**Purchase Order API**

The Purchase Order API will be implemented using 3 Classes

* /api/purchase\_orders/
* /api/purchase\_orders/{po\_id}/
* /api/purchase\_orders/{po\_id}/acknowledge

**/api/purchase\_orders/**

This class will handle GET & POST Requests.

* GET – To List all the Purchase Orders
* POST – To Create an New Purchase Order

**/api/purchase\_orders/{po\_id}/**

This class will handle GET, PUT & DELETE Requests.

* GET – To get information on an specific Purchase Order
* PUT – To Update the Purchase Order
* DELETE – To DELETE an Purchase Order ( Softly )

**/api/purchase\_orders/{po\_id}/acknowledge**

This class will only handle POST Request for the Vendor to acknowledge the Purchase Oder given to him

* POST – To Acknowledge to Purchase Order

Note :- Implement the Functionality that the intended Vendor is acknowledging the Purchase order, use Token for the purpose

**Purchase Order Model Fields**

The Purchase Order Model will have several fields to implement a successful vendor management system.

* Po\_Number – An Unique Number Identifying the purchase order
* Vendor – Foreign key Linking the Purchase order with the Vendor
* Order\_Date – The date on which the order was made
* Delivery\_Date – The Date on which the order was successfully delivered
* Items – An Chat Field Consist of JSON Grammer which gives the details of the items In the Purchase Order (There could be more than 1 item in PO)
* Quantity – The No. of individual Items in the PO
* Status – The Status of the Purchase Order
* Quality\_Rating – An Floating Point rating given by the Customer for the purchase order (Lets keep it from 1 to 10 or the Customer can give None)
* Issue\_Date – The Date on which the Purchase was given to the Vendor ( This will be set to the same date as Order\_Date )
* Acknowledge\_Date – The Date on which the vendor said “YES” to the Purchase order
* Delete – Soft Deletion of the Purchase Order