asap SYSTEMS

API Guide

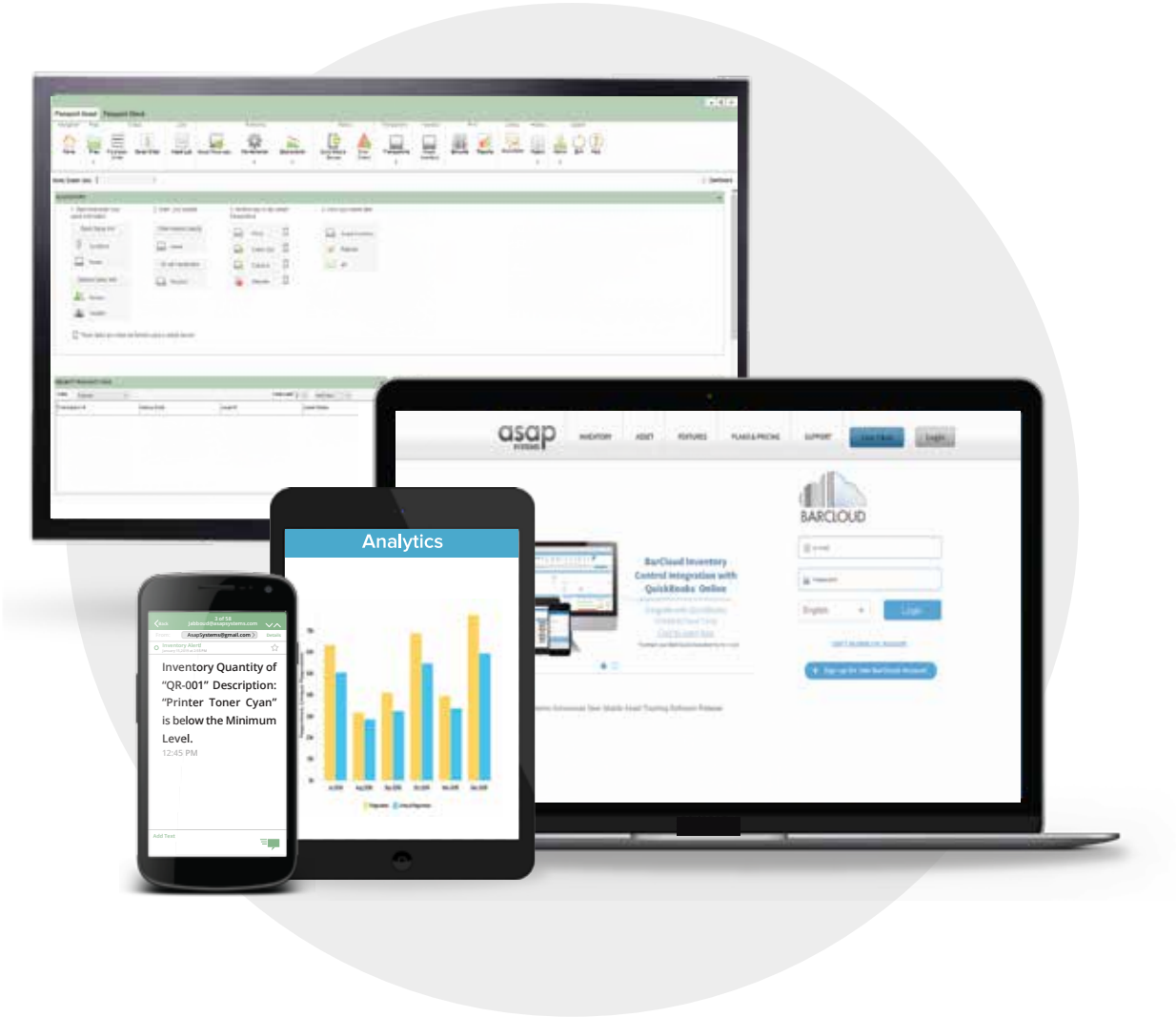
Develop your own Custom Inventory Management & Asset Tracking Apps

BarCloud offers developers a platform for a multitude of uses for custom integration, user interfaces, added features and functionality.

BarCloud API Guide

BarCloud offers developers a platform for a multitude of uses for custom integration, user interfaces, added features and functionality. For example you may need to integrate with an ERP system, provide an application specific or user specific user interface, externally crunch data and place the result back into BarCloud, or provide data analytics, charts and graphs.

This API provides an overview of the building blocks for building apps that are compatible with the BarCloud inventory management and asset tracking system.



API Objects

LoginInfo

Property Name	Type	Required	Notes
ApplicationKey	String	Yes	
CustomerKey	String	Yes	
Username	String	Yes	
Password	String	Yes	

UOM

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
UnitCode	String	Yes	Unique in the System
UnitName	String	Yes	
Multiplier	Decimal	No	
CreateDate	DateTime	No	Default will use current DateTime
CreatedBy	String	Yes	Will use Current user of the passed credentials
LastModifyDate	Date Time	No	Default will use current DateTime
LastModifyOperator	String	Yes	Will use Current user of the passed credentials

AssetMoveTransaction

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
LinteItems	List	Yes	
Notes	Nvarchar(500)	No	

AssetModel

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
ModelNumber	Nvarchar(70)	Yes	Unique in the System
ModelName	Nvarchar(70)	No	
AssetModelType	Nvarchar(70)	No	References the Asset Model Type file
Category	Nvarchar(100)	No	
Cost	Decimal	No	Default zero if not specified
Manufacturer	Nvarchar(100)	No	
Notes	Nvarchar(500)	No	
Description	Nvarchar(255)	No	
Deleted	Boolean	No	Default false
CreateDate	DateTime	No	Default will use current DateTime
CreatedBy	Nvarchar(60)	Yes	Will use Current user of the passed credentials
LastModifyDate	Date Time	Yes	Default will use current DateTime
LastModifyOperator	Nvarchar(60)	Yes	Will use Current user of the passed credentials

AssetReceiveLineItems

Property Name	Type	Required	Notes
AssetModelNumber	Nvarchar(70)	No	References the Asset Model file
AssetNumber	Nvarchar(30)	Yes	Unique in the System
AssetName	Nvarchar(100)	No	
Cost	Decimal	No	Defaults to zero
SerialNumber	Nvarchar(50)	No	
Asset Notes	Nvarchar(500)	No	Will be applied to the Asset

Asset

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
AssetNumber	Nvarchar(30)	Yes	Unique in the System
Name	Nvarchar(100)	No	
SiteName	Nvarchar(70)	No	References the Site file
AssetModel	Nvarchar(70)	No	References the Asset Model file
PurchaseType	Nvarchar(50)	Yes	Default Purchase if not specified, references the Purchase Type file
SerialNumber	Nvarchar(50)	No	
IsSerialUnique	Boolean	No	
Condition	Nvarchar(50)	No	Referemces the Condition file
Custodian	Nvarchar(50)	No	References the Person file
HomeLocation	Nvarchar(50)	Yes	Default CurrentLocation if not specified, references the Location file
CurrentLocation	Nvarchar(50)	Yes	References the Location file
PurchaseOrderNumber	Nvarchar(50)	No	
PurchaseDate	DateTime	No	
DepreciationMethod	Nvarchar(50)	No	References the Depreciation Method file
YearsToDepreciate	Decimal	No	
Cost	Decimal	No	Default zero if not specified
ScrapValue	Decimal	No	
LeaseFrom	Nvarchar(70)	No	References the Supplier file
LeaseStartDate	DateTime	No	
LeaseEndDate	DateTime	No	
LeaseNumber	Nvarchar(50)	No	

Asset

Property Name	Type	Required	Notes
WarrantyNumber	Nvarchar(50)	No	
WarrantyStartDate	DateTime	No	
WarrantyEndDate	DateTime	No	
MaintenanceContractSupplier	Nvarchar(70)	No	References the Site file
MaintenanceContractCost	Decimal	No	
MaintenanceContractStartDate	DateTime	No	
MaintenanceContractEndDate	DateTime	No	
Manufacturer	Nvachar(100)	No	
Height	Decimal	No	
Length	Decimal	No	
Width	Decimal	No	
Size	Nvarchar(30)	No	
Color	Nvarchar(30)	No	
Resolution	Nvarchar(30)	No	
ScreenSize	Nvarchar(30)	No	
VIN	Nvarchar(30)	No	
LicensePlate	Nvarchar(20)	No	
Mileage	Decimal	No	
SoftwareVersion	Nvarchar(70)	No	
SoftwareLicenses	Nvarchar(70)	No	
CPU	Nvarchar(50)	No	
CPUSpeed	Nvarchar(50)	No	

Asset

Property Name	Type	Required	Notes
HDSize	Nvarchar(50)	No	
IPAddress	DateTime	No	
OperatingSystem	DateTime	No	
BIOS	Nvarchar(50)	No	
RAM	Nvarchar(50)	No	
VideoCard	Nvarchar(50)	No	
MACAddress	Nvarchar(50)	No	
Insured	Boolean	No	
Authorized	Boolean	No	
OvertimeStartAfter	Int	No	
OvertimeInterval	Nvarchar(60)	No	References the Time Increment file
StdHourlyRate	Decimal	No	
BillingIncrement	Int	No	
billingIncrementInterval	Nvarchar(60)	No	References the Time Increment file
OverTimeHourlyRate	Decimal	No	
Notes	Nvarchar(500)	No	
Description	Nvarchar(500)	No	
Deleted	Boolean	No	Default false
DisposeDate	DateTime	No	
DisposeToId	Nvarchar(100)	No	References the Dispose To file
DisposeOperator	Nvarchar(60)	No	
CreateDate	DateTime	No	Default will use current DateTime

Asset

Property Name	Type	Required	Notes
CreatedBy	Nvarchar(60)	Yes	Will use Current user of the passed credentials
LastModifyDate	DateTime	Yes	Default will use current DateTime
LastModifyOperator	DateTime	Yes	Will use Current user of the passed credentials
CheckedOut	Boolean	No	

Customer

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
Code	Nvarchar(50)	Yes	Unique in the System
Name	Nvarchar(150)	No	
SiteName	Nvarchar(70)	No	References the Site file
CustomerType	Nvarchar(100)	Yes	Defaults to Company if not specified
Address1	Nvarchar(100)	No	
Address2	Nvarchar(100)	No	
City	Nvarchar(100)	No	
State	Nvarchar(100)	No	
Zip	Nvarchar(100)	No	
Country	Nvarchar(100)	No	
Phone1	Nvarchar(100)	No	
Phone2	Nvarchar(100)	No	
Fax	Nvarchar(100)	No	
Email	Nvarchar(100)	No	

Customers

Property Name	Type	Required	Notes
Website	Nvarchar(100)	No	
Contact	Nvarchar(100)	No	
ContactPhone1	Nvarchar(100)	No	
ContactPhone2	Nvarchar(100)	No	
ContactEmail	Nvarchar(100)	No	
ContactFax	Nvarchar(100)	No	
Make	Nvarchar(100)	No	
IModel	Nvarchar(100)	No	
Color	Nvarchar(100)	No	
VIN	Nvarchar(100)	No	
LicensePlate	Nvarchar(100)	No	
ContractWith	Nvarchar(50)	No	References the Customer File
ContractDate	DateTime	No	
ContractExpirationDate	DateTime	No	References the Time Increment file
Notes	Nvarchar(500)	No	
CreateDate	DateTime	Yes	Default will use current DateTime
CreatedBy	Nvarchar(60)	Yes	Will use Current user of the passed credentials
LastModifyDate	DateTime	Yes	Default will use current DateTime
LastModifyOperator	Nvarchar(60)	Yes	Will use Current user of the passed credentials
Deleted	Boolean	No	Default false

Location

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
Code	Nvarchar(50)	Yes	Unique in the System
Name	Nvarchar(100)	No	
SiteName	Nvarchar(70)	Yes	Default Site1, references the Site file
LocationType	Nvarchar(60)	No	References the Location Type file
AddressName	Nvarchar(100)	No	References the Address file
Building	Nvarchar(50)	No	
Floor	Nvarchar(30)	No	
Area	Nvarchar(60)	No	
Shelf	Nvarchar(60)	No	
Room	Nvarchar(20)	No	
Aisle	Nvarchar(30)	No	
Bin	Nvarchar(30)	No	
Make	Nvarchar(30)	No	
Model	Nvarchar(30)	No	
Color	Nvarchar(30)	No	
VIN	Nvarchar(30)	No	
LicensePlate	Nvarchar(30)	No	
HomeLocation	Nvarchar(50)	No	References the Location file
CurrentLocation	Nvarchar(50)	No	References the Location file
PaletNumber	Nvarchar(20)	No	
Notes	Nvarchar(500)	No	

Location

Property Name	Type	Required	Notes
CreateDate	DateTime	Yes	Default will use current DateTim
CreatedBy	Nvarchar(60)	Yes	Will use Current user of the passed credentials
LastModifyDate	DateTime	Yes	Default will use current DateTime
LastModifyOperator	Nvarchar(60)	Yes	Will use Current user of the passed credentials
Deleted	Boolean	No	Default false

