ID": "14",  "PART\_ID": "11033",  "QUANTITY": "6"  },  {  "CREATE\_DATE": "2014-03-29 12:03:50",  "CREATE\_USER": "",  "LAST\_UPDATE\_DATE": "2014-03-29 12:03:50",  "LAST\_UPDATE\_USER": "",  "TEN\_ID": "",  "INVMOBTRA\_ID": "14",  "INVMOBTRADET\_ID": "14",  "PART\_ID": "11033",  "QUANTITY": "6"  },  {  "CREATE\_DATE": "2014-03-29 12:03:50",  "CREATE\_USER": "",  "LAST\_UPDATE\_DATE": "2014-03-29 12:03:50",  "LAST\_UPDATE\_USER": "",  "TEN\_ID": "",  "INVMOBTRA\_ID": "14",  "INVMOBTRADET\_ID": "14",  "PART\_ID": "11033",  "QUANTITY": "6"  },  {  "CREATE\_DATE": "2014-03-29 12:03:50",  "CREATE\_USER": "",  "LAST\_UPDATE\_DATE": "2014-03-29 12:03:50",  "LAST\_UPDATE\_USER": "",  "TEN\_ID": "",  "INVMOBTRA\_ID": "14",  "INVMOBTRADET\_ID": "14",  "PART\_ID": "10988",  "QUANTITY": "6"  },  {  "CREATE\_DATE": "2014-03-29 12:03:50",  "CREATE\_USER": "",  "LAST\_UPDATE\_DATE": "2014-03-29 12:03:50",  "LAST\_UPDATE\_USER": "",  "TEN\_ID": "",  "INVMOBTRA\_ID": "14",  "INVMOBTRADET\_ID": "14",  "PART\_ID": "10988",  "QUANTITY": "6"  },  {  "CREATE\_DATE": "2014-03-29 12:03:50",  "CREATE\_USER": "",  "LAST\_UPDATE\_DATE": "2014-03-29 12:03:50",  "LAST\_UPDATE\_USER": "",  "TEN\_ID": "",  "INVMOBTRA\_ID": "14",  "INVMOBTRADET\_ID": "14",  "PART\_ID": "10988",  "QUANTITY": "6"  },  {},  {}  ]  }  ],  "type": "rpc",  "tid": 144  } |

|  |  |
| --- | --- |
| **Method Name** | **Status** |
| PostRepairOrder | Available |

This method is a generic method that will return records from a table based on the value of the filtering column received.

|  |  |
| --- | --- |
| **IN (data object)** | **Description** |
| RepairWorkOrders(Must) | create\_date ,  create\_user ,  ten\_id ,  worord\_id ,  cus\_id ,  fac\_id ,  total\_labour ,  total\_parts ,  total\_shopsupply,  total\_surcharge ,  total\_tax ,  certificate\_date ,  last\_insert\_username ,  last\_update\_username ,  last\_insert\_date |
| Must to have values | estimate\_flag IN ('Y','N')  hubo\_reading\_uom IN ('KM','MI')  wo\_source IN ('SERVER','MOBILE')  wo\_ws\_status IN ('SUCCESS','PENDING','ERROR') |
| RepairWorkOrders(Optional) |  |
| RepairWorkOrdersLabour(Must) | create\_date,  create\_user ,  ten\_id ,  worordlab\_id ,  worord\_id ,  seq\_number ,  labour\_hourly\_rate,  last\_insert\_username,  last\_update\_username,  last\_insert\_date |
| (Optional) |  |
| RepairWorkOrdersParts(Must) | create\_date ,  create\_user ,  ten\_id ,  worordpar\_id,  worordlab\_id ,  item\_cost ,  item\_charge,  last\_insert\_username,  last\_update\_username,  last\_insert\_date |
| (Optional) |  |
| UserName, Name | **e.g. 4-master, 4-master** |
| Action | System |
| method | PostRepairOrder |
| type, tid | rpc, 144 |
|  |  |
| **OUT (result object)** |  |
| total |  |
| results |  |
| code |  |
| message |  |

