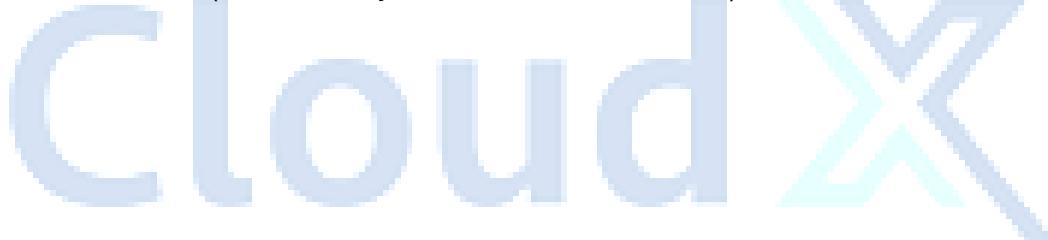


# API SaaS POS Proposal

## Table of Contents

### Contents

API SaaS POS Proposal .....	1
Table of Contents .....	1
Message / Brief.....	2
Specifications in Our Own Words.....	3
Specific Key Features: .....	5
Technical Specifications .....	8
Screenshots.....	13
Sage Theme .....	20
References.....	22
API's We Would Need (API Calls may differ from documentation) .....	23



## Message / Brief

### **Subject: Proposal to Develop a SaaS POS Integrated with the Sage Ecosystem**

Dear Sir/Madam,

I hope you are doing well.

The POS system we are developing aims to be a web-based application that offers the feel of a lightweight desktop program, optimized for both desktop and tablet use. It will integrate directly with Sage Pastel via their API, allowing users to connect to multiple companies just as they would within Sage itself. Upon login, the POS will retrieve the companies linked to the user's Sage account and provide a simple dropdown to switch between them. Each company will load its own dashboard and data, enabling smooth and seamless transitions.

The transaction screen will not rely on picture-based or touchscreen-style layouts; instead, it will feature a familiar Microsoft Excel-like interface where products can be added line by line with inline product search for quick access. While a visual, picture-based POS interface is a nice-to-have, it is not essential.

We plan to include all basic POS features such as sales processing and refunds, along with essential reporting functions like sales summaries. Most customer data will remain managed on the Sage side, with the POS interacting via API as needed.

Design-wise, it is important that the POS closely matches the look and feel of Sage Pastel One—including fonts, colors, and overall layout—so users feel they are within the same ecosystem.

From a technical standpoint, this will be a SaaS solution with a minimal and scalable database design, supporting a multi-tenant structure where each Sage company acts as a tenant. Subscription-based access will also be supported. We will supply the necessary API documentation and access for integration and testing purposes.

Thank you for your continued support and collaboration. Please let me know if you require any further information or clarification.

Kind regards,  
Bradley Steynbergh

## Specifications in Our Own Words

### ◆ Main Goal

- We want a **desktop/tablet-friendly web POS application** that integrates with Sage Pastel using their API functions.
  - The POS will live on the web but should have the **feel of a lightweight desktop system**.
- 

### ◆ POS Screen Design

- The transaction screen **must not** be a picture-based or touchscreen-type POS.
  - Instead, we want a **Microsoft Excel-style layout**, where each product is added as a **line item/row**.
  - It must have **inline product search** to quickly find and add items.
  - Optionally, it would be nice if users can **switch to a visual (picture) POS screen**, but that's not a must-have.
- 

### ◆ Company Management & Dashboard

- In Sage, you can load multiple **companies** — we want the same behavior in our POS.
  - Once a user logs into the POS, it should:
    - **Connect to Sage**, fetch available companies linked to the account.
    - Show a **company selection dropdown (top-right)** similar to how Sage handles it.
    - When a company is selected, the POS should load that **company's dashboard and data**.
  - Basically, **each company is like its own environment**, and switching between them should be smooth and simple — just like Sage.
- 

### ◆ General POS Features

- POS must have **basic functionality** that any normal POS system would offer (sales, refunds, etc.).
  - **Reporting** should also be available — basic reports like sales, summaries, etc.
  - Most **customer/client data** will be managed from the **Sage side**, not within the POS.
    - The POS will simply **pull or save data via API** as needed.
- 

### ◆ Design Look & Feel

- The goal is for the POS system to **match Sage Pastel One's design**, including:
    - Fonts
    - Colors
    - Layout/theme
  - We want the UI to feel like part of the Sage ecosystem.
- 

#### ◆ Technical Notes

- This will be a **SaaS POS product**, so we need a **minimal database structure** for easy scalability.
- A **multi-tenant setup** is required — each “Company” from Sage acts like a tenant.
- The POS must support **subscription-based access**.
- We'll provide **API documentation and access to Sage Pastel One for testing and integration**.