Department

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
Code	Nvarchar(50)	Yes	Unique in the System
Name	Nvarchar(100)	No	
Deleted	Boolean	No	
CreateDate	DateTime	No	Default will use current DateTime
CreatedBy	Nvarchar(60)	Yes	Will use Current user of the passed credentials
LastModifyDate	DateTime	Yes	Default will use current DateTime
LastModifyOperator	Nvarchar(60)	Yes	Will use Current user of the passed credentials

Person

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
Code	Nvarchar(50)	Yes	Unique in the System
Name	Nvarchar(100)	No	
SiteName	Nvarchar(70)	No	References the Site file

Person

Property Name	Type	Required	Notes
FirstName	Nvarchar(100)	No	
LastName	Nvarchar(100)	No	
Active	Boolean	No	Default value true
Title	Nvarchar(100)	No	
HireDate	DateTime	No	
LeaveDate	DateTime	No	
Department	Nvarchar(50)	No	References the Department file
Manager	Nvarchar(50)	No	Refrences the person file
Address1	Nvarchar(100)	No	
Address2	Nvarchar(100)	No	
City	Nvarchar(100)	No	
State	Nvarchar(100)	No	
Zip	Nvarchar(100)	No	
Country	Nvarchar(100)	No	
Phone1	Nvarchar(100)	No	
Phone2	Nvarchar(100)	No	
SMSAddress	Nvarchar(100)	No	
Email	Nvarchar(100)	No	
Notes	Nvarchar(500)	No	
CreateDate	DateTime	Yes	Default will use current DateTime
CreatedBy	Nvarchar(60)	Yes	Will use Current user of the passed credentials
LastModifyDate	DateTime	Yes	Default will use current DateTime

Person

Property Name	Type	Required	Notes
LastModifyOperator	Nvarchar(60)	Yes	Will use Current user of the passed credentials
Deleted	Boolean	No	Default false

AssetReceiveTransaction

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
PurchaseOrderNumber	Nvarchar(30)	No	References the Purchase Order file
LocationTo	Nvarchar(50)	Yes	References the Location file
SiteName	Nvarchar(70)	No	References the Site file
PersonTo	Nvarchar(50)	No	References the Person file
LineItems	List	Yes	
Notes	Nvarchar(250)	No	

AssetMoveTransactionLineItems

Property Name	Type	Required	Notes
AssetNumber	Nvarchar(30)	Yes	Refrences the Asset file
LocationTo	Nvarchar(50)	Yes	References the Location file
CustodianTo	Nvarchar(50)	No	References the Person file

AssetDisposeTransaction

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
LinteItems	List	Yes	
Notes	Nvarchar(500)	No	

AssetDisposeTransactionLineItems

Property Name	Type	Required	Notes
AssetNumber	Nvarchar(30)	Yes	Refrences the Asset file
DisposeTo	Nvarchar(100)	Yes	References the Dispose Tofile

Supplier

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
Code	Nvarchar(70)	Yes	Unique in the System
Name	Nvarchar(100)	No	
SiteName	Nvarchar(70)	No	References the Site file
CertificationDate	DateTime	No	
ContractNumber	Nvarchar(100)	No	
Address1	Nvarchar(100)	No	
Address2	Nvarchar(100)	No	
City	Nvarchar(100)	No	
State	Nvarchar(100)	No	
Zip	Nvarchar(100)	No	
Country	Nvarchar(100)	No	
Phone1	Nvarchar(100)	No	
Phone2	Nvarchar(100)	No	
Fax	Nvarchar(100)	No	
Email	Nvarchar(100)	No	
Website	Nvarchar(100)	No	
Contact	Nvarchar(100)	No	

Supplier

Property Name	Type	Required	Notes
ContactPhone1	Nvarchar(100)	No	
ContactPhone2	Nvarchar(100)	No	
ContactEmail	Nvarchar(100)	No	
ContactFax	Nvarchar(100)	No	
Notes	Nvarchar(500)	No	
CreateDate	DateTime	Yes	Default will use current DateTime
CreatedBy	Nvarchar(60)	Yes	Will use Current user of the passed credentials
LastModifyDate	DateTime	Yes	Default will use current DateTime
LastModifyOperator	Nvarchar(60)	Yes	Will use Current user of the passed credentials
Deleted	Boolean	No	Default false

AssetCheckoutTransaction

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
LinteItems	List	Yes	
Notes	String	No	

AssetCheckoutTransactionLineItem

Property Name	Type	Required	Notes
AssetNumber	String	Yes	
CheckoutToLocation	String	Yes	Could be omitted if CheckoutToCustodian is provided
CheckoutToCustodian	String	Yes	Could be omitted if CheckoutToLocation is provided
DueDays	Int	Yes	

AssetCheckoutTransactionLineItem

Property Name	Type	Required	Notes
DueDate	DateTime	Yes	
Condition	String	No	
Notes	String	No	

AssetCheckinTransaction

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
LinteItems	List	Yes	
Notes	String	No	

AssetCheckinTransactionLineItem

Property Name	Type	Required	Notes
AssetNumber	String	Yes	
CheckoutToLocation	String	Yes	Could be omitted if CheckoutToCustodian is provided
CheckoutToCustodian	String	Yes	Could be omitted if CheckoutToLocation is provided
Condition	String	No	
Notes	String	No	

AssetPurchaseOrder

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
PONumber	String	Yes	Unique in the System
SiteName	String	Yes	
AssetModel	String	Yes	

AssetPurchaseOrder

Property Name	Type	Required	Notes
OrderDate	DateTime	Yes	
Supplier	String	Yes	
ShipTo	String	No	
ExpectedDeliveryDate	DateTime	No	
DueDate	DateTime	No	
OrderStatus	DateTime	Yes	
Note	String	No	
Deleted	Bool	No	Default false
IsClosed	Bool	No	Default false
SubTotal	Decimal	No	Default zero
Discount	Decimal	No	Default zero
OtherAdjustment	Decimal	No	Default zero
Tax	Decimal	No	Default zero
Shipping	Decimal	No	Default zero
LineItems	List	Yes	
CreateDate	DateTime	No	Default will use current DateTime
CreatedBy	String	Yes	Will use Current user of the passed credentials
LastModifyDate	DateTime	Yes	Default will use current DateTime
LastModifyOperator	String	Yes	Will use Current user of the passed credentials

AssetPurchaseOrderLiteItem

Property Name	Type	Required	Notes
LineNumber	String	Yes	

AssetPurchaseOrderLiteItem

Property Name	Type	Required	Notes
AssetModel	String	Yes	
IsTaxable	String	No	Default false
UnitCost	Decimal	No	Default zero
UnitsOrdered	Integer	Yes	
ExtendedCost	Decimal	No	Default zero
UnitsReceived	Integer	No	Default zero
Deleted	Bool	No	Default false

BarCloudUser

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
Username	String	Yes	Unique in the System
Password	String	Yes	
Email	String	Yes	
Person	String	Yes	
Offset	String	Yes	Users Time zone
SecurityLevel	String	Yes	
Deleted	Bool	No	Default false
Active	Bool	No	Default true
CreateDate	DateTime	No	Default will use current DateTime
CreatedBy	String	Yes	Will use Current user of the passed credentials
LastModifyDate	DateTime	Yes	Default will use current DateTime
LastModifyOperator	String	Yes	Will use Current user of the passed credentials

AssetMaintenanceScheduleOneTime

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
MaintenanceCode	String	Yes	Unique in the System
AssetNumber	String	Yes	
MaintenanceType	String	Yes	
ScheduledFor	DateTime	No	
ScheduledOn	DateTime	No	
Person	String	No	
LastPerformed	DateTime	No	
Notes	String	No	
SiteName	String	No	Default Site1
IsPerformed	Bool	No	Default false
CreateDate	DateTime	No	Default will use current DateTime
CreatedBy	String	No	Will use Current user of the passed credentials
LastModifyDate	DateTime	No	Default will use current DateTime
LastModifyOperator	String	No	Will use Current user of the passed credentials

AssetMaintenanceScheduleRecurring

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
MaintenanceCode	String	Yes	Unique in the System
AssetNumber	String	Yes	
MaintenanceType	String	No	
ScheduledOn	DateTime	No	

AssetMaintenanceScheduleRecurring

Property Name	Type	Required	Notes
TimeUnit	String	Yes	Default MonthsDefault Months
Frequency	Int	Yes	
DueNext	DateTime	No	Default Current DateTime
LastPerformed	DateTime	No	
StartDate	DateTime	No	Default Current DateTime
EndDate	DateTime	No	Default A year from Current DateTime
Notes	String	No	
SiteName	String	No	Default Site1
ReccuringBasis	DateTime	No	For recurring Maintenance [First Due Next is based on this value as the start value]
ScheduledBased	Bool	No	default true [due next is based on scheduled date not actual performed date]
IsPerformed	Bool	No	Default false
CreateDate	DateTime	No	Default will use current DateTime
CreatedBy	String	No	Will use Current user of the passed credentials
LastModifyDate	DateTime	No	Default will use current DateTime
LastModifyOperator	String	No	Will use Current user of the passed credentials

AssetPerformedMaintenance

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
PerformMaintenanceCode	String	Yes	Unique in the System
Person	String	Yes	Peformed By
PerformDate	DateTime	Yes	Default Current DateTime
LaborCost	Decimal	No	

AssetPerformedMaintenance

Property Name	Type	Required	Notes
PartsCost	Decimal	No	
MaintenanceAssetSchedule	String	No	
PerformNonDate	Int	No	

Stock Inventory

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
StockNumber	Nvarchar(100)	No	Unique in the System
Description	Nvarchar(200)	No	Stock number Description
SiteName	Nvarchar(70)	No	Site Name of the site where the inventory is located
InventoryLocation	Nvarchar(50)	No	Location where the inventory is located
InventoryLocationDescription	Nvarchar(100)	No	Description of the Location where the inventory is located
InventoryNumber	Nvarchar(100)	No	Inventory Number, Different from stock Number for inventory types other than Standard.
Availability	Decimal	No	Quantity in Stock

StockIssueTransaction

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
CustomerCode	String	Yes	
SalesOrderNumber	String	No	
LinteItems	List	Yes	
Notes	String	No	

StockIssueTransactionLineItem

Property Name	Type	Required	Notes
InventoryNumber	String	Yes	
LocationFrom	String	Yes	
Price	String	No	Default zero
QuantityToIssue	Int	Yes	
SalesOrderLineNumber	Int	Yes	Required if issueing out against a Sales Order

HistoryRequestObject

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
Page	Int	No	Default 1
PageSize	Int	No	Readonly set to 20 per page
SqlFilter	String	No	Separate values with spaces for example Cost = 100 AND InvHistoryType = 'Receive'
Notes	String	No	

AssetAllHistory [AssetBaseHistoryFields + Below Fields] (Read Only Values)

Property Name	Type	Required	Notes
InvHistoryType	String		
LocationMoveFrom	String		
LocationMoveTo	String		
PersonMoveFrom	String		
PersonMoveTo	String		
DisposeTo	String		

AssetBaseHistoryFields (Read Only Values)

Property Name	Type	Required	Notes
TotalRecords	Int		
TransactionNumber	String		
HistoryDateTime	DateTime		
AssetNumber	String		
AssetModelNumber	String		
AssetModelName	String		
PurchaseType	String		
LeaseFrom	String		
LeaseNumber	String		
LeaseStartDate	DateTime		
LeaseEndDate	DateTime		
Cost	Decimal		
PurchaseDate	DateTime		
AssetType	String		
Description	String		
Manufacturer	String		
Height	Decimal		
Length	Decimal		
Width	Decimal		
Size	String		
Color	String		
Resolution	String		

AssetBaseHistoryFields (Read Only Values)

Property Name	Type	Required	Notes
ScreenSize	String		
VIN	String		
LicensePlate	String		
Mileage	Int		
SoftwareVersion	String		
SoftwareLicenses	String		
Notes	String		
Category	String		
SerialNumber	String		
WarrantyNumber	String		
WarrantyStart	DateTime		
WarrantyEnd	DateTime		
MaintenanceContractSupplier	String		
MaintenanceContractCost	String		
MaintenanceContractStartDate	DateTime		
MaintenanceContractEndDate	DateTime		
Condition	String		
PONumber	String		
Insured	Bool		
BIOS	String		
CPU	String		
CPUSpeed	String		

AssetBaseHistoryFields (Read Only Values)

Property Name	Type	Required	Notes
HDSize	String		
IPAddress	String		
MAC	String		
OperatingSystem	String		
RAM	String		
VideoCard	String		
SiteName	String		

AssetCheckinHistory [AssetBaseHistoryFields + Below Fields] (Read Only Values)

Property Name	Type	Required	Notes
LocationMoveFrom	String		
LocationMoveTo	String		
PersonMoveFrom	String		
PersonMoveTo	String		

AssetCheckoutHistory [AssetBaseHistoryFields + Below Fields] (Read Only Values)

