








Rain Webservice User Manual

1. Webservice Platform

Rain webservices developed by Microsoft .Net Framework and WCF technology.

This service is a windows service and called as "Rain Agent".

 Radio Management Service	Radio Management and Airplane Mode Service
 Rain Agent	Provide authority, validation and ... Rain services.
 Rain Customer Service	Validate installed client customers
 Rain Data Distributer	Provide distribute data service between Rain clients.
 Rain Devices	Check peripheral devices to system assistant
 Rain Scheduler Service	
 Remote Access Auto Connection Manager	Creates a connection to a remote network whenever a p

Default URL of this service is :

<http://localhost:6800/RainService/Sale/>

It has a custom credential that can be configure in system configurations.

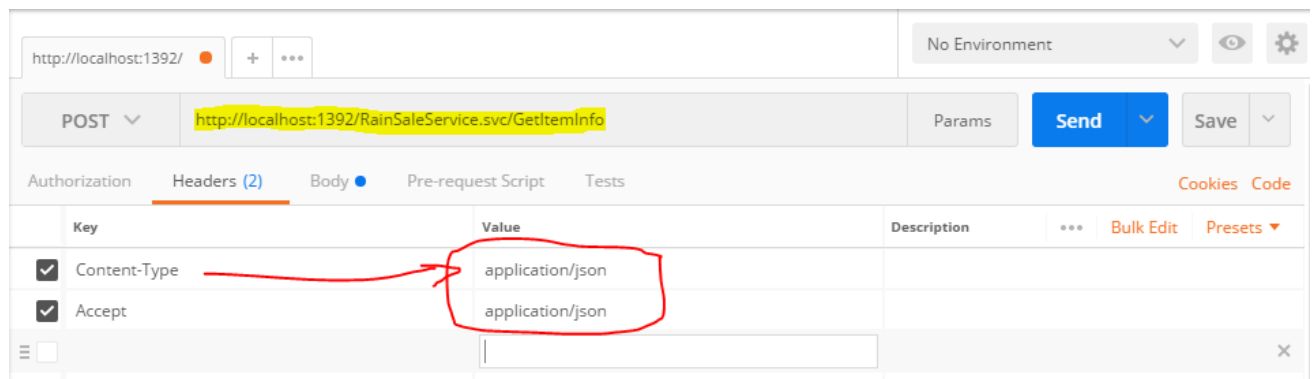
Default Credential :

User : Rain

Password : 123

2. Use the service in Rest mode

In another platforms except .Net Framework you can this functions in Ajax mode. Please, see this samples :



The screenshot shows a REST client interface with the following details:

- URL: `http://localhost:1392/RainSaleService.svc/GetItemInfo`
- Method: `POST`
- Headers (2):
 - `Content-Type`: `application/json`
 - `Accept`: `application/json`

Authorization Headers (2) **Body** Pre-request Script Tests

form-data x-www-form-urlencoded **raw** binary **JSON (application/json)**

```

1 {
2   "barcode" : "6260108213878",
3   "stockId" : "7A9AC053-4D21-4B38-B5A1-6009363FF253"
4 }

```

POST http://localhost:1392/RainSaleService.svc/GetItemInfo Params **Send** Save

Pretty Raw Preview **JSON**

```

1 {
2   "GetItemInfoResult": {
3     "Barcode": "6260108213878",
4     "BonusCount": 0,
5     "BulkPrice": 0,
6     "ConsumerPrice": 180650,
7     "ContractID": "d00750c6-aa4e-4edf-863e-08e67265d4b9",
8     "ContractTypeID": 0,
9     "Cost": 113000,
10    "CurrentDiscount": 0,
11    "DamagedCount": 0,
12    "DepartmentID": "0d2ffce0-a4d5-4fbe-b9a0-bab5ef517b2d",
13    "DepartmentName": "سايغ ظرف",
14    "Description": null,
15    "IsNotFound": false,
16    "IsNotInOrder": false,
17    "ItemID": "1d9ee527-3621-4505-84ec-0016567f999f",
18    "LineItemID": 0,
19    "MaximumOrderOverage": 0,
20    "Name": "سايغ ظرف 4 ليتير جام",
21    "OrderCount": 0,
22    "PackCount": 0,
23    "PackNumber": null,
24    "PackUnitCount": 0,
25    "ParentID": "",
26    "Price": 102760,
27    "Quantity": 0,
28    "RecieveMoreThanOrder": false,
29    "ScannedCount": 0,
30    "StockQuantity": 50,
31    "StockSection": ""