## Specific Key Features:

### Multi-Tenant Architecture

- Full tenant isolation for clients
- Each client operates within their own secured environment

### Multi-Manager Application (POS Admin / POS Client)

- Role-based access for both system administrators and client-level managers
- Clean separation of permission scope and data visibility

### Single Warehouse Mode (Default)

- Default warehouse setup for clients
- Ideal for small businesses or entry-tier subscriptions

### Manager-to-Manager Delegation

- Admins can transfer or assign management between client managers

### User Limit Enforcement

- Clients are restricted to a user count based on their subscription tier
- System prevents creation of excess users when limits are reached

### Multi-Warehouse Upgrade (Optional)

- Additional warehouse support can be enabled per client
- Available as a paid add-on or part of advanced tiers

### License Suspension & Warning System

- Automatic warnings for inactive, expired, or suspended licenses
- Suspended clients are blocked from creating or editing data

### Central Sage API Control

- All API interactions with Sage are managed at the Admin level
- Includes validation, syncing, and monitoring of connected companies

### Coupon & Gift Management (Client Control)

- Enable or disable access to gift card/coupon tools per client
- Admin-level visibility over usage and setup

### White-Labeled Branding

- Each client can apply their own logo and theme to the POS interface
- Ensures brand consistency for their staff and stores

---

## ■ POS Client Side (Storefront Operations)

This layer includes the features available to businesses using the POS software at store level — covering transactions, inventory, loyalty, branding, and third-party integrations.

---

### **Gift Cards & Coupons**

- Issue, redeem, and track gift vouchers and discount codes
- Optional feature based on admin-level permissions

### **Return Management**

- Process product returns with validation and automated stock update
- Include reasons and staff notes for audit

### **Refunds & Loyalty System**

- Cash/card refunds supported
- Built-in loyalty system with points tracking and customer rewards

### **Store Inventory Limit**

- Restrict number of products per store
- Useful for performance optimization or plan-based limitations

### **Device Limit per Store**

- Control how many terminals or mobile devices can run POS per location

### **Payment Gateway Integration**

- Native support for **Adumo** and **Yoco** payment processing
- Extendable to future payment services

### **Picking Slip Generation**

- Create printable picking slips for fulfilling sales orders or stock movements

### **Global POS Settings**

- Configure global parameters like tax, receipts, currency, operating hours
- Managed by client-level admin

### **Client POS Branding**

- Client-specific logo and theme
- Applies to dashboard, POS screen, receipts

 **Sales Dashboard**

- Real-time sales tracking and reporting
- Filter by date, staff, product, and more

 **Sales Docket Generation**

- Generate printable sales order dockets per transaction
- Optional emailing or SMS integration possible



## Technical Specifications

### Tech Stack

- **Laravel Framework**

Laravel will be used as the base backend framework. It gives us a strong, scalable structure using MVC and is well-suited for a POS system.

- **MySQL or PostgreSQL**

A relational database like MySQL or PostgreSQL will handle all product, transaction, and customer data securely and efficiently.

- **Vue.js (or React)**

We want the frontend to be fast and responsive. Vue.js or React will help us build an interactive UI, which is key for daily POS usage.

- **Bootstrap or Tailwind CSS**

For styling and layout, we'll use either Bootstrap or Tailwind CSS to keep the UI clean and easy to maintain.

- **RESTful API**

Laravel has built-in support for REST APIs, which we'll use to connect frontend and backend, as well as link to the Sage API.

- **Composer**

Laravel's dependency management will be handled using Composer for managing PHP libraries cleanly.

### ◆ Recommended Libraries & Packages

- **Laravel Passport**

For API authentication (OAuth 2.0). This secures all communication and access between services.