Property Name	Type	Required	Notes
LocationMoveFrom	String		
LocationMoveTo	String		
PersonMoveFrom	String		
PersonMoveTo	String		
DueDate	DateTime		

AssetMoveHistory [AssetBaseHistoryFields + Below Fields] (Read Only Values)

Property Name	Type	Required	Notes
LocationMoveFrom	String		
LocationMoveTo	String		
PersonMoveFrom	String		
PersonMoveTo	String		

AssetReceiveHistory [AssetBaseHistoryFields + Below Fields] (Read Only Values)

Property Name	Type	Required	Notes
PurchaseOrderDate	DateTime		
Supplier	String		
Location	String		

StockPurchaseOrder

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
SONumber	String	Yes	Unique in the System
SiteName	String	Yes	
StockNumber	String	Yes	
OrderDate	DateTime	Yes	
Supplier	String	Yes	
ShipTo	String	No	
ExpectedDeliveryDate	DateTime	No	
DueDate	DateTime	No	
OrderStatus	String	Yes	

StockPurchaseOrder

Property Name	Type	Required	Notes
Note	String	No	
Deleted	Bool	No	Default false
IsClosed	Bool	No	Default false
SubTotal	Decimal	No	Default zero
Discount	Decimal	No	Default zero
OtherAdjustment	Decimal	No	Default zero
Tax	Decimal	No	Default zero
Shipping	Decimal	No	Default zero
LineItems	Decimal	No	
CreateDate	List	No	Default will use current DateTime
CreatedBy	String	Yes	Will use Current user of the passed credentials
LastModifyDate	String	Yes	Default will use current DateTime
LastModifyOperator	String	Yes	Will use Current user of the passed credentials

StockPurchaseOrderLiteItem

Property Name	Type	Required	Notes
LineNumber	String	Yes	
StockNumber	String	Yes	
IsTaxable	String	No	Default false
UnitCost	Decimal	No	Default zero
UnitsOrdered	Integer	Yes	
ExtendedCost	Decimal	No	Default zero
UnitsReceived	Integer	No	Default zero

StockPurchaseOrder

Property Name	Type	Required	Notes
Deleted	Bool	No	Default false

StockSalesOrder

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
SONumber	String	Yes	Unique in the System
SiteName	String	Yes	
StockNumber	String	Yes	Default zero
OrderDate	DateTime	Yes	
Customer	String	Yes	Default zero
ShipTo	String	No	Default zero
Note	String	No	
Deleted	Bool	No	
IsClosed	Bool	No	Default false
SubTotal	Decimal	No	Default zero
Discount	Decimal	No	
OtherAdjustment	Decimal	No	Default zero
Tax	Decimal	No	Default zero
Shipping	Decimal	No	
LineItems	List	Yes	Default false
CreateDate	DateTime	No	Default zero
CreatedBy	String	Yes	
LastModifyDate	DateTime	Yes	Default zero

LastSalesOrder

Property Name	Type	Required	Notes
LastModifyOperator	string	Yes	Will use Current user of the passed credentials

StockSalesOrderLiteItem

Property Name	Type	Required	Notes
LineNumber	String	Yes	
StockNumber	String	Yes	Unique in the System
IsTaxable	String	Yes	
UnitPrice	String	Yes	Default zero
Quantity	DateTime	Yes	
ExtendedPrice	String	Yes	Default zero
IssuedQuantity	String	No	Default zero

StockReturnTransaction

Property Name	Type	Required	Notes
SecurityKey	String	Yes	
CustomerCode	String	Yes	
SalesOrderNumber	String	No	
LinteItems	List	Yes	
Notes	DateTime	No	

StockReturnTransactionLineItem

Property Name	Type	Required	Notes
InventoryNumber	String	Yes	
LocationFrom	String	Yes	

StockReturnTransactionLineItem

Property Name	Type	Required	Notes
Price	String	No	Default zero
QuantityToReturn	Int	Yes	
SalesOrderLineNumber	Int	Yes	Required if issueing out against a Sales Order

StockBaseHistotyFields (Read Only Values)

Property Name	Type	Required	Notes
TotalRecords	Int		
TransactionNumber	String		
HistoryDateTime	DateTime		
StockNumber	String		
InventoryID	String		
Description	String		
StockExtendedDescription	String		
Cost	Decimal		
UOM	String		
SiteName	String		
QuantityChange	Decimal		

StockDisposeHistory [StockBaseHistoryFields + Below Fields] (Read Only Values)

Property Name	Type	Required	Notes
InventoryNumber	String		
LocationFrom	String		
LocationFrom	String		

StockIssueHistory [StockBaseHistoryFields + Below Fields] (Read Only Values)

Property Name	Type	Required	Notes
Category	String		
CategoryDescription	String		
Customer	String		
CustomerDescription	String		
ExtendedCost	Decimal		
ExtendedPrice	Decimal		
LocationFrom	String		
Price	Decimal		
SONumber	String		
SalesOrderDate	DateTime		
SalesOrderNotes	String		
StockNotes	String		

StockMoveHistory [StockBaseHistoryFields + Below Fields] (Read Only Values)

Property Name	Type	Required	Notes
HistoryNotes	String		
LocationMoveFrom	String		
LocationMoveTo	String		

StockReceiveHistory [StockBaseHistoryFields + Below Fields] (Read Only Values)

Property Name	Type	Required	Notes
Category	String		
CategoryDescription	String		

StockReturnHistory [StockBaseHistoryFields + Below Fields] (Read Only Values)

Property Name	Type	Required	Notes
Customer	String		
CustomerDescription	String		
ExtendedCost	Decimal		
ExtendedPrice	Decimal		
HisoryNotes	String		
LocationFrom	String		
Price	Decimal		
SONumber	String		
SalesOrderDate	DateTime		
SalesOrderNotes	String		
StockNotes	String		

StockItem

Property Name	Type	Required	Notes
SecurityToken	String	Yes	
StockNumber	String	Yes	Unique in the System
Deleted	Bool	No	Default false
SiteName	String	No	
Type	String	No	Default “Standard”
CostType	String	No	Default “Last/Simple”
Category	String	No	
Inventory_UOM	String	No	Default “EA”
Recieve_UOM	String	No	Default “EA”

StockItem

Property Name	Type	Required	Notes
Issue_UOM	String	Yes	
LeadTimeInterval	String	Yes	Unique in the System
DefaultLocation	Bool	No	Default false
Description	String	No	
ModelDescription	String	No	Default “Standard”
Cost	String	No	Default “Last/Simple”
IS_FIXEDPRICE	String	No	
PriceMarkupPercentage	String	No	Default “EA”
PriceMarkup	String	No	Default “EA”
FIXEDUNITPRICE	String	Yes	
MINIMUMQUANTITY	String	Yes	Unique in the System
MAXIMUMQUANTITY	Bool	No	Default false
LEADTIME	String	No	
ISTAXABLE	String	No	Default false
CreateDate	DateTime	No	Default will use current DateTime
CreatedBy	String	Yes	Will use Current user of the passed credentials
LastModifyDate	Datetime	Yes	Default will use current DateTime
LastModifyOperator	String	Yes	Will use Current user of the passed credentials
ReturnMessage	String	No	Any errors or messages from the server

SqlFilter

Property Name	Type	Required	Notes
SecurityToken	String	Yes	

SqlFilter

Property Name	Type	Required	Notes
IFilter	String	No	If left empty, all records are retrived

View objects in new window: Objects

API Methods

GetSecurityToken(LoginInfo loginInfo)

Description: This method is used to get the security token that is used in every other method

Parameters: LoginInfo object

Return: String representation of the SecurityToken.

```
private string GetSecurityToken()
{
LoginInfo loginInfo = new LoginInfo();
loginInfo.ApplicationKey = "Your Application Key";
loginInfo.CustomerKey = "Customer Key";
loginInfo.UserName = "Username";
loginInfo.Password = "Password";
string SecurityToken = string.Empty;
var serializer = new DataContractJsonSerializer(typeof(LoginInfo));
var ms = new MemoryStream();
serializer.WriteObject(ms, loginInfo);
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
SecurityToken = client.GetSecurityToken(Encoding.UTF8.GetString(ms.ToArray()));
client.Close();
return SecurityToken;
}
```

AddUOM(UOM UomToAdd)

Description: This method is used to Add a Unit of measure into the system.

Parameters: UOM object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string AddUOM()
{
   // Create an instance of a UOM
   UOM uomToadd = new UOM();

   // fill in the UOM Parameter values.
   uomToadd.SecurityToken = “SecurityToken”;
   uomToadd.UnitCode = “UnitCode”;
   uomToadd.Multiplier = multiplier;
       etc…

    string returnedMessage;
    // Convert the Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(UOM));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, uomToadd);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the AddUOM Method.
    returnedMessage = client.AddUOM(Encoding.UTF8.GetString(ms.ToArray()));

    client.Close();
    return returnedMessage;
}
```

GetUOM(RequestObject GPObject)

Description: This method is used to get a specific UOM from the System.

Paramaters: RequestObject object

Return: The requested UOM if it exists in the System.

Sample C# Code:

```
private UOM GetUOM()
{
//Create a RequestObject
       RequestObject Robject = new RequestObject();

       //fill in the Parameter values
       Robject.SecurityToken = "SecurityToken";
       Robject.ObjectIdentifier= "UOMCode to fetch";

       // Convert the RequestObject Object to JSON
       var serializer = new DataContractJsonSerializer(typeof(RequestObject));
       var ms = new MemoryStream();
       serializer.WriteObject(ms, Robject);
```

```
        // create an instance of the web service
        ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

        // Call the GetUOM Method.
        UOM foundUOM = client.GetUOM(Encoding.UTF8.GetString(ms.ToArray()));

        //Close the client connection
        client.Close();
        return foundUOM;
}
```

UpdateUOM(UOM uomToUpdate)

Description: This method is used to update a UOM that already exists in the system.

Paramaters: UOM object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string UpdateUOM()
{
   //Create a RequestObject
   RequestObject Robject = new RequestObject();

   //fill in the Parameter values
   Robject.SecurityToken = "Your SecurityToken";
   Robject.ObjectIdentifier= "UOMCode to edit";

   // Convert the RequestObject Object to JSON
   var serializer = new DataContractJsonSerializer(typeof(RequestObject));
   var ms = new MemoryStream();
   serializer.WriteObject(ms, Robject);

   // create an instance of the web service
   ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
   UOM foundObject = client.GetUOM(Encoding.UTF8.GetString(ms.ToArray()));

   //Close the client connection
   client.Close();

   //now that we have the UOM that we want to modify, let's set up the properties
   // that need modifications.  Note: only the changed properties will be modi-
fied.
   foundObject.UnitName = "New UOM Name";
    etc….
```

```
        // Convert the RequestObject Object to JSON
   var Updateserializer = new DataContractJsonSerializer(typeof(UOM));
   var Updatems = new MemoryStream();
   Updateserializer.WriteObject(Updatems, foundObject);

   // create an instance of the web service
   ServiceReference2.ServiceClient UpdateClient = new ServiceReference2.Service-
Client();

   string Message= UpdateClient.UpdateUOM(Encoding.UTF8.GetString(Updatems.ToAr-
ray()));

   //Close the client connection
   UpdateClient.Close();

   return Message;
}
```

GetUOMs (RequestObject GPObject)

Description: This method is used to get All UOMs from the System.

Paramaters: RequestObject object

Return: All UOMs in the System.

Sample C# Code:

```
private UOM[] GetUOMs()
{
//Create a RequestObject
       RequestObject Robject = new RequestObject();

       //fill in the Parameter values
       Robject.SecurityToken = "SecurityToken";

       // Convert the RequestObject Object to JSON
       var serializer = new DataContractJsonSerializer(typeof(RequestObject));
       var ms = new MemoryStream();
       serializer.WriteObject(ms, Robject);

       // create an instance of the web service
       ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

       // Call the GetUOMs Method.
       UOM[] foundObjects = client.GetUOMs(Encoding.UTF8.GetString(ms.ToAr-
ray()));

       //Close the client connection
       client.Close();
       return foundObjects;
}
```

AddAsset (RequestObject GPObject)

Description: This method is used to get a specific Asset from the System.

Paramaters: RequestObject object

Return: The requested Assets if it exists in the System.

```
private Asset GetAsset()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = "SecurityToken";
Robject.ObjectIdentifier= "AssetNumber to fetch";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the GetAsset Method.
Asset foundAsset = client.GetAsset(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundAsset;
}
```

GetAssets (RequestObject GPObject)

Description: This method is used to get all Assets from the System.

Paramaters: RequestObject object

Return: An Array of Assets.

```
private Asset[] GetAssets()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Security Token Parameter values
Robject.SecurityToken = "Your SecurityToken";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the GetAssets Method.
Asset[] foundAssets = client.GetAssets(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundAssets;
}
```

UpdateAsset (Asset AssetToUpdate)

Description: This method is used to update an Asset that already exists in the system.

