**Version 0.3.1**

**Updated 6/17/2017**

**About Server HTTP API**

The Server HTTP API provides developers with a standard interface to and response from the server.

The Server HTTP API requires knowledge of the html programming language and a familiarity with json.

**Using Server HTTP API**

Client side html files will send a request to the server in a standard get format. The server will return a json file with the requested information.

Server runs on - localhost:4000

**Server HTTP API Version 1.0**

The client should send GET requests and POST updates. The following Each GET or POST is paired with its json response:

/billofmaterials

/lineitems

/manufacturers

/parts

/productionruns

/products

**Bill of Materials:**

|  |  |
| --- | --- |
| **url extensions for localhost:4000/billofmaterials** | **Return Response** |
|  |  |
| / | returns list of bill of materials |
| /bom\_id\_all/:bom\_id | return all bom\_id info by bom\_id |
| /lineitem\_all/:line\_item\_id | return all lineitem info by lineitem\_id |
| /get\_product/:bom\_id | return product info |
| /get\_version/:bom\_id | return version info |
| /:version\_id | returns specific bill of materials |
| /new/:product\_id | creates new bom |
| /new\_bom\_version/:pro\_id/:version | creates new bom with product and version |
| /add\_lineitem/:bom\_id/:quantity/:part\_id | updates by adding lint item with quantity and part\_id |
| /update\_lineitem/:bom\_id/:line\_item\_id/:part/:quantity | Updates line item id in bom |
| /delete\_lineitem/:bom\_id | Deletes all line items in BOM Please check its post cmd not delete |
| /update\_version\_id/:bom\_id/:version\_id | updates -- index bom\_id to update version\_id |
| /update\_product/:bom\_id/:product\_id | updates -- index bom\_id to update product |
| /insert\_product/:product\_id/:version | INSERT new bom INTO billofmaterials (product,version\_id) |
| /:bom\_id | delete bom |

**Line Items:**

|  |  |
| --- | --- |
| **url extensions for localhost:4000/lineitems** | **Return Response** |
|  |  |
| / | return list of lineitems |
| /:bom\_id | return list of lineitems where bom\_id = something |
| /bom\_part\_lineitem\_id/:line\_item\_id | return specific item info |
| /bom\_id\_all/:bom\_id | return all bom\_id info |
| /lineitem\_all/:line\_item\_id | return all lineitem info |
| /get\_bom/:line\_item\_id | return bom by lineitem |
| /get\_part/:line\_item\_id | return part |
| /get\_quantity/:line\_item\_id | return quantity info |
| /new/:one | create new lineitem |
| /bom\_part\_quantity/:bom/:part/:quantity | create new bom\_part\_quantity in lineitem |
| /update\_part/:line\_item\_id/:part | updates -- index line\_item\_id to update part |
| /update\_quantity/:line\_item\_id/:quantity | updates -- index line\_item\_id to update quantity |
| /:line\_item\_id | Delete line item |
|  |  |

**Manufacturers:**

|  |  |
| --- | --- |
| **url extensions for localhost:4000/manufacturers** | **Return Response** |
|  |  |
| / | return list of manufacturers |
| /mfr\_id/:mfr\_id | return manufacturer details by mfr\_id |
| /mfr\_name/:mfr\_name | insert manufacturer into database using name |
| /update\_name/:name/:manufacturer\_id | update a Manufacturer |
| /:manufacturer\_id | delete by manufacturer id |
|  |  |

**Parts:**

|  |  |
| --- | --- |
| **url extensions for localhost:4000/parts** | **Return Response** |
|  |  |
| / | return list of parts |
| /:part\_id | return all from part by part\_id |
| /mfrid/:mfr\_id | return all with mfr-id |
| /new/:part\_name/:mfr\_id | insert part into database using name |
| /new/:part\_name/:value/:units/:mfrprtnum/:mfr\_id/:cost/:time/:qty | insert part into database using all |
| /update\_all/:prt\_id/:part\_name/:value/:units/:mfrprtnum/:mfr\_id/:cost/:time/:qty | Update part into database using all |
| /update\_description/:part\_id/:description | updates – description |
| /update\_value/:part\_id/:value | updates – value |
| /update\_units/:part\_id/:units | Updates units |
| /update\_mfrPartNumber/:part\_id/:mfrpartnumber | Updates mfrPartNumber |
| /update\_manufacturer/:part\_id/:manufacturer | Updates manufacturer |
| /update\_standardCost/:part\_id/:standard cost | updates -- standardCost |
| /update\_standardLeadTime/:part\_id/:leadtime | pdates -- standardLeadTime |
| /update\_quantityOnHand/:part\_id/:quantyonhand | updates -- quantityOnHand |
| /:part\_id | Delete part |
|  |  |

**Production Runs:**

|  |  |
| --- | --- |
| **url extensions for localhost:4000/productionruns** | **Return Response** |
|  |  |
| / | return list of productionruns |
| /:prd\_id | return all from production run by id |
| /new/:pro\_id | create new productionRun |
| /update\_all/:prd\_id/:pro/:qun/:start/:end/:status | updates – all |
| /update\_product/:production\_run\_id/:product | updates – product |
| /update\_quantity/:production\_run\_id/:quantity | updates – quantity |
| /update\_targetStartDate/:production\_run\_id/:startdate | updates -- targetStartDate DATE |
| /update\_quantity/:production\_run\_id/:enddate | updates -- targetCompleteDate DATE |
| /:production\_run\_id | Deletes produtionrun |
|  |  |

**Products:**

|  |  |
| --- | --- |
| **url extensions for localhost:4000/products** | **Return Response** |
|  |  |
| / | return list of products |
| /:pdt\_id | return specific product info |
| /new/:productName | Create new product |
| /new\_all/:productName/:standardSalesPrice/:productDesignation | Create new product with all |
| /update\_product/:product\_id/:productName/:standardSalesPrice/:productDesignation | Update product |
| /update\_productname/:product\_id/:productName | update product name |
| /update\_salesprice/:product\_id/:standardSalesPrice | update product salesprice |
| /update\_designation/:product\_id/:productDesignation | update product designation |
| /: product\_id | delete a product |
|  |  |

The server will return the following json files:

ProductionRun:

{

"product":

{

"productDesignation": "Designation of the product",

"productName":"Name of the product",

"standardSalesPrice" : double

},

"quantity" : int,

"targetStartDate" : Date,

"targetCompleteDate" : Date,

"status":"RunStatus"

}

Product:

{

"productDesignation": "Designation of the product",

"productName":"Name of the product",

"standardSalesPrice" : double

}

BillOfMaterials:

{

"product":

{

"productDesignation": "Designation of the product",

"productName":"Name of the product",

"standardSalesPrice" : double

},

"version" : int

"items" : [

{“bom”,

“part”,

“quantity”

}]

}

LineItem:

{

"bom":

{

"product":

{

"productDesignation": "Designation of the product",

"productName":"Name of the product",

"standardSalesPrice" : double

},

"version" : int

"items" : [

{“bom”,

“part”,

“quantity”

}]

},

"part":" “Name of Part”,

"quantity":" int

}

Part:

{

"description" : "Description of part",

"value" : double,

"units" : "Number of Units",

"mfrPartNumber":"Manufacturers part number",

"manufacturer":

{

"name":"Name of Manufacturer”

},

"standardCost": double,

"standardLeadTime":int,

"quantityOnHand":int

}

Manufacturer:

{

"name":"Name of Manufacturer”

}