- **Laravel Cashier**

If we roll out subscriptions or need Stripe integration for billing, Cashier is ideal.

- **Laravel Horizon**

Will be used for monitoring and managing background tasks and queues, like report generation.

- **Laravel Echo**

For real-time notifications — e.g., new orders, stock updates — using WebSockets.

- **Laravel Scout**

For adding powerful search functionality (Algolia or Elasticsearch support).

- **Laravel Excel**

To support Excel/CSV import/export for bulk uploads or data reporting.

- **Guzzle HTTP**  
For making external HTTP requests securely — essential for API calls (like Sage or payment providers).
  - **PHPUnit**  
Default testing framework in Laravel to ensure the app is stable and reliable through unit/feature tests.
- 

#### ◆ Additional Backend Features & Considerations

- **Queue Workers**  
Handle background processes like sending receipts, processing payments, and more.
- **Localization & Internationalization**  
If the product is used in multiple countries, Laravel's localization features will help switch languages/content.
- **Security**  
We'll use Laravel's built-in protections (CSRF, XSS, validation) to ensure strong app security.
- **Deployment**  
The POS can be deployed on cloud platforms (AWS, Azure), or containerized using Docker for easy scaling.

### POS System Modules

#### 1. User Interface (UI)

- **What it does:** Front-end interface used by staff or customers.
- **Key Features:**
  - Clean and simple UI for speed and ease of use
  - Touchscreen compatibility for devices like tablets
  - Option to customize layout based on store setup

#### 2. Product Management

- **What it does:** Manage products, inventory, and pricing.
- **Key Features:**
  - Store product details (SKU, price, categories, descriptions)
  - Track real-time stock levels

- Support for promotions and discount pricing
- 

### 3. Sales & Checkout

- **What it does:** Handle transactions and customer checkout.
  - **Key Features:**
    - Accept multiple payment types (cash, card, mobile)
    - Manage returns and exchanges
    - Print or email receipts
- 

### 4. Customer Management (CRM)

- **What it does:** Manage customer data and loyalty.
  - **Key Features:**
    - Store profiles and transaction history
    - Manage loyalty points and rewards
    - Provide customer data for targeted marketing
- 

### 5. Reporting & Analytics

- **What it does:** Give insights on performance.
  - **Key Features:**
    - Sales reports (daily/weekly/monthly)
    - Stock movement and inventory levels
    - Analytics on sales trends, profits, and staff performance
- 

### 6. Integrations & APIs

- **What it does:** Link POS to other platforms.
  - **Key Features:**
    - Connect to payment gateways like Stripe
    - Integrate with ERP tools (like Sage)
    - Sync with online shops for shared inventory
-

## 7. Security

- **What it does:** Keep everything secure.
  - **Key Features:**
    - Data encryption for safe data transfer
    - Role-based permissions (cashier, manager, etc.)
    - PCI compliance for payment safety
- 

## 8. Scalability & Performance

- **What it does:** Ensure POS runs fast and can grow.
  - **Key Features:**
    - Use caching and optimized queries
    - Load balancing across servers
    - Cloud-ready deployment support
- 

## 9. Support & Maintenance

- **What it does:** Keep the system healthy.
  - **Key Features:**
    - 24/7 monitoring and alerts
    - Automated updates and patching
    - Training documentation for staff
- 

## 10. Hardware Integration

- **What it does:** Work with POS hardware.
  - **Key Features:**
    - Support for barcode scanners and printers
    - Compatibility with multiple POS terminal models
    - Support for mobile/tablet POS setups
- 

## Summary

This is the **tech foundation and module breakdown** for building a SaaS-based POS system connected to **Sage Pastel One**.



The goal is a **lightweight, scalable, secure POS** that feels familiar to Sage users, runs reliably in high-traffic retail environments, and provides enough flexibility to grow with our customer base.