Paramaters: Asset object

Return: Any errors in a string representation. An empty return value means operation completed

```
private string UpdateAsset()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = "Your SecurityToken";
Robject.ObjectIdentifier= "AssetNumber to retreive";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the GetAsset Method.
Asset foundAsset = client.GetAsset(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
//now that we have the asset that we want to modify, lets set up the properties
// that need modifications.  Note: only the changed properites will be modified.
foundAsset.BIOS = "New Bios values";
foundAsset.Color = "New Color";
etc….
// Convert the RequestObject Object to JSON
var Updateserializer = new DataContractJsonSerializer(typeof(Asset));
var Updatems = new MemoryStream();
Updateserializer.WriteObject(Updatems, foundAsset);
// create an instance of the web service
ServiceReference2.ServiceClient UpdateClient = new ServiceReference2.ServiceCli-
ent();
string Message= UpdateClient.UpdateAsset(Encoding.UTF8.GetString(Up-
datems.ToArray()));
//Close the client connection
UpdateClient.Close();
return returnedUpdateMessage
}
```

DeleteAsset (RequestObject rqstObject)

Description: This method is used to delete an Asset that already exists in the system.

Paramaters: RequestObject object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string DeleteAsset()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier= "AssetNumber of Asset to delete";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the DeleteAsset Method.
string message = client.DeleteAsset(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return message;
}
```

AddAssetModel (AssetModel AssetModelToAdd)

Description: This method is used to Add an AssetModel into the system.

Paramaters: AssetModel object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string AddAssetModel()
{
string returnedMessage = string.Empty;
// Create an instance of an Asset
AssetModel assetModelToadd = new AssetModel();
// fill in the Paramater values.
assetModelToadd.SecurityToken = _SecurityToken;
assetModelToadd.ModelNumber = "MyTestModel";
assetModelToadd.AssetModelType = "Computer";
etc….
// Convert the Object to JSON
var serializer = new DataContractJsonSerializer(typeof(AssetModel));
var ms = new MemoryStream();
serializer.WriteObject(ms, assetModelToadd);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Add Method.
returnedMessage = client.AddAssetModel(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return returnedMessage;
}
```

GetAssetModel (RequestObject GPObject)

Description: This method is used to get a specific AssetModel from the System.

Paramaters: RequestObject object

Return: The requested AssetModel if it exists in the System.

```
private AssetModel GetAssetModel()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = "SecurityToken";
Robject.ObjectIdentifier= "ModelNumber to fetch";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
AssetModel foundObject = client.GetAssetModel(Encoding.UTF8.GetString(ms.ToAr-
ray()));
//Close the client connection
client.Close();
return foundObject;
}
```

GetAssetModels (RequestObject GPObject)

Description: This method is used to get all AssetModels from the System.

Paramaters: RequestObject object

Return: An Array of AssetModels.

```
private AssetModel[] GetAssetModels()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Security Token Parameter values
Robject.SecurityToken = "Your SecurityToken";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
AssetModel[] foundObjects =
client.GetAssetModels(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundObjects;
}
```

UpdateAssetModel (AssetModel AssetModelToUpdate)

Description: This method is used to update an Asset that already exists in the system.

Paramaters: Asset object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string UpdateAssetModel()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = "Your SecurityToken";
Robject.ObjectIdentifier = "ModelNumber to retreive";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
AssetModel foundObject = client.GetAssetModel(Encoding.UTF8.GetString(ms.ToAr-
ray()));
//Close the client connection
client.Close();
//now that we have the Object that we want to modify, lets set up the properties
// that need modifications.  Note: only the changed properites will be modified.
foundObject.Manufacturer = "MyManufacturer";
etc….
// Convert the RequestObject Object to JSON
var Updateserializer = new DataContractJsonSerializer(typeof(AssetModel));
var Updatems = new MemoryStream();
Updateserializer.WriteObject(Updatems, foundObject);
// create an instance of the web service
ServiceReference2.ServiceClient UpdateClient = new ServiceReference2.ServiceCli-
ent();
string Message= UpdateClient.UpdateAssetModel(Encoding.UTF8.GetString(Up-
datems.ToArray()));
//Close the client connection
UpdateClient.Close();

return Message;
}
```

DeleteAssetModel (RequestObject rqstObject)

Description: This method is used to Delete an AssetModel that already exists in the system.

Paramaters: RequestObject object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string DeleteAssetModel()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "ModelNumber of AssetModel to delete";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Delete Method.
string message = client.DeleteAssetModel(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return message;
}
```

AddCustomer (Customer CustomerToAdd)

Description: This method is used to Add a Customer into the system.

Paramaters: Customer object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string AddCustomer()
{
// Create an instance of a Customer
Customer customerToadd = new Customer();
// fill in the Paramater values.
customerToadd.SecurityToken = _SecurityToken;
customerToadd.Code = "MyCustomer";
customerToadd.Name = "MyCustomer Name";
etc…
string returnedMessage = string.Empty;
// Convert the Object to JSON
var serializer = new DataContractJsonSerializer(typeof(Customer));
var ms = new MemoryStream();
serializer.WriteObject(ms, customerToadd);
// create and instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Add Method.
returnedMessage = client.AddCustomer(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return returnedMessage;
}
```

GetCustomer (RequestObject GPObject)

Description: This method is used to get a specific Customer from the System.

Paramaters: RequestObject object

Return: The requested Customer if it exists in the System.

```
private Customer GetCustomer()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "MyCustomer";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
Customer foundObject = client.GetCustomer(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundObject;
}
```

GetCustomers (RequestObject GPObject)

Description: This method is used to get all Customers from the System.

Paramaters: RequestObject object

Return: An Array of Customers.

```
private Customer[] GetCustomers()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Security Token Parameter values
Robject.SecurityToken = _SecurityToken;
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create and instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the GetCustomers Method.
Customer[] foundObjects =
client.GetCustomers(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundObjects;
}
```

UpdateCustomer (Customer CustomerToUpdate)

Description: This method is used to update a customer that already exists in the system.

Paramaters: Customer object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string UpdateCustomer()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "MyCustomer";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
Customer foundObject = client.GetCustomer(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
//now that we have the Object that we want to modify, lets set up the properties
// that need modifications.  Note: only the changed properites will be modified.
foundObject.Address1 = "MyAddress1";
// Convert the RequestObject Object to JSON
var Updateserializer = new DataContractJsonSerializer(typeof(Customer));
var Updatems = new MemoryStream();
Updateserializer.WriteObject(Updatems, foundObject);
// create an instance of the web service
ServiceReference2.ServiceClient UpdateClient = new ServiceReference2.ServiceCli-
ent();
string Message = UpdateClient.UpdateCustomer(Encoding.UTF8.GetString(Updatems.-
ToArray()));
//Close the client connection
UpdateClient.Close();
return Message;
}
```

AddLocation (Location LocationToAdd)

Description: This method is used to Add a Location into the system.

Paramaters: Location object

Return: Any errors in a string representation. An empty return value means operation completed

```
private string AddLocation()
{
string returnedMessage = string.Empty;
// Create an instance of a Location
Location locationToadd = new Location();
// fill in the Paramater values.
locationToadd.SecurityToken = _SecurityToken;
locationToadd.Code = "MyLocation";
locationToadd.Name = "MyLocation Name";
etc…
// Convert the Object to JSON
var serializer = new DataContractJsonSerializer(typeof(Location));
var ms = new MemoryStream();
serializer.WriteObject(ms, locationToadd);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Add Method.
returnedMessage = client.AddLocation(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return returnedMessage;
}
```

GetLocation (RequestObject GPObject)

Description: This method is used to get a specific Location from the System.

Paramaters: RequestObject object

Return: The requested Location if it exists in the System.

```
private Location GetLocation()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "MyLocation";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
Location foundObject = client.GetLocation(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundObject;
}
```

GetLocations (RequestObject GPObject)

Description: This method is used to get all Locations from the System.

Paramaters: RequestObject object

Return: An Array of Locations.

```
private Location[] GetLocations()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Security Token Parameter values
Robject.SecurityToken = _SecurityToken;
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the GetLocations Method.
Location[] foundObjects =
client.GetLocations(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundObjects;
}
```

UpdateLocation (Location LocationToUpdate)

Description: This method is used to update a location that already exists in the system.

Paramaters: Location object

Return: Any errors in a string representation. An empty return value means operation completed

```
private string UpdateLocation()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "MyLocation";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
Location foundObject = client.GetLocation(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
//now that we have the Object that we want to modify, lets set up the properties
// that need modifications.  Note: only the changed properites will be modified.
foundObject.PaletNumber = "123a";
```

```
// Convert the RequestObject Object to JSON
var Updateserializer = new DataContractJsonSerializer(typeof(Location));
var Updatems = new MemoryStream();
Updateserializer.WriteObject(Updatems, foundObject);
// create an instance of the web service
ServiceReference2.ServiceClient UpdateClient = new ServiceReference2.ServiceCli-
ent();
string Message =
UpdateClient.UpdateLocation(Encoding.UTF8.GetString(Updatems.ToArray()));
//Close the client connection
UpdateClient.Close();
return Message;
}
```

DeleteLocation (RequestObject rqstObject)

Description: This method is used to delete a Location that already exists in the system.

Paramaters: RequestObject object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string DeleteLocation()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "Code of Location to delete";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Delete Method.
string message = client.DeleteLocation(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return message;
}
```

AddPerson (Person PersonToAdd)

Description: This method is used to Add a Person into the system.

Paramaters: Person object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string AddPerson()
{
string returnedMessage = string.Empty;
// Create an instance of a Location
Person PersonToadd = new Person();
// fill in the Paramater values.
PersonToadd.SecurityToken = _SecurityToken;
PersonToadd.Code = "MyPerson";
PersonToadd.FirstName = "Me";
// Convert the Object to JSON
var serializer = new DataContractJsonSerializer(typeof(Person));
var ms = new MemoryStream();
serializer.WriteObject(ms, PersonToadd);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Add Method.
returnedMessage = client.AddPerson(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return returnedMessage;
}
```

DeleteLocation (RequestObject GPObject)

Description: This method is used to get a specific Person from the System.

Paramaters: RequestObject object

Return: The requested Person if it exists in the System.

```
private Person GetPerson()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "MyPerson";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
Person foundObject = client.GetPerson(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundObject;
}
```

GetPersons (RequestObject GPObject)

Description: This method is used to get all Persons from the System.

Paramaters: RequestObject object

Return: An Array of Persons.

```
private Person[] GetPersons()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Security Token Parameter values
Robject.SecurityToken = _SecurityToken;
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the GetAssets Method.
Person[] foundObjects = client.GetPersons(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundObjects;
}
```

UpdatePerson (Person PersonToUpdate)

Description: This method is used to update a Person that already exists in the system.

Paramaters: Person object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string UpdatePerson()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "MyPerson";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
Person foundObject = client.GetPerson(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
```

```
client.Close();
//now that we have the Object that we want to modify, lets set up the properties
// that need modifications.  Note: only the changed properites will be modified.
foundObject.FirstName = "John Doe";
// Convert the RequestObject Object to JSON
var Updateserializer = new DataContractJsonSerializer(typeof(Person));
var Updatems = new MemoryStream();
Updateserializer.WriteObject(Updatems, foundObject);
// create an instance of the web service
ServiceReference2.ServiceClient UpdateClient = new ServiceReference2.ServiceCli-
ent();
string Message =
UpdateClient.UpdatePerson(Encoding.UTF8.GetString(Updatems.ToArray()));
//Close the client connection
UpdateClient.Close();
return Message;
}
```

DeletePerson (RequestObject rqstObject)

Description: This method is used to delete a Person that already exists in the system.

Paramaters: RequestObject object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string DeletePerson()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "MyPerson";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Delete Method.
string message = client.DeletePerson(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return message;
}
```

AddSupplier (Supplier SupplierToAdd)

Description: This method is used to Add a Supplier into the system.

Paramaters: Supplierobject

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string AddSupplier()
{
string returnedMessage = string.Empty;
// Create an instance of a Location
Supplier SupplierToadd = new Supplier();
// fill in the Paramater values.
SupplierToadd.SecurityToken = _SecurityToken;
SupplierToadd.Code = "MySupplier";
SupplierToadd.Contact = "John Doe";
Etc…
// Convert the Object to JSON
var serializer = new DataContractJsonSerializer(typeof(Supplier));
var ms = new MemoryStream();
serializer.WriteObject(ms, SupplierToadd);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Add Method.
returnedMessage = client.AddSupplier(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return returnedMessage;
}
```

GetSupplier (RequestObject GPObject)

Description: This method is used to get a specific Supplier from the System.

Paramaters: RequestObject object

Return: The requested Supplier if it exists in the System.

```
private Supplier GetSupplier()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "MySupplier";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
Supplier foundObject = client.GetSupplier(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundObject;
}
```

GetSuppliers (RequestObject GPObject)

Description: This method is used to get all Suppliers from the System.

Paramaters: RequestObject object

Return: An Array of Suppliers.

