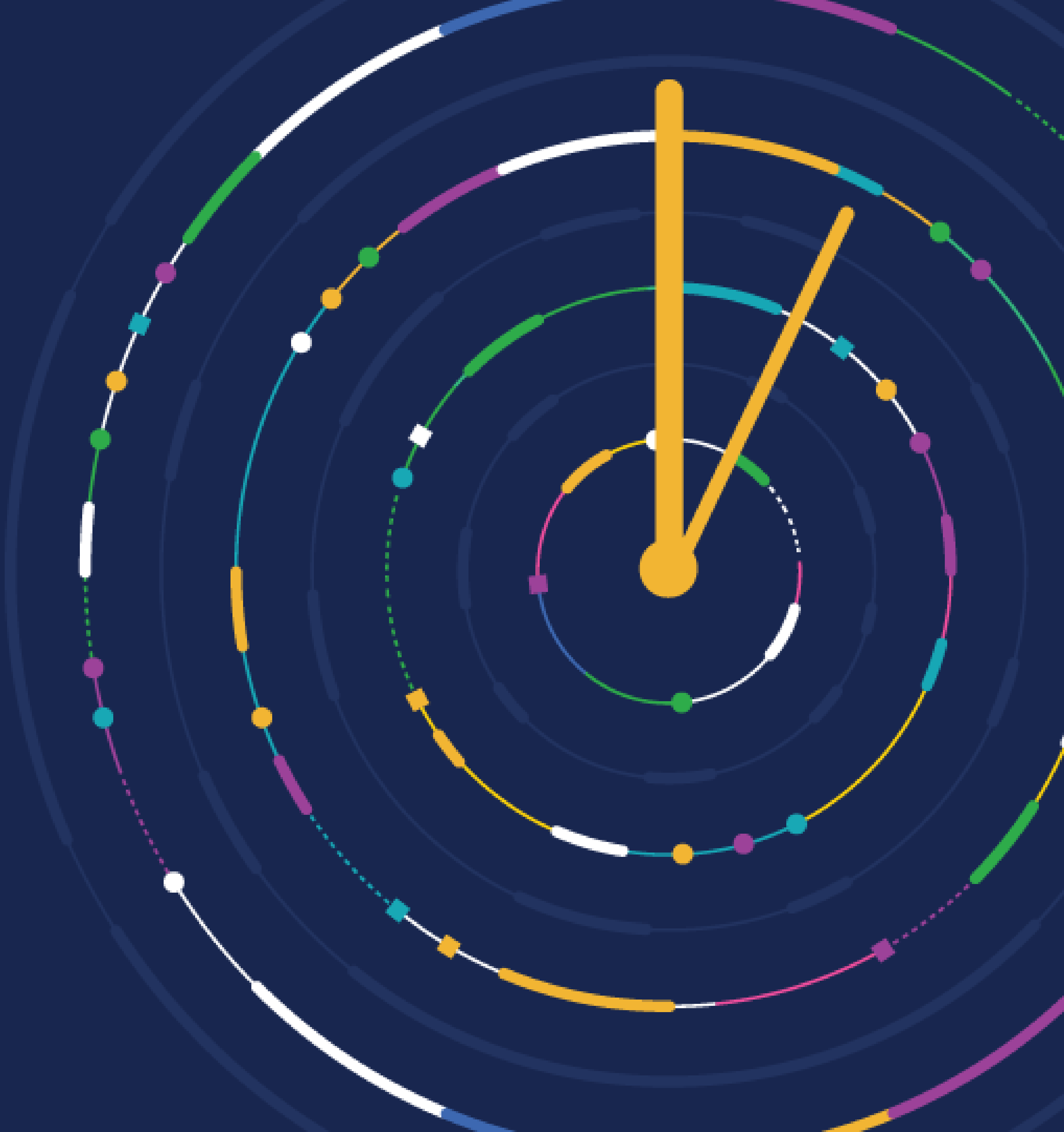
tecman

DIY Integration! Learn how to make tools & systems work together by integrating them via APIs! Save time & increase your productivity by extracting the data you need from BC.