## Screenshots

The screenshot shows the Sage Accounting dashboard with the following sections:

- To Do List:** Shows 4 Expired Quotes (R71,760.00).
- Banking:** No Bank Accounts set up. Tasks include adding a bank account, capturing expenses, and setting up bank feeds.
- Sales History:** A chart comparing sales from Last Year and This Year across months from Mar to Feb.
- Top Customers by Sales:** No items sold to customers. Tasks include adding an invoice, a customer, or an item.
- Customer Balances - Days Outstanding:** A chart showing the number of days outstanding for different customer balances.
- Top Customers by Outstanding Balance:** A list of customers with their outstanding balances.

The screenshot shows the Sage Accounting dashboard with the following sections:

- Customer Balances - Days Outstanding:** A chart showing the number of days outstanding for different customer balances.
- Top Customers by Outstanding Balance:** A list of customers with their outstanding balances.
- Sales History:** A chart comparing sales from Last Year and This Year across months from Mar to Feb.
- Top Customers by Sales:** No items sold to customers. Tasks include adding an invoice, a customer, or an item.

The screenshot shows the Sage Accounting dashboard with the following sections:

- To Do:** A context menu for 'To Do' items, listing options like Add a Customer, Lists, Transactions, Reports, Special, Time Tracking, and Debtors Manager.
- Banking:** No Bank Accounts set up. Tasks include adding a bank account, capturing expenses, and setting up bank feeds.

Sage | Accounting Search transactions

CloudX DEMO API Database BS

Home ▾ Quick View ▾ Customers ▾ Suppliers ▾ Items ▾ Banking ▾ Accounts ▾ Accountant's Area ▾ Reports ▾ Company ▾ Administration ▾

Information Centre

### Dashboard

To Do List

Name	Description	Due Date	Printed	Amount
+ 4 Expired Quotes R71,760.00				

Sales History

R1  
R1

- Accounting Intelligence Reporting
- Customers
- Suppliers
- Items
- Sales and Purchases
- Banks and Credit Cards
- Accounts
- VAT
- Financial Statements
- Asset Report
- Accountants Reports
- Budget Report
- Other
- Time Tracking
- Debtors Manager

Sage | Accounting Search transactions

Home ▾ Quick View ▾ Customers ▾ Suppliers ▾ Items ▾ Banking ▾ Accounts ▾ Accountant's Area ▾ Reports ▾ Company ▾ Administration ▾

### Dashboard

To Do List

Name	Description
+ 4 Expired Quotes R71,760.00	

Sales History

R1  
R1

- Add an Item
- Lists
- Transactions
- Special
- Reports

- Item Listing
- Sales by Item
- Purchases by Item
- Customer Quotes by Item
- Customer Sales Orders by Item
- Supplier Purchase Orders by Item

- Item Movement
- Item Valuation
- Price List
- Item Bundle
- Item Quantities

Sage | Accounting Search transactions

Home ▾ Quick View ▾ Customers ▾ Suppliers ▾ Items ▾ Banking ▾ Accounts ▾ Accountant's Area ▾ Reports ▾ Company ▾ Administration ▾ CloudX DEMO API Database ▾ BS ▾

**List of Items**

Add Item Import Item Search: Search View: All (No Filter) CSV Quick Reports ▾

Actions Delete Mark As Active/Inactive Update Create Item Bundle

Code ▾	Description	Category	Price Excl.	Price Incl.	Avg Cost	Last Cost	Qty On Hand	Active	Actions
<input type="checkbox"/> LAB	Labour / Call Out		R 0.00	R 0.00	R 0.00	R 0.00	R 0.00	<input checked="" type="checkbox"/>	<a href="#">Actions ▾</a>

First **1** Last Display 1 - 1 of 1

Copyright © 2025 Accounting, powered by Sage.  
[Terms & Conditions](#) | [Add - Ons and Downloads](#) | [Contact Us](#) | [Need Help?](#) | [Feedback](#)

OLD Line BASED POS but More Modern and Sage “Themed”

Pastel Point of Sale V10.1.0 (User - User 0) - [Pastel Point of Sale]

File Process View Setup Help

Quotation Order Sales Returns

Document Number IN100024  
 Account Number CASH  
 CASH SALES ACCOUNT

Total Due	Tendered
0.00	0.00

Place on Hold Cash on Delivery Order Number CAS

Code Quantity S/N Description Exclusive Price Discount

1.00 0.00 0.00 0.00

Delete Line <Ctrl D> Insert Line <Ctrl I> Total Tax 0.00

<F2> Cash <F3> Cheque <F4> Card <F7> EFT <F8> Voucher <F9> On Account

10 December 2008 - 16:38

<F10> Cash In/Out <F11> Cash up <F12> Reprint <SPACE> Next Edit Update Link Close