```

Credential of service :

Authorizaiotn type : Basic Authentication

User Name and Password : One of windows users that service hosted on it.

Authorization **Basic Auth** Headers (3) Body Pre-request Script Tests

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables.

Username administrator

Password *****

Show Password

Preview Request

You can see the header, body and result of request in Postman application in JSON serialization mode.

3. Service Functions for Sale

List of some implemented functions for sale system :

- GetCustomer
- GetCustomerByCode
- GetCustomerCredit
- AddCustomerFavorites
- RemoveCustomerFavorites
- GetAvailableGiftCards
- UpdateGiftCardStatus
- GetItemInfo
- SaveSaleInvoice
- CalcInvoice
- VoidOrder
- GetOrder
- GetOrderByInvoiceNumber
- SaveCustomer
- GetCustomerPurchaseHistory
- GetCustomerPurchaseHistoryDetail
- UseCustomerCredit
- CreateGiftCard
- GetAllStock
- BookStockDocument
- ReturnInvoiceModel
- SaveReturnInvoice

4. Structure of implemented functions

- Structure of GetCustomer method
Description : Return customer information as serialized XML data.

```
CustomerInfo GetCustomer(Guid customerId);
```

Input : Customer UniquelIdentifier

Output : Customer info as XML

-
- Structure of GetCustomerByCode method

Description : Return customer information as serialized XML data.

```
CustomerInfo GetCustomerByCode(string customerCode, string password);
```

Input :

- 1- CustomerCode as string
- 2- Password as string

Output : Customer info as XML

- Structure of GetCustomerCredit method

Description : Return customer current credit & points as a data structure

```
CustomerCredit GetCustomerCredit(Guid customerID);
```

Input : Customer UniquelIdentifier

Output : CustomerCredit data structure

```
public class CustomerCredit
{
    public decimal TotalPoint { set; get; }
    public decimal CurrentPoint { set; get; }
    public decimal CurrentCredit { set; get; }
    public decimal RemainCredit { set; get; }
}
```

- Structure of AddCustomerFavorites methods

Description : Add customer favorites such as product, store and etc.

```
void AddCustomerFavorites(Guid customerID, int typeID, string id);
```

Input :

- 1- Customer ID
- 2- Type of entity such as product equal to 275
- 3- Product Id or Store Id

Output : None

- Structure of RemoveCustomerFavorites methods

Description : Remove customer favorites such as product, store and etc.

```
void RemoveCustomerFavorites(Guid customerID, int typeID, string id);
```

Input :

- 1- Customer ID
- 2- Type of entity such as product equal to 275
- 3- Product Id or Store Id

Output : None

- Structure of GetAvailableGiftCards methods
Description : Return active gift cards by serial number
`List<GiftCard> GetAvailableGiftCards(string serialNumber string password);`

Input : Serial number of gift card

Output : Return array of GiftCard data structure

```
public class GiftCard
{
    public string ID { set; get; }
    public string SerialNumber { set; get; }
    public string Name { set; get; }
    public decimal Amount { set; get; }
    public decimal Thershold { set; get; }
    public int TypeID { set; get; }
}
```

- Structure of UpdateGiftCardStatus method
Description : Update gift card status after use.

```
void UpdateGiftCardStatus(Guid id, string serialNumber, decimal amount,
    int status);
```

Input :

- 1- Gift card uniqueidentifier
- 2- Gift card serial number
- 3- Status Id
 - a. Created = 352
 - b. Sold = 353
 - c. Paid = 354
 - d. Voided = 355

