

Claire Streb Portfolio 2015 – Present


Contents

- LogMeIn API Redocumentation Project – 2019..... 2
- Business Intelligence Data Analytics – 2017 3
 - Proof of Concept for KPI Dashboards 3
 - Final Design of an Actual KPI Dashboard 4
- Mobile Inventory – 2017 – Project was still in-progress when I left 5
 - Wireframe (UI Design) for taking inventory by scanning bar codes on smart phones and entering counts 5
- Menu Display – 2016 – Complete, Operational, and Generating Revenue..... 6
 - Snapshots of SOAP API Documentation Generated with Sandcastle and C# 6
 - Sales/Marketing Information..... 8
- API and SDK Documentation – 2015 – Anonymized Project 10

LogMeIn API Redocumentation Project – 2019

This project is being done in phases: 1) Most urgent additions and rewriting; 2) Fix typos and grammar; and 3) Reorganize most of the existing outline.

- Live URL (LogMeIn bought Nanorep in 2017) – Currently in phase 1 as of July 2019
<https://support.nanorep.com/API-Integrations/General-API/1009692792/Bold360ai-API-overview.htm>
- Snapshot of completed REST API Documentation Generated with Postman, Markdown, and JSON (DELETE, GET, POST, and PUT expand for more information) whose styling will be used on this project as can be seen at <https://goto-developer.logmeininc.com/content/gotomeeting-api-reference#/>

Developer Center APIs ▾ Resources ▾ Support ▾ API Status My Apps Sign in 

Organizers

Operations on organizers. Only available to users with the admin role.

Show/Hide | List Operations | Expand Operations

DELETE	/organizers	Delete organizer by email
GET	/organizers	Get organizer by email / Get all organizers
POST	/organizers	Create organizer
DELETE	/organizers/{organizerKey}	Delete organizer
GET	/organizers/{organizerKey}	Get organizer
PUT	/organizers/{organizerKey}	Update organizer
GET	/organizers/{organizerKey}/attendees	Get attendees by organizer
GET	/organizers/{organizerKey}/historicalMeetings	Get historical meetings by organizer
GET	/organizers/{organizerKey}/meetings	DEPRECATED: Get meetings by organizer