## Invoicing Style of Sage Accounting

**Sales Invoice: SI-14 PAID**

To: Combidesk

Invoice Address	Delivery Address	Invoice Date	Due Date	Reference
Whatever Street 4 Los Angeles 90001	Whatever Street 4 Los Angeles 90001	Oct 07, 2019	Nov 06, 2019	1021

Description	Qty/Hrs	Price/Rate	Discount	Tax	Subtotal
Short Sleeve T-Shirt Sales/Fees Discounts (4900)	1.00	49.00	0.00 0.00%	4.66 90001 Los Angeles 9.50%	49.00
				Subtotal	\$49.00
				Tax Breakdown	
				90001 Los Angeles: 49.00 @ 9.50%	4.66
				Total (1 item)	<b>\$53.66</b>

We'd love your feedback

English (US) © The Sage Group plc 2019

CREATED SENT VIEWED PAID  
AMOUNT PAID \$53.66 AMOUNT OUTSTANDING \$0.00  
1 payment on Oct 07, 2019

Email Print  
Edit Share Link  
Copy Add credit note  
Download Print packing slip  
Void

## AI Generated IMAGES:

**POS**

Order	Order	Sale	Returns
Document Number 100024	R 0,0	R 0,0	R 0,00

Place on Hold Cash on Delivery Cash on Hold

Category	Code	Description	Opening Stock	PurPrice	SalesPrice	Amount
Accessories	A0001	Brown Velvet Bowtie	5	R10.00	R10.0	R 30.0
Accessories	A0002	Green Velvet Bowtie	5	R10.00	R 20.0	R 400.0
Accessories	A0003	Silk Patterned Scarf	10	R30.00	R8.00	R 400.0
Accessories	A0004	Wool Men's Cap	5	R15.00	R 80.0	R 400.0
Accessories	A0005	Mixed Wool Gloves	15	R10.00	R 40.0	R 200.0
Accessories	A0006	Stik Gloves	3	R 5.00	R 80.0	R 200.0
Accessories	A0007	Striped Pink Scarf	9	R 15.00	R 30.0	R 1000.0
Accessories	A0010	Polka Dot Umbrella	10	R 10.00	R 80.0	R 400.0
Accessories	A0011	Black Umbrella	20	R20.00	R 60.0	R 300.0
Accessories	A0012	Classic Leather Beit	13	R 10.00	R 60.0	R 600.0
Accessories	A0012	Classic Leather Beit	14	R 15.00	R 30.0	R 1200.0

<F1> Item Search <F2> Pay <F3> Remove item Total Excl. R 0.00 Total Tax R 0,00

# POS

- [Dashboard](#)
- [Sales](#)
- [Products](#)