**Response**

|  |  |  |
| --- | --- | --- |
| **Response field** | **If successful** | **If failure** |
| Total | 1 | 0 |
| Results | array with requested records | null |
| Code | SUCCESS | *see response codes* |
| Message |  | *see response codes* |

**Function-specific response codes**

|  |  |  |
| --- | --- | --- |
| **Code** | **Message** | **Description** |
| See API Exception |  |  |

**Example**

|  |
| --- |
| **URL:** http://precisetech.dyndns.org:9002/Api/RepairOrder/PostRepairOrder  **Header:**  User-Agent: Fiddler  Host: precisetech.dyndns.org:9002  Content-Type: application/json  accept: application/json  **Content:**  {  "action": "System",  "method": "PostRepairOrder",  "username": "4-master",  "password": "4-master",  "data": [  {  "RepairWorkOrders": [  {  "SYSTEM\_NOTES": "1",  "MOBILE\_NOTES": "1",  "VOID\_NOTES": "",  "USE\_ID\_PREPARED\_BY": "1",  "USE\_ID\_APPROVED\_BY": "1",  "USE\_ID\_VOIDED\_BY": "0",  "WO\_WS\_STATUS": "PENDING",  "UNIT\_NUMBER": "351282",  "UNIT\_LICENSE\_PLATE": "K8005T",  "UNIT\_SERIAL\_VIN": "1",  "UNIT\_REEFER\_NUMBER": "",  "UNIT\_UNIMAK\_ID": "1",  "UNIT\_UNIMOD\_ID": "1",  "UNIT\_NOTES": "1",  "CERTIFICATE\_LETTER": "1",  "CERTIFICATE\_NUMBER": "1",  "CERTIFICATE\_DATE": "30-Mar-2014 4:36:52 PM",  "CERTIFICATE\_OTHER": "1",  "LAST\_INSERT\_USERNAME": "4-master",  "LAST\_UPDATE\_USERNAME": "4-master",  "LAST\_INSERT\_DATE": "30-Mar-2014 4:36:52 PM",  "CREATE\_USER": "User",  "LAST\_UPDATE\_DATE": "30-Mar-2014 4:36:52 PM",  "LAST\_UPDATE\_USER": "1",  "TEN\_ID": "1",  "WORORD\_ID": "4",  "ESTIMATE\_NUMBER": "1",  "WO\_NUMBER": "4",  "WO\_SOURCE": "1",  "REFERENCE\_NUMBER": "1",  "PO\_NUMBER": "",  "CUS\_ID": "1",  "CUS\_ID\_INVOICE\_TO": "1",  "CUSTOMER\_NAME": "",  "CUS\_ID\_WO": "0",  "CONTACT\_NAME": "1",  "WO\_LOCATION": "",  "CUSLOC\_ID": "1",  "WORORDTYP\_ID": "1",  "WORORDCAT\_ID": "1",  "WORORDUNISTA\_ID": "0",  "SHOSUP\_ID": "1",  "FAC\_ID": "1",  "WAR\_ID": "1",  "UNI\_ID": "1",  "OT\_READING": "1",  "RT\_READING": "0",  "HUBO\_READING": "1",  "REE\_ID": "1",  "HUBO\_READING\_UOM": "1",  "ESTIMATE\_FLAG": "1",  "ESTIMATE\_DATE": "30-Mar-2014 4:36:52 PM",  "ESTIMATE\_COMPLETE\_DATE": "30-Mar-2014 4:36:52 PM",  "ESTIMATE\_INT\_APPROVED\_DATE": "30-Mar-2014 4:36:52 PM",  "ESTIMATE\_CUSTOMER\_HOLD\_DATE": "30-Mar-2014 4:36:52 PM",  "ESTIMATE\_CUST\_APPROVED\_DATE": "30-Mar-2014 4:36:52 PM",  "ESTIMATE\_NOTES": "1",  "COMPLETED\_DATE": "30-Mar-2014 