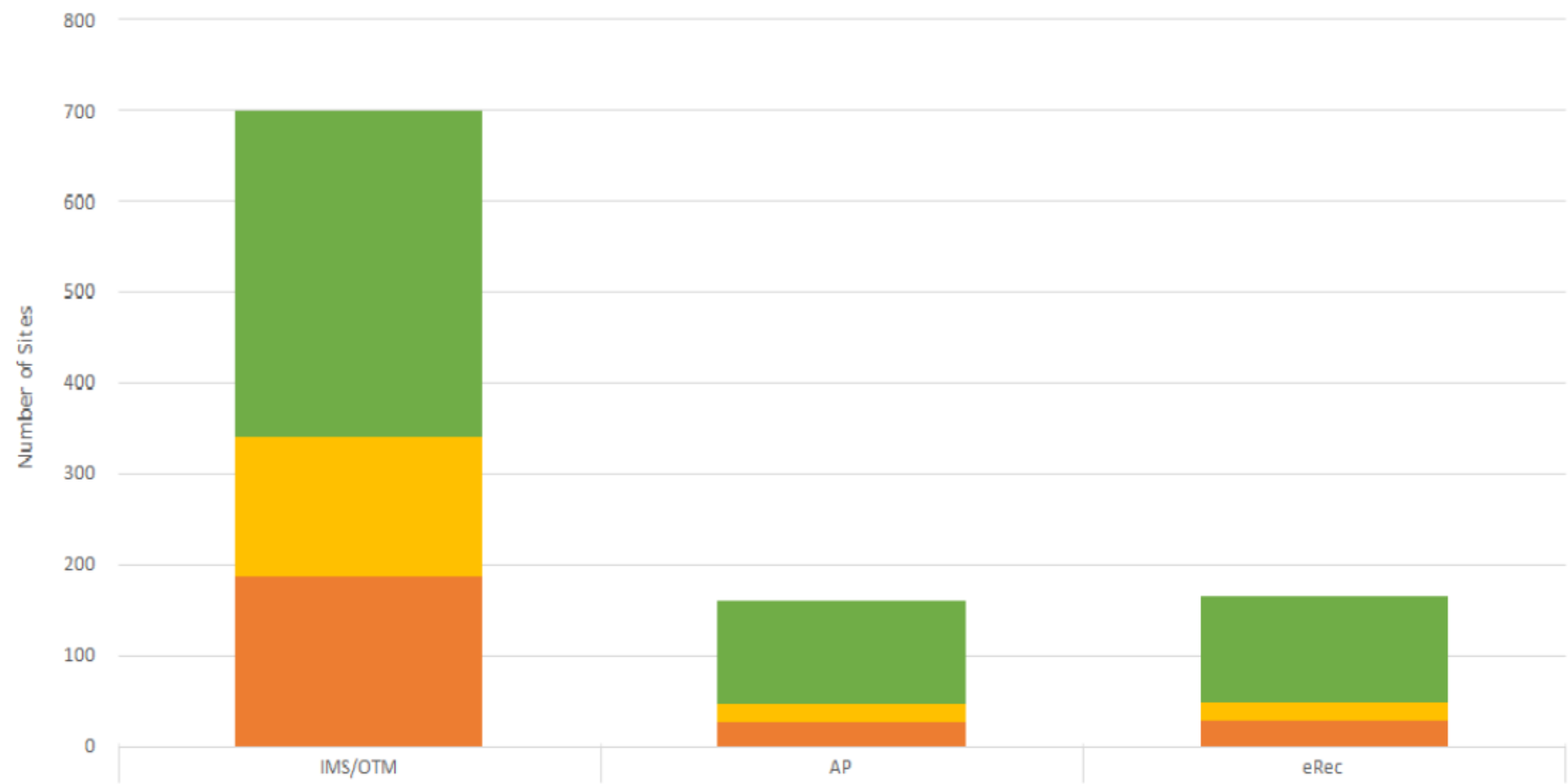
Business Intelligence Data Analytics – 2017

Brought Qlik Sense into the company to utilize its first Data Visualization software

Proof of Concept for KPI Dashboards



Interfaces Comparison, On-Premise



Mobile Inventory – 2017 – Project was still in-progress when I left


Wireframe (UI Design) for taking inventory by scanning bar codes on smart phones and entering counts

Native Phone Header

Setting

Title


Search



Name

Online / Leave


Information



Tally

status


Information



Tally

status


Information



Tally

status


Information



Tally

status

Information



Tally

status

Information

Update

Refresh

Add

Next

Inventory Control Sheet

Food Item

Beans

Purchase Unit (bag, can, etc.)


Number of Units on Hand

CBORD.NetMenu.Framework.CBORDWebData Namespace

WebService Namespace: <http://cbord.com/webData/>

CBORD Menu Display Interface (MDI) is a web service API that retrieves data from a NetMenu or FMS database for display on digital menu boards.

▲ Classes

Class	Description
 CBORDWebData	<p>CBORD Web Data for MDI Version 2, a sessionless Web Service (SOAP 1.1 and 1.2).</p> <p>MDI provides access to business data and processes defined by NetMenu or FMS. Access is accomplished by calling each web method with identifying parameters. Each web method except for VersionInfo returns a string containing an XML document that has its own schema.</p> <p>As of MDI Version 2, the encryptedUserName/encryptedUserPassword credentials are assigned to real users authorized to access Facilities in NetMenu or FMS. The best practice is to assign one MDI user to all of the Facilities in the enterprise. In the case of a CBORD-hosted cloud, one MDI user should be assigned to all of the Facilities for each tenant (enterprise).</p>

CBORDWebDataGetFacilities Method

Menu Display Method. Retrieves zero or more Facilities the user is allowed to access.