## Point of Sale

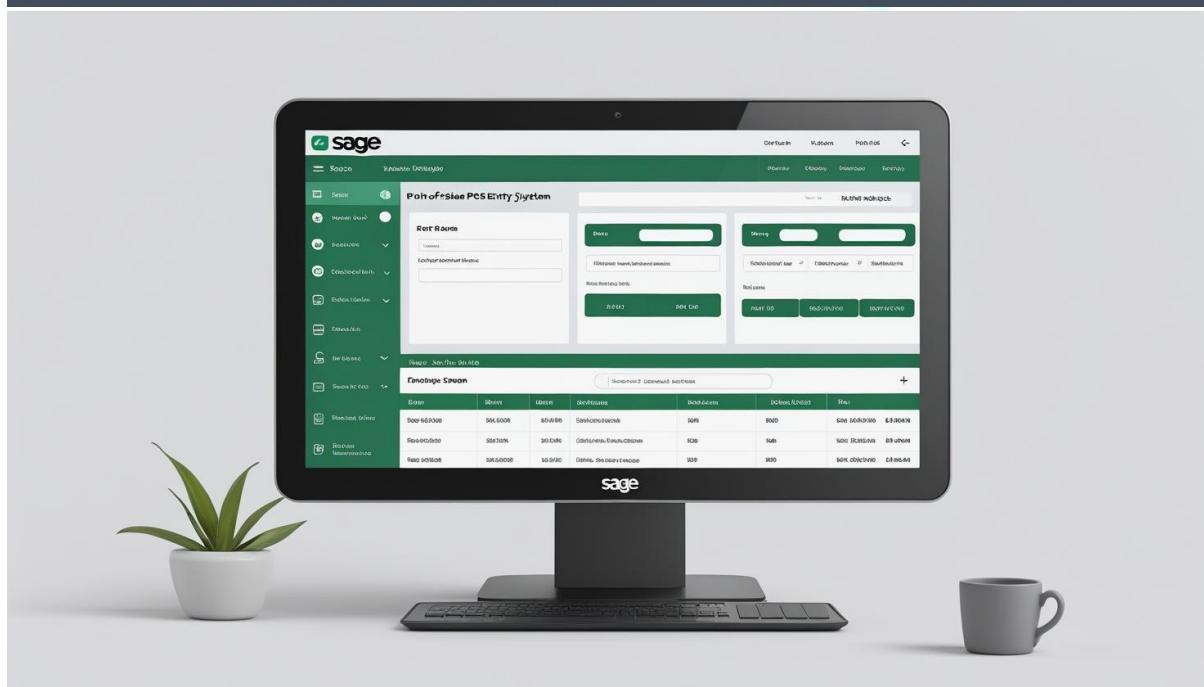
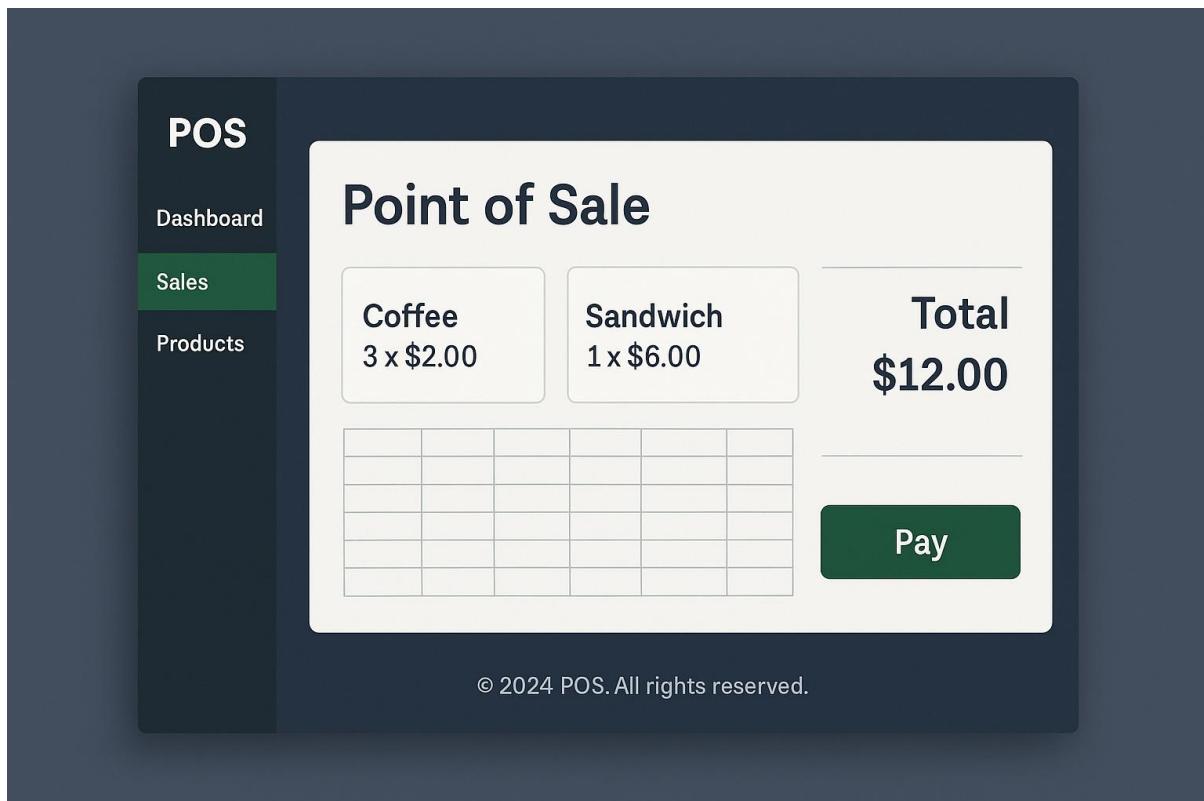
Account Number CAS  
Cash Sales Account