Output : None

-
- Structure of GetItemInfo methods

Description : Return information of an item with stock quantity

```
ItemInfo GetItemInfo(string barcode, Guid stockId);
```

Input : Item barcode, with stock identifier (optional)

Output : Return ItemInfo data structure

```
public class ItemInfo
{
    public int LineItemID { set; get; }
    public string ContractID { set; get; }
    public int ContractTypeID { set; get; }
    public decimal SupplierDiscountPercent { set; get; }
    public string ItemID { set; get; }
    public string ParentID { set; get; }
    public int TypeID { set; get; }
    public string DepartmentID { set; get; }
    public int UnitOfMeasureID { set; get; }
    public string Barcode { get; set; }

    public string Name { set; get; }
    public string PackNumber { set; get; }
    public string StockSection { set; get; }
    public decimal StockQuantity { set; get; }
    public decimal Cost { set; get; }
    public decimal Price { set; get; }
    public decimal ConsumerPrice { set; get; }
    public decimal BulkPrice { set; get; }
    public decimal CurrentDiscount { set; get; }
    public decimal TemporaryPrice { set; get; }
    public decimal OrderCount { get; set; }
    public decimal ScannedCount { get; set; }
    public decimal PackCount { get; set; }
    public decimal PackUnitCount { get; set; }
    public decimal Quantity { get; set; }
    public decimal BonusCount { get; set; }
    public decimal DamagedCount { get; set; }
    public bool IsNotFound { get; set; }
    public bool IsNotInOrder { get; set; }
    public bool RecieveMoreThanOrder { get; set; }
    public decimal MaximumOrderOverage { get; set; }
    public string DepartmentName { set; get; }
    public string Description { set; get; }
}
```

-
- Structure of SaveSaleInvoice method
Description : Create new order or edit an order

```
OperationResult SaveSaleInvoice(OrderModel order, bool calcPromotion, bool calcTax,  
bool allowToMakeInvoice, bool useCredit);
```

Input : OrderModel data structure

```
public class OrderModel  
{  
  
    public Guid Id { set; get; } //Order Id  
  
    public Guid CustomerId { set; get; }  
  
    public int StoreId { set; get; } //Barnch Id  
  
    public Guid UserId { set; get; } //Creation User Id  
  
    public DateTime DeliveryDate { set; get; }  
  
    public TimeSpan DeliveryFromTime { set; get; }  
  
    public TimeSpan DeliveryToTime { set; get; }  
  
    public decimal DeliveryCost { set; get; }  
  
    public decimal ExpectedAmount { set; get; }  
  
    public int DeliveryTypeId { set; get; }  
  
    public string Address { set; get; }  
  
    public List<OrderItemModel> Items { set; get; } // Array of products  
  
    public List<OrderDiscount> Discounts { set; get; } // Array of discount  
  
    public List<PaymentModel> Payments { set; get; } // Array of payment types  
}  
  
public class OrderItemModel  
{  
  
    public Guid OrderId { set; get; } // Order Id  
  
    public int LineItemID { set; get; } // Row Number  
  
    public Guid ItemId { set; get; } // Product Id  
  
    public bool IsDelivery { set; get; } // Allow to delivery  
  
    public decimal Quantity { set; get; }  
}
```

```
    public decimal Price { set; get; }
    public decimal Discount { set; get; }
    public List<OrderDiscount> Discounts { set; get; } // Array of discount
    public decimal Tax { set; get; }
    public decimal NetAmount { set; get; }
    public int ReasonID { set; get; } // Reason of return item
    public string Description { set; get; }
    public SqlAction OperationType { set; get; } // Save or Delete
}
```

```
public class OrderDiscount
{
    public int ReasonID { set; get; }
    public bool AllowMerge { set; get; }
    public decimal DiscountAmount { set; get; }
    public decimal DiscountPercent { set; get; }
}
```

```
public class PaymentModel
{
    public int LineItemID { set; get; } // Row Number
    public int TypeID { set; get; } // Type : POS, Gift, Cash
    public decimal Amount { set; get; }
    public decimal ExchangeAmount { set; get; } //For example : Dollar rate
    public DateTime DueDate { set; get; }
    public DateTime PaymentDate { set; get; }
    public Guid GiftCardID { set; get; }
    public string SerialNumber { set; get; }
    public string BankName { set; get; }
    public string AccountNumber { set; get; }
    public string TerminalID { set; get; }
}
```

```
    public string PANNumber { set; get; }  
    public string ReferenceNumber { set; get; }  
    public string SVCNumber { set; get; }  
}
```