Namespace: [CBORD.NetMenu.Framework.CBORDWebData](#)

Assembly: CbordWebData (in CbordWebData.dll) Version: 12.14.100.8489

▲ Syntax

C# VB C++ F#

[Copy](#)

```
public string GetFacilities(  
    string encryptedUserName,  
    string encryptedPassword,  
    string facilityExternalId,  
    bool getSingleFacility  
)
```

Parameters

encryptedUserName

Type: [SystemString](#)

Contact the MDI Administrator for the encryptedUserName.

encryptedPassword

Type: [SystemString](#)

Contact the MDI Administrator for the encryptedPassword



Menu Display Interface™

Benefits

- Real-time data for your dynamic menu displays
- Use your menu data to highlight and encourage healthy meal choices
- Include food allergen information with menu items
- Stop spending hours posting nutrition information

Overview

Take the time and hassle out of updating dynamic menu displays. With our Menu Display Interface™, you can utilize the allergen and nutrition information you already have in Foodservice Suite® (FSS) to drive your electronic menu displays. When you connect your menu boards directly



Turn CBORD Data Into a Dynamic Menu Board Display

Advertise Your Healthy Meal Options

The data feeds you create with the Menu Display Interface will drive real-time menu data to your display system of choice. The CBORD® Menu Display Interface allows you to feed your FSS data to your presentation system with a standard web-based API. Using the Menu Display Interface Administration site, your menu and marketing teams can coordinate the selection of specific menus or menu items to be displayed on your menu boards. The site requires menu board system interaction with web services (SOAP/XML). No more direct database queries, file exports, network administration and permissions headaches. Set up your menu boards to communicate directly with FSS to present your menu data in a dynamic fashion, changing real-time as your menus change.

Pizza & Deli						= heart healthy = vegetarian = vegan							
		price	cal	fat	sod	carb		price	cal	fat	sod	carb	
Pizza	Design Your Own Personal Pizza						Sandwiches	Deli sandwich made to order					
	Cheese	3.99	590	24	1,320	68		Ham	3.99	220	8	1,430	31
	Vegetable	4.49	190	5	480	28		Turkey	3.99	280	7	890	30
	Meat	4.99	250	12	630	26		Roast Beef	3.99	250	5	1,340	31
Pizza	By The Slice	3.25	650	39	1,430	65	Sides	House Chips	0.99	160	160	90	16
	Pepperoni		340	11	950	43		Pasta Salad	1.99	330	2	790	28
Whole Pie	Cheese	15.00	330	11	920	43	Ala Carte	Potato Salad	1.99	200	14	430	17
Sub Special	Each Additional Topping	.50						Deli Combo	6.99				
	Chicken Parmesan	4.59	540	22	1,500	65	Specials	Calzone	5.99	641	26	1,703	70
								Stromboli	5.99	890	899	1,809	82

© 2016 Epicure Digital Menu Systems

The CBORD Group, Inc.
61 Brown Road, Ithaca, NY 14850
1.844.GO.CBORD • FAX: 607.257.1902
www.cbord.com

XYZ SDK 3Q2015 for JavaScript

The XYZ SDK for JavaScript (hereafter called "the XYZ SDK") is a library with powerful features that enable you, the JavaScript developer, to easily make requests to the XYZ API. The XYZ SDK makes it simple to upload media and send batch requests to your XYZ installed database. The XYZ SDK gives you full control of how your JavaScript code interacts with your specific hosting environment and web framework. The XYZ SDK does all the heavy lifting for you making it as easy as possible to deeply integrate into your installed XYZ platform.

Pre-Requisites

You or your system administrator must have already followed the [XYZ Installation and Implementation Guide](#).

Examples

The following examples demonstrate how you can accomplish tasks with the XYZ SDK:

- [Login](#) – Request XYZ authentication
- [Retrieve](#) – Retrieve one or more XYZ records
- [Upload](#) – Send one or more media files to XYZ

API Reference

For a full list of classes, please see the [API Reference](#).

SDK Release History

For a complete list of releases, please see the [SDK Release History](#).