Tomasz Fica, Marc Tweedie & Dhruv Patel

CUSTOMER DAY 2023



Timings

9:30am - DIY INTEGRATION

10:30am - Coffee Break

11:00am - DIY INTEGRATION

12:00pm - Lunch

1:00pm - DIY INTEGRATION

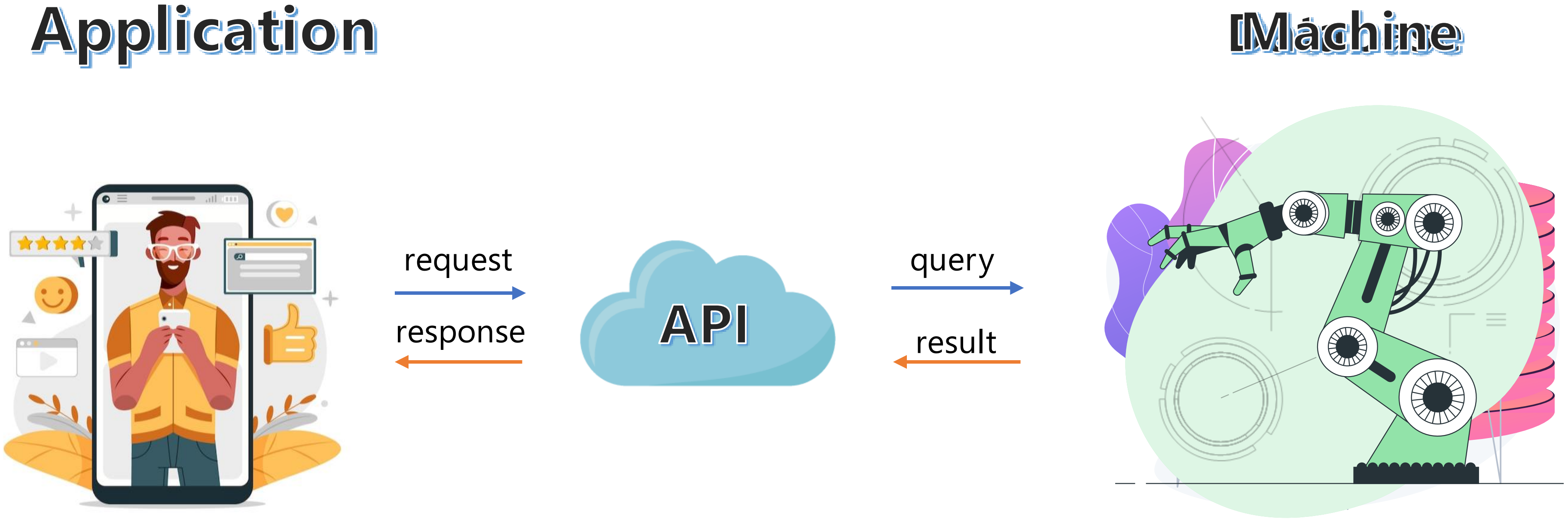
2:00pm - Coffee Break

2:30pm - Sessions

3:00pm - Close

What is API?