Output : OperationResult data structure

```
public class OperationResult  
{  
    public object Index { get; set; } //  
    public ProgressStatus Status { get; set; } // Succeed or Error  
    public string Message { get; set; } // Description or Exception message  
}
```

- Structure of CalcInvoice method
Description : Get an order data and calculate promotions & tax on it and return it.

```
OrderModel CalcInvoice(OrderModel order, bool calcPromotion, bool calcTax);
```

Input : OrderModel data structure such as SaveSaleInvoice method

Output : OrderModel data structure such as SaveSaleInvoice method

- Structure of VoidOrder method
Description : Void or cancel an order by customer decision

```
OperationResult VoidOrder(OrderModel order);
```

Input : OrderModel data structure such as SaveSaleInvoice method

Output : OperationResult data structure such as SaveSaleInvoice method

- Structure of GetOrder method
Description : Get an order to edit or view

```
OrderModel GetOrder(int storeID, Guid orderID);
```

Input :

- 1- Store Id
- 2- Order Id

Output : OrderModel data structure such as SaveSaleInvoice method

-
- Structure of GetOrderByInvoiceNumber method
Description : Get an order to edit or view

```
OrderModel GetOrderByInvoiceNumber(string invoiceNumber);
```

Input : Invoice Number

Output : OrderModel data structure such as SaveSaleInvoice method

- Structure of SaveCustomer method
Description : Create new customer or edit customer information

```
Guid SaveCustomer(CustomerInfo customer);
```

Input : CustomerInfo data structure

```
public class CustomerInfo
{
    public Guid CustomerID { set; get; }
    public string FirstName { set; get; }
    public string LastName { set; get; }
    public string CustomerCode { set; get; }
    public string GroupName { set; get; }
    public string Password { set; get; }
    public string NationalID { set; get; }
    public object CustomerImage { set; get; }
    public string CustomerImageUrl { set; get; }
    public string Birthday { set; get; }
    public string Phone { set; get; }
    public string Mobile { set; get; }
    public string Email { set; get; }
    public string Address { set; get; }
    public decimal Latitude { set; get; }
}
```

```
public decimal Longitude { set; get; }
public int StoreID { set; get; } // Default or nearest branch
public string StoreName { set; get; }
public decimal CurrentPoint { set; get; }
public decimal CurrentCredit { set; get; }
public decimal TotalPurchase { set; get; }
public bool IsMale { set; get; }
public Guid AgentID { set; get; }
public bool IsActive { set; get; }
public string CreationDate { set; get; }
}
```

Output : Customer Id

- Structure of UseCustomerCredit method
Description : Increase or decrease the customer credit or point

```
OperationResult UseCustomerCredit(Guid customerId, Guid groupId, decimal amount,
decimal point, string description)
```

Input :

- 1- CustomerId
- 2- GroupId
- 3- Amount
- 4- Point
- 5- Description

Output : OperationResult data structure such as SaveSaleInvoice method

- Structure of CreateGiftCard method
Description : Make a gift card or a boun for a customer

```
string CreateGifCard(string serial, decimal amount, string startDate, string
expireDate)
```

Input :

- 1- Serial
 - 2- Amount
 - 3- StartDate
 - 4- ExpireDate
-

Output : Return the serial number if it accepted

- Structure of GetCustomerPurchaseHistory method
Description : Get simple customer purchase history by products

`SelectedDataCollection` GetCustomerPurchaseHistory(Guid customerID);