```
private Supplier[] GetSuppliers()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Security Token Parameter values
Robject.SecurityToken = _SecurityToken;
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the GetAssets Method.
Supplier[] foundObjects = client.GetSuppliers(Encoding.UTF8.GetString(ms.ToAr-
ray()));
//Close the client connection
client.Close();
return foundObjects;
```

UpdateSupplier (Supplier SupplierToUpdate)

Description: This method is used to update a Supplier that already exists in the system.

Paramaters: Supplier object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string UpdateSupplier()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "MySupplier";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
Supplier foundObject = client.GetSupplier(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
```

```
client.Close();
//now that we have the Object that we want to modify, let's set up the properties
// that need modifications.  Note: only the changed properites will be modified.
foundObject.Contact = "John Smith";
etc…
// Convert the RequestObject Object to JSON
var Updateserializer = new DataContractJsonSerializer(typeof(Supplier));
var Updatems = new MemoryStream();
Updateserializer.WriteObject(Updatems, foundObject);
// create an instance of the web service
ServiceReference2.ServiceClient UpdateClient = new ServiceReference2.ServiceCli-
ent();
string Message =
UpdateClient.UpdateSupplier(Encoding.UTF8.GetString(Updatems.ToArray()));
//Close the client connection
UpdateClient.Close();
return Message;
}
```

DeleteSupplier (RequestObject rqstObject)

Description: This method is used to delete a Supplier that already exists in the system.

Paramaters: RequestObject object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string DeleteSupplier()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Parameter values
Robject.SecurityToken = _SecurityToken;
Robject.ObjectIdentifier = "MySupplier";
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Delete Method.
string message = client.DeleteSupplier(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return message;
}
```

GetPersons (RequestObject GPObject)

Description: This method is used to get all Persons from the System.

Paramaters: RequestObject object

Return: An Array of Persons.

```
private Person[] GetPersons()
{
//Create a RequestObject
RequestObject Robject = new RequestObject();
//fill in the Security Token Parameter values
Robject.SecurityToken = _SecurityToken;
// Convert the RequestObject Object to JSON
var serializer = new DataContractJsonSerializer(typeof(RequestObject));
var ms = new MemoryStream();
serializer.WriteObject(ms, Robject);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the GetAssets Method.
Person[] foundObjects = client.GetPersons(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return foundObjects;
}
```

ReceiveAsset (AssetReceiveTransaction receiveTransaction)

Description: This method is used to receive an Asset into the system.

Paramaters: AssetReceiveTransaction object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string Receive()
{
string returnedMessage = string.Empty;
AssetReceiveTransaction assetReceiveTransaction = new AssetReceiveTransaction();
assetReceiveTransaction.SecurityToken = _SecurityToken;
assetReceiveTransaction.SiteName = "Main";
assetReceiveTransaction.LocationTo = "Receiving";
// now we will add the line items
AssetReceiveTransactionLineItem lineItem1 = new AssetReceiveTransactionLineI-
tem();
lineItem1.AssetModelNumber = "iPad";
lineItem1.AssetNumber = "iPad000123-1";
lineItem1.AssetName = "My iPad";
lineItem1.Cost = 600;
          etc…
```

```
assetReceiveTransaction.LineItems.Add(lineItem1);
// Add another line items
AssetReceiveTransactionLineItem lineItem2 = new AssetReceiveTransactionLineI-
tem();
lineItem2.AssetModelNumber = "iPad";
lineItem2.AssetNumber = "iPad000123-2";
lineItem2.AssetName = "My iPad-2";
lineItem2.Cost = 600;
          etc…
assetReceiveTransaction.LineItems.Add(lineItem2);
// Convert the Object to JSON
var serializer = new DataContractJsonSerializer(typeof(AssetReceiveTransaction));
var ms = new MemoryStream();
serializer.WriteObject(ms, assetReceiveTransaction);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Add Method.
returnedMessage = client.ReceiveAsset(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return returnedMessage;
}
```

MoveAsset (AssetMoveTransaction moveTransaction)

Description: This method is used to move an existing Asset.

Paramaters: AssetMoveTransaction object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string Move()
{
string returnedMessage = string.Empty;
AssetMoveTransaction assetMoveTransaction = new AssetMoveTransaction();
assetMoveTransaction.SecurityToken = _SecurityToken;
// now we will add the lineitems
AssetMoveTransactionLineItem lineItem1 = new AssetMoveTransactionLineItem();
lineItem1.AssetNumber = "iPad000123-3";
lineItem1.LocationTo = "Lab";
    etc…
assetMoveTransaction.LineItems.Add(lineItem1);
// Add another line item
AssetMoveTransactionLineItem lineItem2 = new AssetMoveTransactionLineItem();
lineItem2.AssetNumber = "iPad000123-2";
lineItem2.LocationTo = "Lab";
    etc…
assetMoveTransaction.LineItems.Add(lineItem2);
// Convert the Object to JSON
var serializer = new DataContractJsonSerializer(typeof(AssetMoveTransaction));
var ms = new MemoryStream();
serializer.WriteObject(ms, assetMoveTransaction);
```

```
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Add Method.
returnedMessage = client.MoveAsset(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return returnedMessage;
}
```

DisposeAsset (DisposeMoveTransaction moveTransaction)

Description: This method is used to dispose an existing Asset.

Paramaters: DisposeMoveTransaction object

Return: ny errors in a string representation. An empty return value means operation completed successfully.

```
private string Dispose()
{
string returnedMessage = string.Empty;
AssetDisposeTransaction assetDisposeTransaction = new AssetDisposeTransaction();
assetDisposeTransaction.SecurityToken = _SecurityToken;
// now we will add the lineitems
AssetDisposeTransactionLineItem lineItem1 = new AssetDisposeTransactionLineI-
tem();
lineItem1.AssetNumber = "iPad000123-3";
lineItem1.DisposeTo = "Recylce";
assetDisposeTransaction.LineItems.Add(lineItem1);
// Add another line item
AssetDisposeTransactionLineItem lineItem2 = new AssetDisposeTransactionLineI-
tem();
lineItem2.AssetNumber = "iPad000123-2";
lineItem2.DisposeTo = "Recylce";
assetDisposeTransaction.LineItems.Add(lineItem2);
// Convert the Object to JSON
var serializer = new DataContractJsonSerializer(typeof(AssetDisposeTransaction));
var ms = new MemoryStream();
serializer.WriteObject(ms, assetDisposeTransaction);
// create an instance of the web service
ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Add Method.
returnedMessage = client.DisposeAsset(Encoding.UTF8.GetString(ms.ToArray()));
//Close the client connection
client.Close();
return returnedMessage;
}
```

GetAllInventory (RequestObject GPObject)

Description: This method is used to get all the Stock Inventory from the System.

Paramaters: RequestObject object

Return: An Array of StockInventory objects.

Sample C# Code:

```
private StockInventory[] GetAllInventory()
{
    //Create a RequestObject
    RequestObject Robject = new RequestObject();

    //fill in the Security Token Parameter values
    Robject.SecurityToken = _SecurityToken;

    // Convert the RequestObject Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(RequestObject));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, Robject);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the GetAllInventory Method.
    StockInventory[] foundObjects =
       client.GetAllInventory(Encoding.UTF8.GetString(ms.ToArray()));


    //Close the client connection
    client.Close();
    return foundObjects;
}
```

GetInventoryByStockNumber (RequestObject GPObject)

Description: This method is used to get all the Stock Inventory from the System per stockNumber.

Paramaters: RequestObject object

Return: An Array of StockInventory objects.

Sample C# Code:

```
private StockInventory[] GetInventoryByStockNumber()
{
    //Create a RequestObject
    RequestObject Robject = new RequestObject();

    //fill in the Security Token Parameter values
    Robject.SecurityToken = _SecurityToken;
    Robject.ObjectIdentifier = "StockNumberToFilterBy";
```

GetAllInventory (RequestObject GPObject)

Description: This method is used to get all the Stock Inventory from the System.

Paramaters: RequestObject object

Return: An Array of StockInventory objects.

Sample C# Code:

```
private StockInventory[] GetAllInventory()
{
    //Create a RequestObject
    RequestObject Robject = new RequestObject();

    //fill in the Security Token Parameter values
    Robject.SecurityToken = _SecurityToken;

    // Convert the RequestObject Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(RequestObject));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, Robject);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the GetAllInventory Method.
    StockInventory[] foundObjects =
       client.GetAllInventory(Encoding.UTF8.GetString(ms.ToArray()));


    //Close the client connection
    client.Close();
    return foundObjects;
}
```

GetInventoryByStockNumber (RequestObject GPObject)

Description: This method is used to get all the Stock Inventory from the System per stockNumber.

Paramaters: RequestObject object

Return: An Array of StockInventory objects.

Sample C# Code:

```
private StockInventory[] GetInventoryByStockNumber()
{
    //Create a RequestObject
    RequestObject Robject = new RequestObject();

    //fill in the Security Token Parameter values
    Robject.SecurityToken = _SecurityToken;
    Robject.ObjectIdentifier = "StockNumberToFilterBy";
```

```
    // Convert the RequestObject Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(RequestObject));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, Robject);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the GetInventoryByStockNumber Method.
    StockInventory[] foundObjects =
       client.GetInventoryByStockNumber(Encoding.UTF8.GetString(ms.ToArray()));


    //Close the client connection
    client.Close();
    return foundObjects;
}
```

GetInventoryByLocation (RequestObject GPObject)

Description: This method is used to get all the Stock Inventory from the System per Location.

Paramaters: RequestObject object

Return: An Array of StockInventory objects.

Sample C# Code:

```
private StockInventory[] GetInventoryByLocation()
{
    //Create a RequestObject
    RequestObject Robject = new RequestObject();

    //fill in the Security Token Parameter values
    Robject.SecurityToken = _SecurityToken;
    Robject.ObjectIdentifier = "LocationToFilterBy";


    // Convert the RequestObject Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(RequestObject));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, Robject);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the GetInventoryByLocation Method.
    StockInventory[] foundObjects =
       client.GetInventoryByLocation(Encoding.UTF8.GetString(ms.ToArray()));


    //Close the client connection
    client.Close();
    return foundObjects;
}
```

GetInventoryBySiteName (RequestObject GPObject)

Description: This method is used to get all the Stock Inventory from the System per Site.

Paramaters: RequestObject object

Return: Aan Array of StockInventory objects.

Sample C# Code:

```
private StockInventory[] GetInventoryBySiteName()
{
    //Create a RequestObject
    RequestObject Robject = new RequestObject();

    //fill in the Security Token Parameter values
    Robject.SecurityToken = _SecurityToken;
    Robject.ObjectIdentifier = "SiteToFilterBy";


    // Convert the RequestObject Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(RequestObject));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, Robject);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the GetInventoryByLocation Method.
    StockInventory[] foundObjects =
       client.GetInventoryBySiteName(Encoding.UTF8.GetString(ms.ToArray()));

    //Close the client connection
    client.Close();
    return foundObjects;
}
```

CheckoutAsset (AssetCheckoutTransaction checkoutTransac-

Description: This method is used to Checkout an existing Asset, please note that all assets associated with this asset will also be checked out.

Paramaters: AssetCheckoutTransaction object

Return: any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string CheckoutAsset()
{
     string returnedMessage = string.Empty;
     AssetCheckoutTransaction assetCheckoutTransaction = new AssetCheckoutTrans-
action();
     assetCheckoutTransaction.SecurityToken = _SecurityToken;
```

```
// now we will add the line items

     AssetCheckoutTransactionLineItem lineItem1 = new AssetCheckoutTransaction-
LineItem();
     lineItem1.AssetNumber = "AssetToCheckout";
     lineItem1.CheckoutToLocation = "LocationToCheckoutTo";
     lineItem1.DueDays = 10;
     lineItem1.DueDate = DateTime.Now.AddDays(10);
    etc…
     assetCheckoutTransaction.LineItems.Add(lineItem1);

     AssetCheckoutTransactionLineItem lineItem2 = new AssetCheckoutTransaction-
LineItem();
     lineItem2.AssetNumber = "AssetToCheckout_2";
     lineItem2.CheckoutToLocation = "LocationToCheckoutTo";
     lineItem2.DueDays = 10;
     lineItem2.DueDate = DateTime.Now.AddDays(10);
    etc…
     assetCheckoutTransaction.LineItems.Add(lineItem2);

       etc…


     // Convert the Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(AssetCheckoutTransac-
tion));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, assetCheckoutTransaction);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Checkout Method.
     returnedMessage = client.CheckoutAsset(Encoding.UTF8.GetString(ms.ToAr-
ray()));
     //Close the client connection
     client.Close();

      return returnedMessage;
 }
```

CheckinAsset (AssetCheckinTransaction checkinTransaction)

Description: This method is used to Checkin an existing Asset, please note that all assets associated with this asset will also be checked in.