4:36:52 PM",  "APPROVED\_DATE": "30-Mar-2014 4:36:52 PM",  "TRANSMIT\_TIMESTAMP": "30-Mar-2014 4:36:52 PM",  "VOID\_DATE": "01-Jan-0001 12:00:00 AM",  "TOTAL\_LABOUR": "2",  "TOTAL\_PARTS": "2",  "TOTAL\_SHOPSUPPLY": "1",  "TOTAL\_SURCHARGE": "1",  "TOTAL\_TAX": "1",  "NOTES": "Note(s)"  }  ],  "RepairWorkOrderLabour": [  {  "CREATE\_DATE": "2014-03-30 11:37:03",  "CREATE\_USER": "",  "LAST\_UPDATE\_DATE": "01-Jan-0001 12:00:00 AM",  "LAST\_UPDATE\_USER": "0",  "TEN\_ID": "",  "WORORD\_ID": "4",  "SEQ\_NUMBER": "1",  "DESCRIPTION": "PERFORM ALIGNMENT",  "ESTIMATE\_HOURS": "0",  "ACTUAL\_HOURS": "1.5",  "LABOUR\_HOURLY\_RATE": "",  "VMRSYSASS\_ID": "892",  "WORORDLABPOS\_ID": "5",  "WORORDLABSHI\_ID": "2",  "NOTES": "",  "DELETED\_FLAG": "",  "OVERTIME\_FLAG": "",  "LAST\_INSERT\_USERNAME": "",  "LAST\_UPDATE\_USERNAME": "",  "LAST\_INSERT\_DATE": "30-Mar-2014 4:37:03 PM",  "WORORDLABREA\_ID": "3"  }  ],  "RepairWorkOrderParts": [  {  "LAST\_UPDATE\_USERNAME": "",  "LAST\_INSERT\_DATE": "30-Mar-2014 4:37:12 PM",  "CREATE\_DATE": "2014-03-30 11:37:12",  "CREATE\_USER": "",  "LAST\_UPDATE\_DATE": "01-Jan-0001 12:00:00 AM",  "LAST\_UPDATE\_USER": "0",  "TEN\_ID": "",  "WORORDPAR\_ID": "9",  "WORORDLAB\_ID": "5",  "SEQ\_NUMBER": "1",  "PAR\_ID": "10988",  "UOM\_ID": "7",  "ESTIMATE\_QTY": "0",  "CHARGE\_QTY": "3",  "ITEM\_COST": "555",  "ITEM\_CHARGE": "0",  "WARRANTY\_FLAG": "",  "ISSUED\_FLAG": "",  "WAR\_ID": "20",  "DELETED\_FLAG": "",  "WARRANTY\_SN": "",  "WARRANTY\_NOTES": "",  "NOTES": "",  "LAST\_INSERT\_USERNAME": ""  },  {  "LAST\_UPDATE\_USERNAME": "",  "LAST\_INSERT\_DATE": "30-Mar-2014 4:37:22 PM",  "CREATE\_DATE": "2014-03-30 11:37:22",  "CREATE\_USER": "",  "LAST\_UPDATE\_DATE": "01-Jan-0001 12:00:00 AM",  "LAST\_UPDATE\_USER": "0",  "TEN\_ID": "",  "WORORDPAR\_ID": "10",  "WORORDLAB\_ID": "5",  "SEQ\_NUMBER": "2",  "PAR\_ID": "11032",  "UOM\_ID": "7",  "ESTIMATE\_QTY": "0",  "CHARGE\_QTY": "3",  "ITEM\_COST": "555",  "ITEM\_CHARGE": "0",  "WARRANTY\_FLAG": "",  "ISSUED\_FLAG": "",  "WAR\_ID": "20",  "DELETED\_FLAG": "",  "WARRANTY\_SN": "",  "WARRANTY\_NOTES": "",  "NOTES": "",  "LAST\_INSERT\_USERNAME": ""  }  ]  }  ],  "type": "rpc",  "tid": 144  } |

|  |  |
| --- | --- |
| **Method Name** | **Status** |
| CreateRepairOrderSequence | Available |

This method is a generic method that will return records from a table based on the value of the filtering column received.