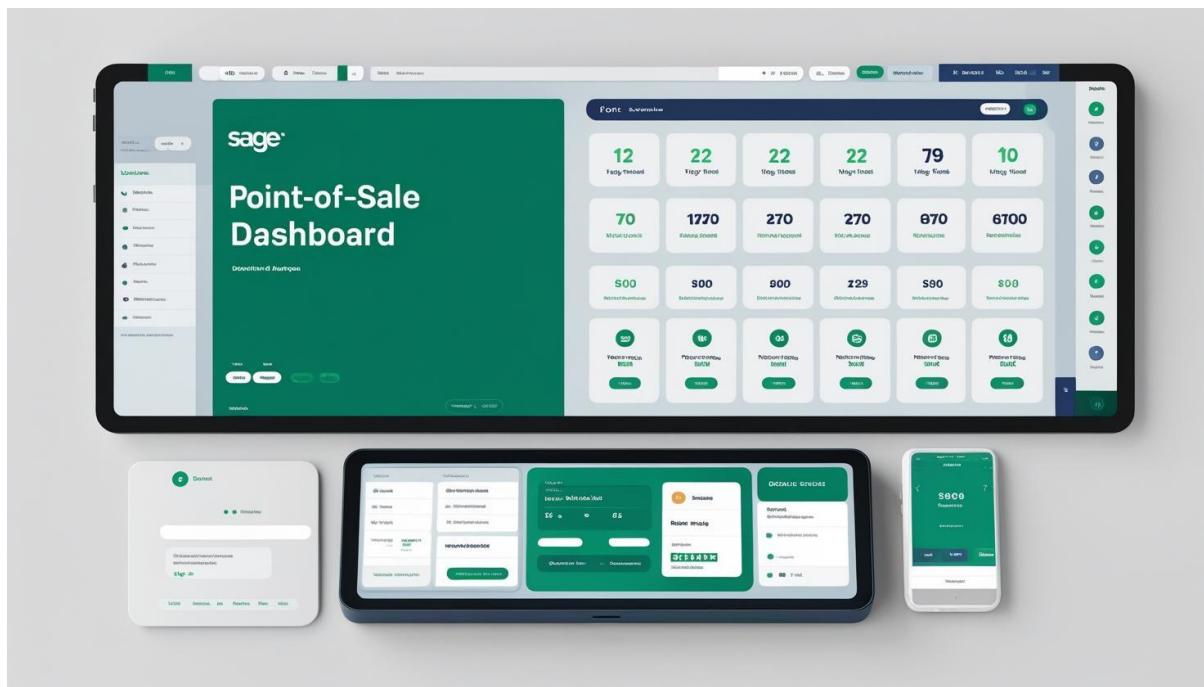
Total Due	\$8.00	Tendered
\$8.00		

Code	Quantity	Description	Exclusive Price	Total
10000	1.00	Coffee	\$2.00	\$2.00
10001	1.00	Sandwich	\$6.00	\$6.00
			Total	\$8.00

Cash
Cheque
Card
EFT
On Account

© 2024 POS. All rights reserved.





# CloudX

## Sage Theme

### Typography

- **Primary Font:** Lato  
Clean and modern — consistent with Sage's design language.