Paramaters: AssetCheckinTransaction object

Return: ny errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string CheckinAsset()
{
     string returnedMessage = string.Empty;
     AssetCheckinTransaction assetCheckoutTransaction = new AssetCheckinTransac-
tion();
     assetCheckinTransaction.SecurityToken = _SecurityToken;

     // now we will add the line items

     AssetCheckinTransactionLineItem lineItem1 = new AssetCheckinTransaction-
LineItem();
     lineItem1.AssetNumber = "AssetToCheckin";
     lineItem1.CheckinToLocation = "LocationToCheckInto";
    etc…
     assetCheckinTransaction.LineItems.Add(lineItem1);

     AssetCheckinTransactionLineItem lineItem2 = new AssetCheckinTransaction-
LineItem();
     lineItem2.AssetNumber = "AssetToCheckin2";
     lineItem2.CheckoutToLocation = "LocationToCheckInto";
       etc…
     assetCheckinTransaction.LineItems.Add(lineItem2);

       etc…


     // Convert the Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(AssetCheckinTransac-
tion));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, assetCheckinTransaction);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Checkin Method.
     returnedMessage = client.CheckinAsset(Encoding.UTF8.GetString(ms.ToAr-
ray()));
     //Close the client connection
     client.Close();

      return returnedMessage;
 }
```

AddAssetPurchaseOrder (AssetPurchaseOrder AssetPO-

Description: This method is used to Add an Asset Purchase Order into the system.

Paramaters: AssetPurchaseOrder object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string AddAssetPurchaseOrder()
{
   // Create an instance of an Asset
   AssetPurchaseOrder assetPOToadd = new AssetPurchaseOrder();

   // fill in the Purchase Order Parameter values.
   assetPOToadd.SecurityToken = "SecurityToken";
   assetPOToadd.PONumber = "AssetPONumber";
   assetPOToadd.OrderDate = "CurrentDate";
       etc…
   // fill in the Purchase Order Line Items Parameter values.
   AssetPurchaseOrderLineItem LI1 = new AssetPurchaseOrderLineItem();
   LI1.AssetModel = "AssetModel";
   LI1.LineNumber = 1;
    etc…
   assetPOToadd.LineItems.Add(LI1);

   AssetPurchaseOrderLineItem LI2 = new AssetPurchaseOrderLineItem();
   LI2.AssetModel = "AssetModel2";
   LI2.LineNumber = 2;
    etc…
   assetPOToadd.LineItems.Add(LI2);

       etc….


    string returnedMessage;
    // Convert the Asset Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(AssetPurchaseOrder));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, assetPOToadd);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the AddAssetPurchaseOrder Method.
    returnedMessage = client.AddAssetPurchaseOrder(Encoding.UTF8.Get-
String(ms.ToArray()));

    client.Close();
    return returnedMessage;
}
```

GetAssetPurchaseOrder (RequestObject GPObject)

Description: This method is used to get a specific Asset Purchase Order from the System.

Paramaters: RequestObject object

Return: The requested Asset Purchase Order if it exists in the System.

Sample C# Code:

```
private AssetPurchaseOrder GetAssetPurchaseOrder()
{
//Create a RequestObject
       RequestObject Robject = new RequestObject();

       //fill in the Parameter values
       Robject.SecurityToken = "SecurityToken";
       Robject.ObjectIdentifier= "AssetPONumber to fetch";

       // Convert the RequestObject Object to JSON
       var serializer = new DataContractJsonSerializer(typeof(RequestObject));
       var ms = new MemoryStream();
       serializer.WriteObject(ms, Robject);

       // create an instance of the web service
       ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

       // Call the GetAssetPurchaseOrder Method.
       AssetPurchaseOrder foundPO =
             client.GetAssetPurchaseOrder(Encoding.UTF8.GetString(ms.ToArray()));

       //Close the client connection
       client.Close();
       return foundPO;
}
```

GetAssetPurchaseOrders (SqlFilter GPObject)

Description: This method is used to get all Asset Purchase Orders from the System.

Paramaters: SqlFilter object

Return: he requested Asset Purchase Orders that fit the filter, if filter is passed as an empty string, all Asset purchase orders will be returned.

Sample C# Code:

```
 private AssetPurchaseOrder[] GetAssetPurchaseOrders()
{
//Create a RequestObject
       SqlFilter Robject = new SqlFilter();

       //fill in the Parameter values
       Robject.SecurityToken = "SecurityToken";
       Robject.Filter = "Deleted = 0 and OrderStatus = 'closed' ";

       // Convert the RequestObject Object to JSON
       var serializer = new DataContractJsonSerializer(typeof(SqlFilter));
       var ms = new MemoryStream();
       serializer.WriteObject(ms, Robject);
```

```
// create an instance of the web service
       ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

       // Call the GetAssetPurchaseOrder Method.
       AssetPurchaseOrder[] foundObject = client.GetAssetPurchaseOrders(Encod-
ing.UTF8.GetString(ms.ToArray()));
       //Close the client connection
       client.Close();
       return foundObject;
}
```

UpdateAssetPurchaseOrder (AssetPurchaseOrder AssetPOToUpdate)

Description: This method is used to update an Asset Purchase Order header information that already exists in the system.

Paramaters: AssetPurchaseOrder object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string UpdateAssetPurchaseOrder()
{

   //Create a RequestObject
   RequestObject Robject = new RequestObject();

   //fill in the Parameter values
   Robject.SecurityToken = "Your SecurityToken";
   Robject.ObjectIdentifier= "AssetPONumber to edit";

   // Convert the RequestObject Object to JSON
   var serializer = new DataContractJsonSerializer(typeof(RequestObject));
   var ms = new MemoryStream();
   serializer.WriteObject(ms, Robject);

   // create an instance of the web service
   ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
   AssetPurchaseOrder foundObject =
       client.GetAssetPurchaseOrder(Encoding.UTF8.GetString(ms.ToArray()));

   //Close the client connection
   client.Close();

   //now that we have the asset PO that we want to modify, lets set up the prop-
erties
   // that need modifications.  Note: only the changed properties will be modi-
fied.
   foundObject.Supplier = "New supplier";
    etc….
```

```
// Convert the RequestObject Object to JSON
   var Updateserializer = new DataContractJsonSerializer(typeof(AssetPurchaseOr-
der));
   var Updatems = new MemoryStream();
   Updateserializer.WriteObject(Updatems, foundObject);

   // create an instance of the web service
   ServiceReference2.ServiceClient UpdateClient = new ServiceReference2.Service-
Client();

   string Message= UpdateClient.UpdateAssetPurchaseOrder(Encoding.UTF8.Get-
String(Updatems.ToArray()));

   //Close the client connection
   UpdateClient.Close();

   return returnedUpdateMessage
}
```

AddBarCloudUser (BarCloudUser UserToAdd)

Description: This method is used to Add a BarCloud User into the system.

Paramaters: BarCloudUser object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string AddBarCloudUser()
{
    string returnedMessage = string.Empty;

    // Create an instance of a BarCloudUser
    BarCloudUser oUser = new BarCloudUser();

    // fill in the Paramater values.
    oUser.SecurityToken = "Your SecurityToken";
    oUser.Username = "Username";
    oUser.Password = "Password";
    oUser.Person = "Person associated with this user";
    oUser.Offset = TimeZone.CurrentTimeZone.ToString();
    oUser.SecurityLevel = "SecurityLevel";
    oUser.Email = "Email";

    // Convert the Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(BarCloudUser));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, oUser);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();
```

```
    // Call the Add Method.
    returnedMessage = client.AddBarCloudUser(Encoding.UTF8.GetString(ms.ToAr-
ray()));
    //Close the client connection
    client.Close();
    return returnedMessage;
}
```

GetBarCloudUsers (RequestObject GPObject)

Description: This method is used to get all BarCloudUsers from the System.

Paramaters: RequestObject object

Return: An Array of BarCloudUsers.

Sample C# Code:

```
private BarCloudUsers[] GetBarCloudUsers()
{

//Create a RequestObject
       RequestObject Robject = new RequestObject();

       //fill in the Security Token Parameter values
       Robject.SecurityToken = "Your SecurityToken";

       // Convert the RequestObject Object to JSON
       var serializer = new DataContractJsonSerializer(typeof(RequestObject));
       var ms = new MemoryStream();
       serializer.WriteObject(ms, Robject);

       // create an instance of the web service
       ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

       // Call the Get Method.
       BarCloudUser[] foundUsers =
            client.GetBarCloudUsers(Encoding.UTF8.GetString(ms.ToArray()));

//Close the client connection
       client.Close();
       return foundUsers;
}
```

GetBarCloudUserByUsername (RequestObject GPObject)

Description: This method is used to get a specific User from the System.

Paramaters: RequestObject object

Return: The requested User if it exists in the System.

Sample C# Code:

```
private BarCloudUser GetBarCloudUserByUsername()
{
//Create a RequestObject
       RequestObject Robject = new RequestObject();

       //fill in the Parameter values
       Robject.SecurityToken = "SecurityToken";
       Robject.ObjectIdentifier= "Username to fetch";

       // Convert the RequestObject Object to JSON
       var serializer = new DataContractJsonSerializer(typeof(RequestObject));
       var ms = new MemoryStream();
       serializer.WriteObject(ms, Robject);

       // create an instance of the web service
       ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

       // Call the GetBarCloudUserByUsername Method.
       BarCloudUser foundUser =
       client.GetBarCloudUserByUsername(Encoding.UTF8.GetString(ms.ToArray()));

       //Close the client connection
       client.Close();
       return foundUser;
}
```

ResetBarCloudUserPassword (BarCloudUser oUser)

Description: This method is used to reset the Password of a specific User.

Paramaters: BarCloudUser object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string ResetPassword()
{
    //Create a RequestObject
    RequestObject Robject = new RequestObject();

    //fill in the Parameter values
    Robject.SecurityToken = "SecurityToken";
    Robject.ObjectIdentifier= "Username to fetch";

    // Convert the RequestObject Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(RequestObject));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, Robject);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();
```

```
 // Call the Get Method.
    BarCloudUser foundObject = client.GetBarCloudUserByUsername(Encod-
ing.UTF8.GetString(ms.ToArray()));

    //Close the client connection
    client.Close();


    // Convert the RequestObject Object to JSON
    var Updateserializer = new DataContractJsonSerializer(typeof(BarCloudUser));
    var Updatems = new MemoryStream();
    Updateserializer.WriteObject(Updatems, foundObject);

    // create an instance of the web service
    ServiceReference2.ServiceClient oClient = new ServiceReference2.ServiceCli-
ent();

    string Message = oClient.ResetPassword(Encoding.UTF8.GetString(Updatems.ToAr-
ray()));

    //Close the client connection
    oClient.Close();

    return Message;
}
```

GetBarCloudLoggedInUsers (RequestObject GPObject)

Description: This method is used to get all BarCloudUsers that are currently logged in.

Paramaters: RequestObject object

Return: An Array of BarCloudUsers.

Sample C# Code:

```
private BarCloudUsers[] GetLoggedInUsers()
{
    //Create a RequestObject
    RequestObject Robject = new RequestObject();

      //fill in the Parameter values
    Robject.SecurityToken = "SecurityToken";

       // Convert the RequestObject Object to JSON
       var serializer = new DataContractJsonSerializer(typeof(RequestObject));
       var ms = new MemoryStream();
       serializer.WriteObject(ms, Robject);

       // create an instance of the web service
       ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();
```

```
        // Call the Get Method.
        BarCloudUser[] LoggedInUsers =
        client.GetBarCloudLoggedInUsers(Encoding.UTF8.GetString(ms.ToArray()));

        //Close the client connection
        client.Close();
        return LoggedInUsers;
}
```

LogBarCloudUserOut (BarCloudUser oUser)

Description: This method is used to Log out a BarCloudUsers that is currently logged in.

Paramaters: BarCloudUser object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

```
private string LogUserOut()
{
    //Create a RequestObject
    RequestObject Robject = new RequestObject();

    //fill in the Parameter values
    Robject.SecurityToken = "SecurityToken";
    Robject.ObjectIdentifier = "Username of User to Log out";

    // Convert the RequestObject Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(RequestObject));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, Robject);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the Get Method.
    BarCloudUser foundObject =
        client.GetBarCloudUserByUsername(Encoding.UTF8.GetString(ms.ToArray()));

    //Close the client connection
    client.Close();

    // Convert the RequestObject Object to JSON
    var Updateserializer = new DataContractJsonSerializer(typeof(BarCloudUser));
    var Updatems = new MemoryStream();
    Updateserializer.WriteObject(Updatems, foundObject);

    // create an instance of the web service
    ServiceReference2.ServiceClient oClient = new ServiceReference2.ServiceCli-
ent();

    string Message = oClient.LogUserOut(Encoding.UTF8.GetString(Updatems.ToAr-
ray()));
```

```
    //Close the client connection
    oClient.Close();

    return Message;
}
```

AddAssetMaintenanceScheduleOneTime (AssetMaintenanceSheduleOneTime oParam)

Description: This method is used to Add a onetime Asset Maintenance into the system.

Paramaters: AssetMaintenanceSheduleOneTime object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string ScheduleOneTimeAssetMaintenance()
{
   string returnedMessage = string.Empty;

    // Create an instance of the object
    AssetMaintenanceScheduleOneTime oParam = new AssetMaintenanceScheduleOne-
Time();

    // fill in the Paramater values.
    oParam.SecurityToken = "SecurityToken";
    oParam.MaintenanceCode = "AssetMaintenanceCode";
    oParam.AssetNumber = "AssetNumber";
    oParam.ScheduledFor = "Date Scheduled For";
       etc…

    // Convert the Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(AssetMaintenanceSched-
uleOneTime));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, oParam);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the Add Method.
    returnedMessage =
     client.AddAssetMaintenanceScheduleOneTime(Encoding.UTF8.Get-
String(ms.ToArray()));
    //Close the client connection
    client.Close();
    return returnedMessage;
}
```

AddAssetMaintenanceScheduleRecurring (AssetMaintenanceSheduleRecurring oParam)

Description: This method is used to Add a recurring Asset Maintenance into the system.