Input : Customer Id

Output : Array of SelectedData data structure

```
public class SelectedData
{
    public object Index { get; set; } // Product Id
    public string Text { get; set; } // Product Name
    public string SecondText { get; set; } // Store Name
    public decimal Value { set; get; } // Quantity
    public decimal SecondValue { set; get; } // Amount
    public string Code { get; set; } // Barcode
}
```

-
- Structure of GetCustomerPurchaseHistoryDetail method
Description : Retrieve list of customer purchases

Input : -

- 1 – customerId
- 2 – startDate
- 3 – endDate
- 4 - languageld

Output : Array of CustomerPurchase data structure

```
public class CustomerPurchase
{
    public string StoreName { set; get; }
    public string Date { set; get; }
    public string InvoiceNumber { set; get; }
    public string Type { set; get; }
    public string ItemName { set; get; }
    public string Barcode { set; get; }
    public decimal Quantity { set; get; }
    public decimal Amount { set; get; }
    public decimal Discount { set; get; }
    public decimal Tax { set; get; }
}
```

-
- Structure of GetAllStock method
Description : Retrieve all stocks information

Input : -

Output : Array of StockInfo data structure

```
public class StockInfo
{
    public string StockID { get; set; }
    public string StockName { get; set; }
}
```

- Structure of BookStockDocument method
Description : Create new stock document

Input : DocumentInfo data structure

```
public class DocumentInfo
{
    public List<ItemInfo> Items { get; set; }
    public string CreationUser { get; set; }
    public string ExpectedRecieveDate { get; set; }
    public string DocumentDate { get; set; }
    public string TargetStock { get; set; }
    public string TargetStockID { get; set; }
    public string SourceStock { get; set; }
    public string SourceStockID { get; set; }
    public string SupplierName { get; set; }
    public bool IsBonus { get; set; }
    public string DepartmentID { get; set; }
    public string CustomerID { get; set; }
    public string SupplierID { get; set; }
    public string DocumentNumber { get; set; }
    public int Reason { get; set; }
    public int UsageType { get; set; }
}
```

```
public int DocumentType { get; set; }
public string DocumentID { get; set; }
public int BookerWorkstationID { get; set; }
public int BookerStoreID { get; set; }
public string ThirdPartyID { get; set; }
public string Comment { get; set; }
}
```

```
public class ItemInfo
{
    public decimal MaximumOrderOverage { get; set; }
    public bool RecieveMoreThanOrder { get; set; }
    public bool IsNotInOrder { get; set; }
    public bool IsNotFound { get; set; }
    public decimal DamagedCount { get; set; }
    public decimal BonusCount { get; set; }
    public decimal Quantity { get; set; }
    public decimal PackUnitCount { get; set; }
    public decimal PackCount { get; set; }
    public decimal ScannedCount { get; set; }
    public decimal OrderCount { get; set; }
    public decimal TemporaryPrice { get; set; }
    public decimal CurrentDiscount { get; set; }
    public decimal BulkPrice { get; set; }
    public decimal ConsumerPrice { get; set; }
    public decimal Price { get; set; }
    public decimal Cost { get; set; }
    public int LineItemID { get; set; }
    public string ContractID { get; set; }
    public int ContractTypeID { get; set; }
}
```

```
public decimal SupplierDiscountPercent { get; set; }
public string ItemID { get; set; }
public string ParentID { get; set; }
public string DepartmentName { get; set; }
public int TypeID { get; set; }
public int UnitOfMeasureID { get; set; }
public string Barcode { get; set; }
public string Name { get; set; }
public string PackNumber { get; set; }
public string StockSection { get; set; }
public decimal StockQuantity { get; set; }
public string DepartmentID { get; set; }
public string Description { get; set; }
}
```

Output : OperationResult data structure such as SaveSaleInvoice method

- Structure of ReturnInvoiceModel method
Description: This method return items that can be return

```
OrderModel ReturnInvoiceModel(string invoiceNumber)
```

- Structure of SaveReturnInvoice method
Description: This method save an return invoice with refer to another invoice

```
OperationResult SaveReturnInvoice(OrderModel order, string invoiceNumber)
```