|  |  |
| --- | --- |
| **IN (data object)** | **Description** |
| RepairWorkOrder | create\_date,  create\_user,  FAC\_ID,  CUS\_ID,  TEN\_ID,  WORORD\_ID,  WO\_NUMBER,  WO\_SOURCE,  last\_insert\_username,  last\_update\_username,  wo\_ws\_status, |
|  |  |
| UserName, Name | **e.g. 4-master, 4-master** |
| Action | System |
| method | CreateRepairOrderSequence |
| type, tid | rpc, 144 |
|  |  |
| **OUT (result object)** |  |
| total |  |
| results |  |
| code |  |
| message |  |

**Response**

|  |  |  |
| --- | --- | --- |
| **Response field** | **If successful** | **If failure** |
| Total | 1 | 0 |
| Results | array with requested records | null |
| Code | SUCCESS | *see response codes* |
| Message |  | *see response codes* |

**Function-specific response codes**

|  |  |  |
| --- | --- | --- |
| **Code** | **Message** | **Description** |
| See API Exception |  |  |

**Example**

|  |
| --- |
| **URL:** http://precisetech.dyndns.org:9002/Api/RepairOrder/PostRepairOrder  **Header:**  User-Agent: Fiddler  Host: precisetech.dyndns.org:9002  Content-Type: application/json  accept: application/json  **Content:**  {  "action": "System",  "method": "CreateRepairOrderSequence",  "username": "4-master",  "password": "4-master",  "data": [  {  "RepWorkOrder": [  {  "create\_date": "30-Mar-2014 4:36:52 PM",  "create\_user": "4-master",  "FAC\_ID": "1",  "CUS\_ID": "4",  "TEN\_ID": "2",  "WORORD\_ID": "",  "WO\_NUMBER": "",  "WO\_SOURCE": "MOBILE",  "last\_insert\_username": "4-master",  "last\_update\_username": "4-master",  "last\_insert\_date": "30-Mar-2014 4:36:52 PM",  "wo\_ws\_status": "PENDING",  }  ]  }  ],  "type": "rpc",  "tid": 144  } |

|  |  |
| --- | --- |
| **Method Name** | **Status** |
| Admin.assignBarcodeToUniId | Available |

Method that will be called when an barcode assignment is being done. Validations for this method are:

Unit has already a different barcode.

Barcode is assigned to another unit.

|  |  |
| --- | --- |
| **IN (data object)** | **OUT (result object)** |
| uniId | total |
| barcodeNumber | results |
|  | code |
|  | message |

**Response**

|  |  |  |
| --- | --- | --- |
| **Response field** | **If successful** | **If failure** |
| Total | 1 | 0 |
| Results | barcode | null |
| Code | SUCCESS | *see response codes* |
| Message |  | *see response codes* |

**Function-specific response codes**

|  |  |  |
| --- | --- | --- |
| **Code** | **Message** | **Description** |
| UNIT\_WITH\_DIFFERENT\_BARCODE | Unit has a different barcode |  |
| BARCODE\_ASSIGNED\_TO\_ANOTHER\_UNIT | Barcode is assigned to another unit |  |
| SUCCESS | Update successfully. |  |

|  |
| --- |
| **Query Example:**  {  "action":"Admin",  "method":"assignBarcodeToUniId",  "username":"4-master",  "password":"4-master",  "data":[{  "uniId":740  , "barcodeNumber":"T007590"  }],  "type":"rpc",  "tid":144  } |

**API’s Meta Data**

|  |  |  |
| --- | --- | --- |
| **API Name** | **Purpose** | **Content (Input/Output)** |
| getMobileUser | To fetch user information from server and to authenticate locally.Table will be YM\_USERS. | Input: no Output: User List from user table. |
| getTableDdl | To fetch structure of View/ Table from server. Table/ View will be any. | Input: TableName/ ViewName Output: total, results (View/Table Structure), code, message |
| getTableData | To fetch data of View/ Table from server. Table/ View will be any. | Input: ViewName, columnName, filterColumnName, filterValue, pagingStart, pagingLimit Output: total, results (View/Table Data), code, message |
|  |  |  |
|  |  |  |
|  |  |  |

# 

# 

# 

# 

# **R****eferences**