Paramaters: AssetMaintenanceSheduleRecurring object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string ScheduleRecurringAssetMaintenance()
{
   string returnedMessage = string.Empty;

    // Create an instance of the object
    AssetMaintenanceScheduleRecurring oParam = new AssetMaintenanceScheduleRecur-
ring();

    // fill in the Paramater values.
    oParam.SecurityToken = "SecurityToken";
    oParam.MaintenanceCode = "AssetMaintenanceCode";
    oParam.AssetNumber = "AssetNumber";
    oParam.Frequency = 5;
       etc…
    // Convert the Object to JSON
    var serializer =
       new DataContractJsonSerializer(typeof(AssetMaintenanceScheduleRecurring));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, oParam);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the Add Method.
    returnedMessage =
     client.AddAssetMaintenanceScheduleRecurring(Encod-
ing.UTF8.GetString(ms.ToArray()));
    //Close the client connection
    client.Close();
    return returnedMessage;
}
```

PerformAssetMaintenance (AssetMaintenancePerform oParam)

Description: This method is used to Peform an Asset Maintenance.

Paramaters: AssetMaintenancePerform object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string PerformAssetMaintenance()
{
       string returnedMessage = string.Empty;

    // Create an instance of a Purchase Order
    AssetMaintenancePerform oParam = new AssetMaintenancePerform();

    // fill in the Paramater values.
    oParam.SecurityToken = "SecurityToken";
    oParam.PerformMaintenanceCode = "PerformMaintenanceCode";
    oParam.PerformDate = DateTime.Now;
    oParam.Person = "PersonPerformedBy";
       etc…



    // Convert the Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(AssetMaintenancePer-
form));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, oParam);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the Perform Method.
    returnedMessage = client.PerformAssetMaintenance(Encoding.UTF8.Get-
String(ms.ToArray()));
    //Close the client connection
    client.Close();
    return returnedMessage;
}
```

IssueStock (StockIssueTransaction issueTransaction)

Description: This method is used to issue out Stocks.

Paramaters: StockIssueTransaction object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string IssueStock()
{
     string returnedMessage = string.Empty;
     StockIssueTransaction issueTransaction = new StockIssueTransaction ();
     issueTransaction.SecurityToken = "Your Security token";
     issueTransaction.CustomerCode = "Customer Code";
     issueTransaction.SalesOrderNumber = "Sales Order # if any";
     issueTransaction.Notes = "Transaction Notes";
```

```
 // now we will add the line items

     StockIssueTransactionLineItem lineItem1 = new StockIssueTransactionLineI-
tem();
     lineItem1.InventoryNumber = "InventoryNumber to issue out";
     lineItem1.LocationFrom = "From Location";
     lineItem1.Price = 6;
     lineItem1.QuantityToIssue = 1;
     lineItem1.SalesOrderLineNumber = 2;
     issueTransaction.LineItems.Add(lineItem1);

     StockIssueTransactionLineItem lineItem2 = new StockIssueTransactionLineI-
tem();
     lineItem2.InventoryNumber = "InventoryNumber to issue out";
     lineItem2.LocationFrom = "From Location";
     lineItem2.Price = 6;
     lineItem2.QuantityToIssue = 1;
     lineItem2.SalesOrderLineNumber = 2;
     issueTransaction.LineItems.Add(lineItem1);

       etc…

     // Convert the Object to JSON

     var serializer = new DataContractJsonSerializer(typeof(StockIssueTransac-
tion));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, issueTransaction);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Checkout Method.
     returnedMessage = client.IssueStock(Encoding.UTF8.GetString(ms.ToArray()));
     //Close the client connection
     client.Close();

      return returnedMessage;
 }
```

GetAssetAllHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset All History.

Paramaters: HistoryRequestObject object

Return: Array of AssetAllHistory Objects.

Sample C# Code:

```
public AssetAllHistory[] GetAssetAllHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench' AND
InvHistoryType =
                              'Move'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
     AssetAllHistory[] foundObject =
           client.GetAssetAllHistory(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetReceiveHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Receive History.

Paramaters: HistoryRequestObject object

Return: Array of AssetAllHistory Objects.

Sample C# Code:

```
public AssetReceiveHistory[] GetAssetReceiveHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";
```

```
// Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
     AssetReceiveHistory [] foundObject =
           client.GetAssetReceiveHistory(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetMoveHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Move History.

Paramaters: HistoryRequestObject object

Return: Array of AssetMoveHistory Objects.
Sample C# Code:

```
public AssetMoveHistory[] GetAssetMoveHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
     AssetMoveHistory [] foundObject =
           client.GetAssetMoveHistory(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetDisposeHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Dispose History.

Paramaters: HistoryRequestObject object

Return: Array of AssetDisposeHistory Objects.

Sample C# Code:

```
public AssetDisposeHistory[] GetAssetDisposeHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
     AssetDisposeHistory [] foundObject =
           client.GetAssetDisposeHistory(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetCheckoutHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Checkout History.

Paramaters: HistoryRequestObject object

Return: Array of AssetCheckoutHistory Objects.

.

Sample C# Code:

```
public AssetCheckoutHistory[] GetAssetCheckoutHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";
```

```
// Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
     AssetCheckoutHistory [] foundObject =
           client.GetAssetCheckoutHistory(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetCheckinHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Checkin History.

Paramaters: HistoryRequestObject object

Return: Array of AssetCheckinHistory Objects.

Sample C# Code:

```
public AssetCheckinHistory[] GetAssetCheckinHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
     AssetCheckinHistory [] foundObject =
           client.GetAssetCheckinHistory(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

AddStockPurchaseOrde (StockPurchaseOrder StockPOToAdd)

Description: This method is used to Add a Stock Purchase Order into the system.

Paramaters: StockPurchaseOrder object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string AddStockPurchaseOrder()
{
   // Create an instance of a Stock Purchase Order
   StockPurchaseOrder stockPOToadd = new StockPurchaseOrder();

   // fill in the Purchase Order Parameter values.
   stockPOToadd.SecurityToken = "SecurityToken";
   stockPOToadd.PONumber = "StockPONumber";
   stockPOToadd.OrderDate = "CurrentDate";
       etc…
   // fill in the Purchase Order Line Items Parameter values.
   StockPurchaseOrderLineItem LI1 = new StockPurchaseOrderLineItem();
   LI1.StockNumber = "StockNumber1";
   LI1.LineNumber = 1;
    etc…
   stockPOToadd.LineItems.Add(LI1);

   StockPurchaseOrderLineItem LI2 = new StockPurchaseOrderLineItem();
   LI2.StockNumber = "StockNumber2";
   LI2.LineNumber = 2;
    etc…
   stockPOToadd.LineItems.Add(LI2);

       etc….


    string returnedMessage;
    // Convert the Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(StockPurchaseOrder));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, stockPOToadd);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the AddPurchaseOrder Method.
    returnedMessage = client.AddStockPurchaseOrder(Encoding.UTF8.Get-
String(ms.ToArray()));

    client.Close();
    return returnedMessage;
}
```

GetStockPurchaseOrder (RequestObject GPObject)

Description: This method is used to get a specific Stock Purchase Order from the System.

Paramaters: RequestObject object

Return: the requested Stock Purchase Order if it exists in the System.

Sample C# Code:

```
private StockPurchaseOrder GetStockPurchaseOrder()
{
//Create a RequestObject
       RequestObject Robject = new RequestObject();

       //fill in the Parameter values
       Robject.SecurityToken = "SecurityToken";
       Robject.ObjectIdentifier= "StockPONumber to fetch";

       // Convert the RequestObject Object to JSON
       var serializer = new DataContractJsonSerializer(typeof(RequestObject));
       var ms = new MemoryStream();
       serializer.WriteObject(ms, Robject);

       // create an instance of the web service
       ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

       // Call the GetStockPurchaseOrder Method.
       StockPurchaseOrder foundPO =
             client.GetStockPurchaseOrder(Encoding.UTF8.GetString(ms.ToArray()));

       //Close the client connection
       client.Close();
       return foundPO;
}
```

GetStockPurchaseOrders (SqlFilter GPObject)

Description: This method is used to get all Stock Purchase Orders from the System.

Paramaters: SqlFilter object

Return: the requested Stock Purchase Orders that fit the filter, if filter is passed as an empty string, all Stock purchase orders will be returned.

Sample C# Code:

```
private StockPurchaseOrder[] GetStockPurchaseOrders()
{
//Create a RequestObject
       SqlFilter Robject = new SqlFilter();

       //fill in the Parameter values
       Robject.SecurityToken = "SecurityToken";
       Robject.Filter = "Deleted = 0 and OrderStatus = 'closed' ";
```

```
// Convert the RequestObject Object to JSON
       var serializer = new DataContractJsonSerializer(typeof(SqlFilter));
       var ms = new MemoryStream();
       serializer.WriteObject(ms, Robject);

       // create an instance of the web service
       ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

       // Call the GetAssetPurchaseOrder Method.
       AssetPurchaseOrder[] foundObject = client.GetStockPurchaseOrders(Encod-
ing.UTF8.GetString(ms.ToArray()));
       //Close the client connection
       client.Close();
       return foundObject;
}
```

UpdateStockPurchaseOrder (StockPurchaseOrder StockPOToUpdate)

Description: This method is used to update a stock Purchase that already exists in the system.

Paramaters: StockPurchaseOrder object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string UpdateStockPurchaseOrder()
{

   //Create a RequestObject
   RequestObject Robject = new RequestObject();

   //fill in the Parameter values
   Robject.SecurityToken = "Your SecurityToken";
   Robject.ObjectIdentifier= "StockPONumber to edit";

   // Convert the RequestObject Object to JSON
   var serializer = new DataContractJsonSerializer(typeof(RequestObject));
   var ms = new MemoryStream();
   serializer.WriteObject(ms, Robject);

   // create an instance of the web service
   ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
   StockPurchaseOrder foundObject =
       client.GetStockPurchaseOrder(Encoding.UTF8.GetString(ms.ToArray()));

   //Close the client connection
   client.Close();
```

```
//now that we have the PO that we want to modify, lets set up the properties
   // that need modifications.  Note: only the changed properties will be modi-
fied.
   foundObject.Supplier = "New supplier";
    etc….

   // now let's increase line item one quantity by one
   foreach (StockPurchaseOrderLineItem li in foundObject.LineItems)
   {
       if (li.LineNumber == 1)
       {
          li.UnitsOrdered = li.UnitsOrdered+1;
       }
   }

   // now let's Add a new line Item
   StockPurchaseOrderLineItem NewLine = new StockPurchaseOrderLineItem();
   NewLine.StockNumber = "StockNumber3";
   NewLine.LineNumber = 3;
    etc…
   foundObject.LineItems.Add(NewLine);



   // Convert the RequestObject Object to JSON
   var Updateserializer = new DataContractJsonSerializer(typeof(StockPurchaseOr-
der));
   var Updatems = new MemoryStream();
   Updateserializer.WriteObject(Updatems, foundObject);

   // create an instance of the web service
   ServiceReference2.ServiceClient UpdateClient = new ServiceReference2.Service-
Client();

   string Message= UpdateClient.UpdateStockPurchaseOrder(Encoding.UTF8.Get-
String(Updatems.ToArray()));

   //Close the client connection
   UpdateClient.Close();

   return Message;
}
```

AddStockSalesOrder (StockSalesOrder StockSOToAdd)

Description: This method is used to Add a Stock Sales Order into the system.

Paramaters: StockSalesOrder object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string AddStockSalesOrder()
{
   // Create an instance of a Stock Sales Order
   StockSalesOrder stockSOToadd = new StockSalesOrder();

   // fill in the Purchase Order Parameter values.
   stockSOToadd.SecurityToken = "SecurityToken";
   stockSOToadd.SONumber = "StockSONumber";
   stockSOToadd.OrderDate = "CurrentDate";
       etc…
   // fill in the Sales Order Line Items Parameter values.
   StockSalesOrderLineItem LI1 = new StockSalesOrderLineItem();
   LI1.StockNumber = "StockNumber1";
   LI1.LineNumber = 1;
    etc…
   stockSOToadd.LineItems.Add(LI1);

   StockSalesOrderLineItem LI2 = new StockSalesOrderLineItem();
   LI2.StockNumber = "StockNumber2";
   LI2.LineNumber = 2;
    etc…
   stockSOToadd.LineItems.Add(LI2);

       etc….


    string returnedMessage;
    // Convert the Object to JSON
    var serializer = new DataContractJsonSerializer(typeof(StockSalesOrder));
    var ms = new MemoryStream();
    serializer.WriteObject(ms, stockPOToadd);

    // create an instance of the web service
    ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

    // Call the AddSalesOrder Method.
    returnedMessage = client.AddStockSalesOrder(Encoding.UTF8.GetString(ms.ToAr-
ray()));

    client.Close();
    return returnedMessage;
}
```

GetStockSalesOrder (RequestObject GPObject)

Description: This method is used to get a specific Stock Sales Order from the System.