### Menu Text

- Font Size: 12px
- Font Weight: 700
- Line Height: 132px
- Text Alignment: Left

### Page Titles

- Font Size: 24px
- Font Weight: 700

### Subtitles

- Font Size: 22px
- Line Height: 79px

### Buttons

- Font Size: 14px

### Body Text / General Content

- Font Size: 12px

### Footer Text

- Font Size: 11px

---

### Layout & Spacing

- **Header Height:** 40px
- **Section Headings:** 80px height
- **Page Padding:** 80px on all sides (Top, Bottom, Left, Right)
- **Grid System:** Centered 2x2 grid layout for content
- **POS Screen:** Must be centered within the page, with the same 80px padding all around to match general layout rules

---

### Color Palette



Purpose	Color	Hex Code
<b>Primary Green</b>	Green	#2a843f
<b>Primary Blue</b>	Blue	#004089
<b>Text / Surface</b>	White	#ebedef
<b>Background</b>	Wrapper BG	#3c424f

---

This theme guide must be applied consistently across all pages and components, including the POS screen, login view, dashboards, modals, and dropdowns — ensuring we stay close to the **Sage design system** and deliver a familiar experience to our end users.

Let me know if you'd like this converted into a Figma reference board or shared design spec.



## References

### Sage Application Integration Reference

#### Access Link:

 [Sage Reseller Portal](#)

*Access is granted upon acceptance.*

---

### API Access

#### API Documentation & Limits:

 [SageOne Developer Program](#)

#### All Available API Calls:

 [SageOne Accounting API v2.0.0](#)

---

### Product Inspiration & Design References

Below are examples of POS systems we found useful for UX/UI and feature inspiration:

-  [InfyPOS](#) – Clean layout, modular sales workflow
-  [ReadyPOS](#) – POS-focused with dashboard and billing logic

## API's We Would Need (API Calls may differ from documentation)

### Company & Authentication

- **Company/Get** – Retrieve list of companies the user can access
  - **Company/Get/{id}** – Retrieve details for a specific company
  - *Required for multi-company login & selection*
- 

### Product & Inventory Management

- **Products/Get** – Retrieve all products/services
  - **Products/Get/{id}** – Fetch details for a specific product
  - **StockMovements/Get** – Track item in/out movements
  - **StockItems/Get** – Get stock-specific item details
  - *Needed for inline search, stock levels, inventory sync*
- 

### Sales & Transactions

- **SalesInvoices/Get, /Get/{id}** – Retrieve invoices
  - **SalesInvoices/Save** – Create/update a sales invoice
  - **SalesCreditNotes/Get, /Get/{id}** – Fetch credit notes
  - **SalesCreditNotes/Save** – Issue a credit note (returns)
  - **SalesSummary** – Get sales summary reports
  - *Supports transaction posting, returns, dockets, analytics*
- 

### Payments & Refunds

- **Payments/Get, /Get/{id}** – List or fetch payments
  - **Payments/Save** – Create payment records (captures)
  - **Receipts/Get, /Get/{id}** – Retrieve receipts
  - **Receipts/Save** – Register receipts (refunds)
  - *Used for refunds, payout tracking, loyalty credit*
- 

### Customers & CRM

- **Contacts/Get, /Get/{id}** – Retrieve customer/contact data
-

- **Contacts/Save** – Create or update contact profiles
  - *Used for customer lookup, loyalty profiles, returns*
- 

## **Warehouse & Stock**

- **StockItems/Adjust** – Adjust inventory (used for returns/pick slips)
  - **StockMovements/Get** – Retrieve stock transaction history
  - *Essential for warehouse-level stock updates*
- 

## **Gift & Coupon Handling (via Contacts or Custom Objects)**

- Use **Contacts/** endpoints or custom fields (if supported)
  - Alternatively leverage a separate coupon/gift-card API module in Sage
  - *For issuing, redeeming, and tracking incentives*
- 

## **App Settings & Syncing**

- **Settings/Get** – Retrieve business-level settings
  - **Currencies/Get, TaxRates/Get, InvoiceSettings/Get** – Pull config data
  - *Needed for initial setup, tax rules, currency formats*
- 

## **Summary Table**

<b>Feature</b>	<b>Endpoint Group</b>
Company Login & Selection	Company/Get, Company/Get/{id}
Product Search	Products/Get
Stock Levels	StockItems/Get, StockMovements/Get
Sales Transactions	SalesInvoices, SalesCreditNotes
Payments & Refunds	Payments, Receipts
Customer Profiles	Contacts
Inventory Adjustments	StockItems/Adjust
Settings & Tax Rules	Settings, Currencies, TaxRates