**Root: “/”**

Pull Sent For Login: {

Username: “Cade”,

Password: “1234”,

Status: “authorizing”

}

Pull Received For Login {

Username: “Cade”,

Role: “admin”,

Status: “Authorized”

}

Put Sent For register: {

Name: “Cade Powers”,

Username: “Cade”,

Password: “1234”,

EmployeeId: 2345,

Address: “2134 Red Wolf Jonesboro, AR, 72142”,

PhoneNumber: “501-515-4086”,

Role: “admin”

}

Put Received For Register {

Username: “Cade”,

Role: “admin”,

Status: “Authorized”

}

**Root: “/ar”**

Sent Pull request

Respond {{

AccountNumer: 001,

CompanyName: “Redstone”,

AmountOwed: 1200.23,

DateDue: “10/05/20”

},{

…

},…}

List of all the accounts with outstanding debt

**Root: “/ar/paid”**

Post Sent: {

AccountNumber: 001,

AmountPaid: 1000.00,

}

Post Received {

Status: “posted”

}

**Root: “/im/pm”**

Pull Sent: {

PartNumberPattern: “\*125A”

}

Pull Received: {{

PartNumber: “45125A”,

Description: “Gear”,

Quantity: 2,

Price: 120.00

},{

PartNumber: “523125A”

…

},…}

List of part numbers containing sent substring

**Root: “/im/pm/detailed”**

Pull Sent: {

PartNumber: “45125A”

}

Pull Received: {

QuantityOO: 2,

Cost: 50.00,

Brand: “Cobra”,

Source: “Intercontinental”,

ClassID: 001

}

Post Sent: {

PartNumber: “45125A”,

Quantity: 4,

QuantityOO: -2

}

Behavior of QuantityOO, The specified value sent is relative to the current value in DB. So -2 would subtract 2 from the current value in DB. Likewise, 2 would add 2 to the current value of DB.

Post Received: {

Status: “posted”

}

**Root: “/im/pm/add”**

Put Request: {

PartNumber: “101-45214-000”,

\*Description: “Freightliner engine mount”,

Quantity: 8,

\*QuantityOO: 2,

Price: 50.00,

\*Brand: “Freightliner”,

Cost: 25.00,

\*Source: “Freightliner”

ClassID: 005

}

Put Received: {

Status: “put”

}

‘\*’ delineate optional parameters should have default values if one is not provided.

**Root: “/im/classm”**

Pull Sent: {}

Pull Received: {{

ClassID: 001,

ClassDescr: “Gears”,

Margin1: 1.2,

Margin2: 1.6,

Margin3: 2.0

}, {

…

},…}

In the class management nothing will be sent. The frontend will expect a list of all classes present in the db.

**Root: “/im/classm/update”**

Post Sent: {

ClassID: 001,

ClassDescr: “Flywheels”,

Margin1: 1.4

}

Post Received: {

Status: “posted”

}

In an update for a class the fields in the sent should be updated while the ones not provided should be left alone. If the classID sent is not in the db it should be added with what is included in the sent and what is not included should be defaulted.

**Root: “/custm”**

Pull Sent: {

Name: “Red\*”

}

Pull Received: { {

ID: 001,

Name: “Redstone”,

Addr: “1234 Redstone St. Ruelle, AR, 72451”,

Phone: “501-405-5124”

}, {

…

},…}

List of all customers whose name contains the substring Sent

**Root: “custm/detailed”**

Pull Sent: {

ID: 001

}

Pull Received: {

BillingAddr: “POBox 203 Ruelle, AR, 72451”,

ShippingAddr: “ATTN: Robby 1234 Redstone St. Ruelle, AR, 72451

CityTax: 0.05,

StateTax: 0.1,

FederalTax: 0.025,

Type: “Charge”

Post Sent: {

ID: 001,

CityTax, 0.06

}

Post Received: {

Status: “posted”

}

**Root “/custm/add”**

Put Sent: {

Name: “Cade Powers”,

Addr: “245 Redwolf Blvd. Jonesboro, AR, 72154”,

Phone: “501-515-4088”,

Type: “Cash”

}

Put Received: {

Status: “put”

}

Values not provided should be defaulted and cash accounts should use local tax requirements which will be global variables. If this account was a charge account tax information would be necessary.

**Root “/em”**

**Root “/em/detailed”**

**Root “/em/add”**

Employee management will follow the same pattern as the customer management with the fields adjusted.

**Root “/pc/inv”**

Pull Sent {

CustomerName: “Red\*”

} OR {

ID: 30001

}

If CustomerName is in the sent, then the server should return a list of all invoices with the customer containing the sent pattern and they should be sorted in descending order based on date. If the ID is sent, then the server should return the details of the invoice under that ID.

Pull Received if name is sent: {{

ID: 30001,

Customer: “Redstone”,

Date: “10/07/2020”,

FirstPart: “101-45214-000”,

Total: 1000.23

},{

…

},…}

Pull Received if ID is sent {

A list of all details about the invoice that has the ID sent.

}

**Root “pc/invoice/add”**

Pull Sent: {

CustomerID: “001”

}

Pull Received: {

Name: “Redstone”,

BillingAddr: “POBox 203 Ruelle, AR, 72451”,

ShippingAddr: “ATTN: Robby 1234 Redstone St.

Ruelle, AR, 72451”,

Phone: “501-405-5124”

}

Pull Sent: {

PartNumber: “1245AV”

}

Pull Received: {

Description: “Gear”,

PriceM1: 140.00,

PriceM2: 160.00,

PriceM3: 200.00,

Cost: 100.00,

Quantity: 2

}

Sent for each part added to invoice

Pull Sent: {

CustomerID: “001”,

Parts: “{1245AV, 2,160.00},{101-45251-000,4,80.00},…”

}

Pull Received: {

SubTotal: 640.00,

TaxTotal: 112.00,

Total: 752.00

}

Sent to get totals after all parts are added

Put Sent: {

CustomerID: 001,

Parts: “{1245AV, 2,160.00},{101-45251-000,4,80.00},…”,

SubTotal: 640.00,

TaxTotal: 112.00,

Total: 752.00

}

Put Received: {

InvoiceID: 30002

}

Once the invoice is ready to be posted the final put request is sent.

**Root: “pc/quote”**

**Root: “pc/quote/add”**

Follows patter of invoice with the id’s first number decrementing to 2. For example an invoice ID would be 3001 and a quote ID would be 1001.

**Root: “pc/workOrder”**

**Root: “pc/workOrder/add”**

**Root: “pc/workOrder/update”**

Follows the pattern of invoice but the Id is 2001 and fields are adjusted properly.

**Root: “pc/history”**

Pull Sent: {

\*Type: 1,

\*Customer: “Red\*”,

\*PartNumber: “101-254\*”,

\*Date: “10/05/2020”

}

Pull Received: {

A list where the patterns sent are held in the invoices.

}

Any combination of these parameters should be accepted. Type tells the server where to pull the history from. The number is related to the initial number in the ID’s. So 1 would be quotes, 2 would be work orders, and 3 would be invoices.