Paramaters: RequestObject object

Return: The requested Stock Sales Order if it exists in the System.

Sample C# Code:

```
private StockSalesOrder GetStockSalesOrder()
{
//Create a RequestObject
       RequestObject Robject = new RequestObject();

       //fill in the Parameter values
       Robject.SecurityToken = "SecurityToken";
       Robject.ObjectIdentifier= "StockSONumber to fetch";

       // Convert the RequestObject Object to JSON
       var serializer = new DataContractJsonSerializer(typeof(RequestObject));
       var ms = new MemoryStream();
       serializer.WriteObject(ms, Robject);

       // create an instance of the web service
       ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

       // Call the GetStockSalesOrder Method.
       StockSalesOrder foundPO =
             client.GetStockSalesOrder(Encoding.UTF8.GetString(ms.ToArray()));

       //Close the client connection
       client.Close();
       return foundPO;
}
```

UpdateStockSalesOrder (StockSalesOrder StockSOToUpdate)

Description: This method is used to update a stock Sales that already exists in the system.

Paramaters: StockSalesOrder object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
private string UpdateStockSalesOrder()
{

   //Create a RequestObject
   RequestObject Robject = new RequestObject();

   //fill in the Parameter values
   Robject.SecurityToken = "Your SecurityToken";
   Robject.ObjectIdentifier= "StockSONumber to edit";

   // Convert the RequestObject Object to JSON
   var serializer = new DataContractJsonSerializer(typeof(RequestObject));
   var ms = new MemoryStream();
   serializer.WriteObject(ms, Robject);

   // create an instance of the web service
   ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();
```

```
 // Call the Get Method.
   StockSalesOrder foundObject =
       client.GetStockSalesOrder(Encoding.UTF8.GetString(ms.ToArray()));

   //Close the client connection
   client.Close();

   //now that we have the PO that we want to modify, lets set up the properties
   // that need modifications.  Note: only the changed properties will be modified.
   foundObject.Customer = "New Customer";
    etc….

   // now let's increase line item one quantity by one
   foreach (StockPurchaseOrderLineItem li in foundObject.LineItems)
   {
       if (li.LineNumber == 1)
       {
          li.Quantity = li.Quantity+1;
       }
   }

   // now let's Add a new line Item
   StockSalesOrderLineItem NewLine = new StockSalesOrderLineItem();
   NewLine.StockNumber = "StockNumber3";
   NewLine.LineNumber = 3;
    etc…
   foundObject.LineItems.Add(NewLine);



   // Convert the RequestObject Object to JSON
   var Updateserializer = new DataContractJsonSerializer(typeof(StockPurchaseOrder));
   var Updatems = new MemoryStream();
   Updateserializer.WriteObject(Updatems, foundObject);

   // create an instance of the web service
   ServiceReference2.ServiceClient UpdateClient = new ServiceReference2.ServiceClient();

   string Message= UpdateClient.UpdateStockSalesOrder(Encoding.UTF8.GetString(Updatems.ToArray()));

   //Close the client connection
   UpdateClient.Close();

   return Message;
}
```

ReturnStock (StockReturnTransaction returnTransaction)

Description: This method is used to return issued out Stocks.

Paramaters: StockReturnTransaction object

Return: Any errors in a string representation. An empty return value means operation completed successfully.

Sample C# Code:

```
 private string ReturnStock()
{
     string returnedMessage = string.Empty;
     StockReturnTransaction returnTransaction = new StockReturnTransaction ();
     returnTransaction.SecurityToken = "Your Security token";
     returnTransaction.CustomerCode = "Customer Code";
     returnTransaction.SalesOrderNumber = "Sales Order # if any";
     returnTransaction.Notes = "Transaction Notes";

     // now we will add the line items

     StockReturnTransactionLineItem lineItem1 = new StockReturnTransactionLineI-
tem();
     lineItem1.InventoryNumber = "InventoryNumber to Return";
     lineItem1.LocationFrom = "To Location";
     lineItem1.Price = 6;
     lineItem1.QuantityToReturn = 1;
     lineItem1.SalesOrderLineNumber = 2;
     returnTransaction.LineItems.Add(lineItem1);

     StockReturnTransactionLineItem lineItem2 = new StockReturnTransactionLineI-
tem ();
     lineItem2.InventoryNumber = " InventoryNumber to Return ";
     lineItem2.LocationFrom = " To Location ";
     lineItem2.Price = 6;
     lineItem2.QuantityToReturn = 1;
     lineItem2.SalesOrderLineNumber = 2;
     returnTransaction.LineItems.Add(lineItem2);

       etc…

     // Convert the Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(StockReturnTransac-
tion));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, returnTransaction);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Checkout Method.
     returnedMessage = client.ReturnStock(Encoding.UTF8.GetString(ms.ToArray()));
     //Close the client connection
     client.Close();

      return returnedMessage;
 }
```

GetAssetAllHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset All History.

Paramaters: HistoryRequestObject object

Return: Array of AssetAllHistory Objects with Most recent transactions first.

Sample C# Code:

```
 public AssetAllHistory[] GetAssetAllHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench' AND
InvHistoryType =
                              'Move'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));

     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
     AssetAllHistory[] foundObject =
          client.GetAssetAllHistoryOrderByMostRecent(Encoding.UTF8.Get-
String(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetReceiveHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Receive History.

Paramaters: HistoryRequestObject object

Return: Array of AssetReceiveHistory Objects with most recent transactions first.

Sample C# Code:

```
public AssetReceiveHistory[] GetAssetReceiveHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
     AssetReceiveHistory [] foundObject =
client.GetAssetReceiveHistoryOrderByMostRecent(Encod-
ing.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetMoveHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Move History.

Paramaters: HistoryRequestObject object

Return: Array of AssetMoveHistory Objects with Most recent transactions first.

Sample C# Code:

```
public AssetMoveHistory[] GetAssetMoveHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();
```

```
public AssetReceiveHistory[] GetAssetReceiveHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
     AssetReceiveHistory [] foundObject =
client.GetAssetReceiveHistoryOrderByMostRecent(Encod-
ing.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetMoveHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Move History.

Paramaters: HistoryRequestObject object

Return: Array of AssetMoveHistory Objects with Most recent transactions first.

Sample C# Code:

```
public AssetMoveHistory[] GetAssetMoveHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
```

```
// Call the Get Method.
     AssetMoveHistory [] foundObject =
    client.GetAssetMoveHistoryOrderByMostRecent(Encod-
ing.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetDisposeHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Dispose History.

Paramaters: HistoryRequestObject object

Return: Array of AssetDisposeHistory Objects with Most recent transactions first.

Sample C# Code:

```
public AssetDisposeHistory[] GetAssetDisposeHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();

     // Call the Get Method.
     AssetDisposeHistory [] foundObject =
    client.GetAssetDisposeHistoryOrderByMostRecent(Encod-
ing.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetCheckoutHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Checkout History.

Paramaters: HistoryRequestObject object

Return: Array of AssetCheckoutHistory Objects with Most recent transactions first.

Sample C# Code:

```
public AssetCheckoutHistory[] GetAssetCheckoutHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestOb-
ject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceCli-
ent();

     // Call the Get Method.
     AssetCheckoutHistory [] foundObject =       client.GetAssetCheckoutHistory-
OrderByMostRecent(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetAssetCheckinHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Asset Checkin History.

Paramaters: HistoryRequestObject object

Return: Array of AssetCheckinHistory Objects..

Sample C# Code:

```
public AssetCheckinHistory[] GetAssetCheckinHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e AssetNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
     return foundObject;
 }
```

```
// Call the Get Method.
     AssetCheckinHistory [] foundObject =
    client.GetAssetCheckinHistoryOrderByMostRecent(Encod-
ing.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetStockReceiveHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Stock Receive History.

Paramaters: HistoryRequestObject object

Return: Array of StockReceiveHistory Objects.

Sample C# Code:

```
public StockReceiveHistory[] GetStockReceiveHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e StockNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();

     // Call the Get Method.
     StockReceiveHistory [] foundObject =
client.GetStockReceiveHistory(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetStockMoveHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve StockMove History.

Paramaters: HistoryRequestObject object

Return: Array of StockMoveHistory Objects.

Sample C# Code:

```
public StockMoveHistory[] GetStockMoveHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e StockNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();

     // Call the Get Method.
     StockMoveHistory [] foundObject =
    client.GetStockMoveHistory(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetStockDisposeHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve StockDispose History.

Paramaters: HistoryRequestObject object

Return: Array of StockDisposeHistory Objects.

Sample C# Code:

```
public StockDisposeHistory[] GetStockDisposeHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e StockNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

 // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
// Call the Get Method.
     StockDisposeHistory [] foundObject =
    client.GetStockDisposeHistory(Encoding.UTF8.GetString(ms.ToArray()));
```

```
//Close the client connection
     client.Close();
     return foundObject;
 }
```

GetStockIssueHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve StockIssue History.

Paramaters: HistoryRequestObject object

Return: Array of StockIssueHistory Objects.

Sample C# Code:

```
public StockIssueHistory[] GetStockIssueHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e StockNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();

     // Call the Get Method.
     StockIssueHistory [] foundObject =
   client.GetStockIssueHistory(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetStockReturnHistory (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve StockReturn History.

Paramaters: HistoryRequestObject object

Return: Array of StockReturnHistory Objects.

Sample C# Code:

```
public StockReturnHistory[] GetStockReturnHistory()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e StockNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();

     // Call the Get Method.
     StockReturnHistory [] foundObject =
   client.GetStockReturnHistory(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetStockReceiveHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve Stock Receive History.

Paramaters: HistoryRequestObject object

Return: Array of StockReceiveHistory Objects with most recent transactions first.

Sample C# Code:

```
public StockReceiveHistory[] GetStockReceiveHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e StockNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

// create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
```

```
// Call the Get Method.
     StockReceiveHistory [] foundObject =
client.GetStockReceiveHistoryOrderByMostRecent(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetStockMoveHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve StockMove History.

Paramaters: HistoryRequestObject object

Return: Array of StockMoveHistory Objects with Most recent transactions first.

Sample C# Code:

```
public StockMoveHistory[] GetStockMoveHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e StockNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();

     // Call the Get Method.
     StockMoveHistory [] foundObject =
   client.GetStockMoveHistoryOrderByMostRecent(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetStockDisposeHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve StockDispose History.

Paramaters: HistoryRequestObject object

Return: Array of StockDisposeHistory Objects with Most recent transactions first.

Sample C# Code:

```
public StockDisposeHistory[] GetStockDisposeHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e StockNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();

     // Call the Get Method.
     StockDisposeHistory [] foundObject =
   client.GetStockDisposeHistoryOrderByMostRecent(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

GetStockIssueHistoryOrderByMostRecent (HistoryRequestObject paramHistoryRequest)

Description: This method is used to retrieve StockIssue History.

Paramaters: HistoryRequestObject object

Return: Array of StockIssueHistory Objects with Most recent transactions first.

Sample C# Code:

```
public StockIssueHistory[] GetStockIssueHistoryOrderByMostRecent()
{
     //Create a RequestObject
     HistoryRequestObject Robject = new HistoryRequestObject();

     //fill in the Parameter values
     Robject.SecurityToken = "Your Security token";
     Robject.SqlFilter = " Your filter if any i.e StockNumber = 'Wrench'";

     // Convert the RequestObject Object to JSON
     var serializer = new DataContractJsonSerializer(typeof(HistoryRequestObject));
     var ms = new MemoryStream();
     serializer.WriteObject(ms, Robject);

     // create an instance of the web service
     ServiceReference2.ServiceClient client = new ServiceReference2.ServiceClient();
```

```
// Call the Get Method.
     StockIssueHistory [] foundObject =
    client.GetStockIssueHistoryOrderByMostRecent(Encoding.UTF8.GetString(ms.ToArray()));

     //Close the client connection
     client.Close();
     return foundObject;
 }
```

Dependencies

Project

a. Reference to BarCloudWCF Service “http://api.asapsystems.com/Service.svc”

b. GET methods to work with web browser languages like PHP use ”http://api2.asapsystems.com/Service.svc”.
Use these methods that end with “_GET”.
For example:

GetAsset => GetAsset_GET
Other example
AddAsset => AddAsset_GET

c. Reference BarCloudDL.dll can be downloaded from
www.asapsystems.com/Delivery/BarCloud/API/BarCloudDL.zip

d.Net Framework 4

Classes and forms

a. System.Runtime.Serialization.Json

b. System.IO