Secure Communication



Application

request

response

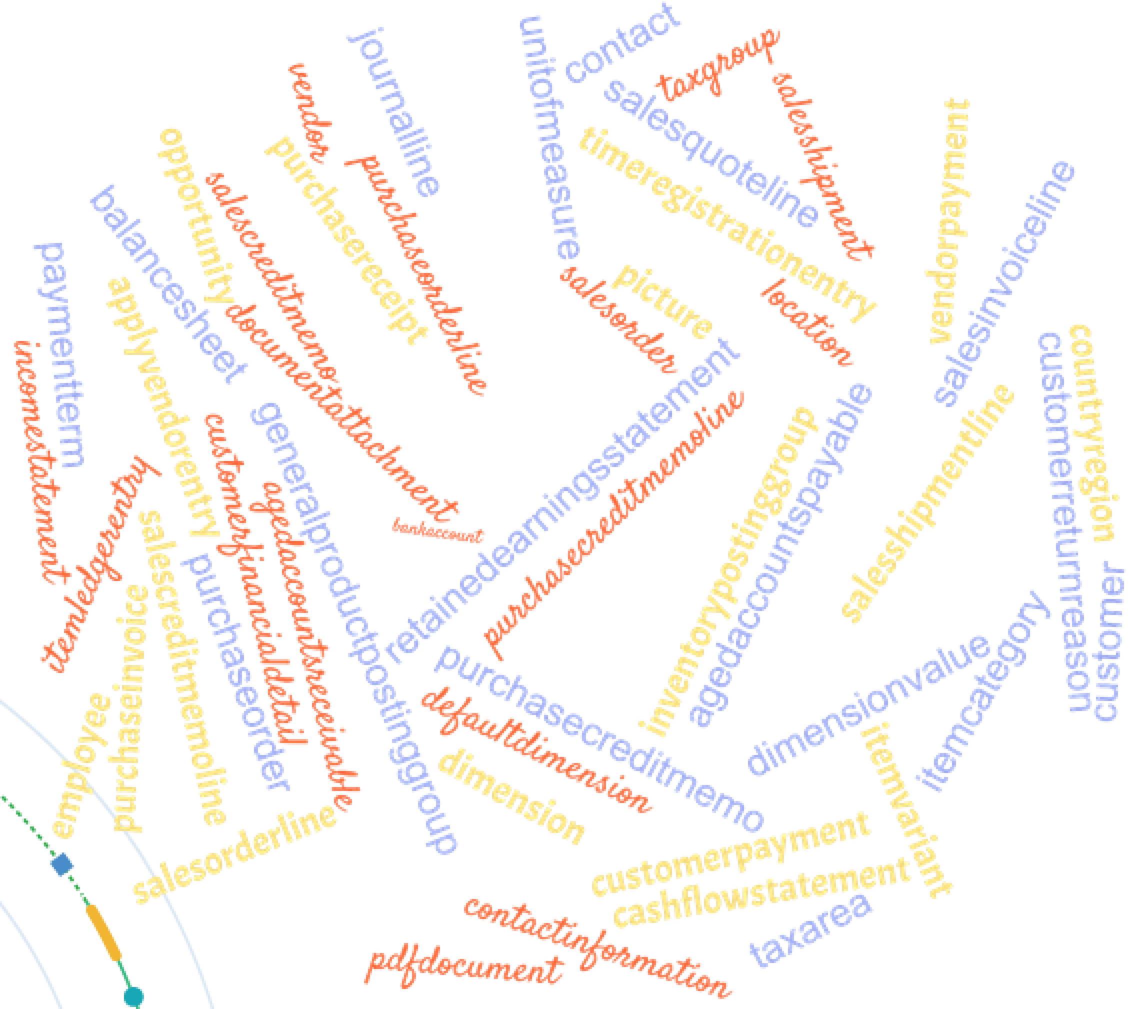
API

query

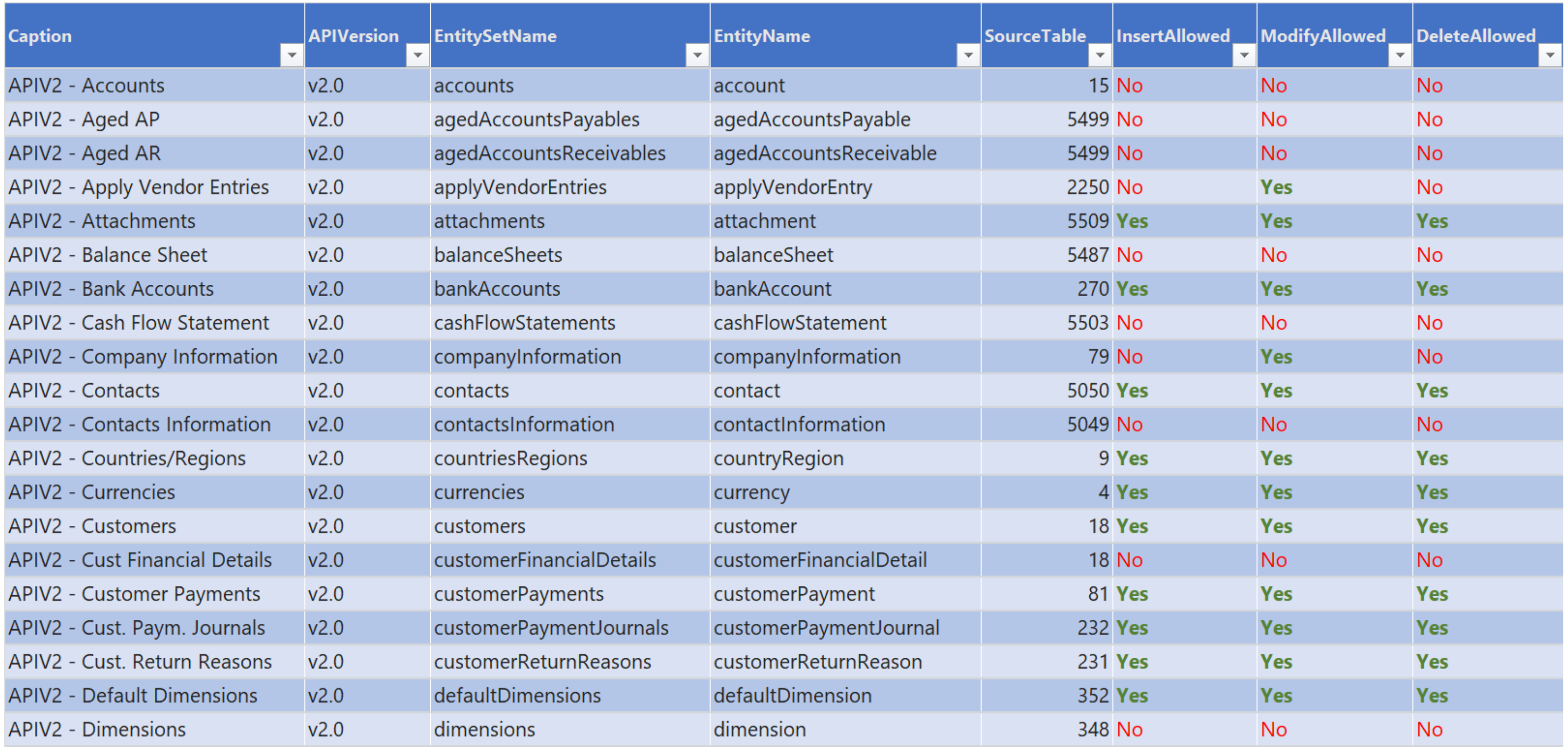
result

Machine

„Standard" APIs

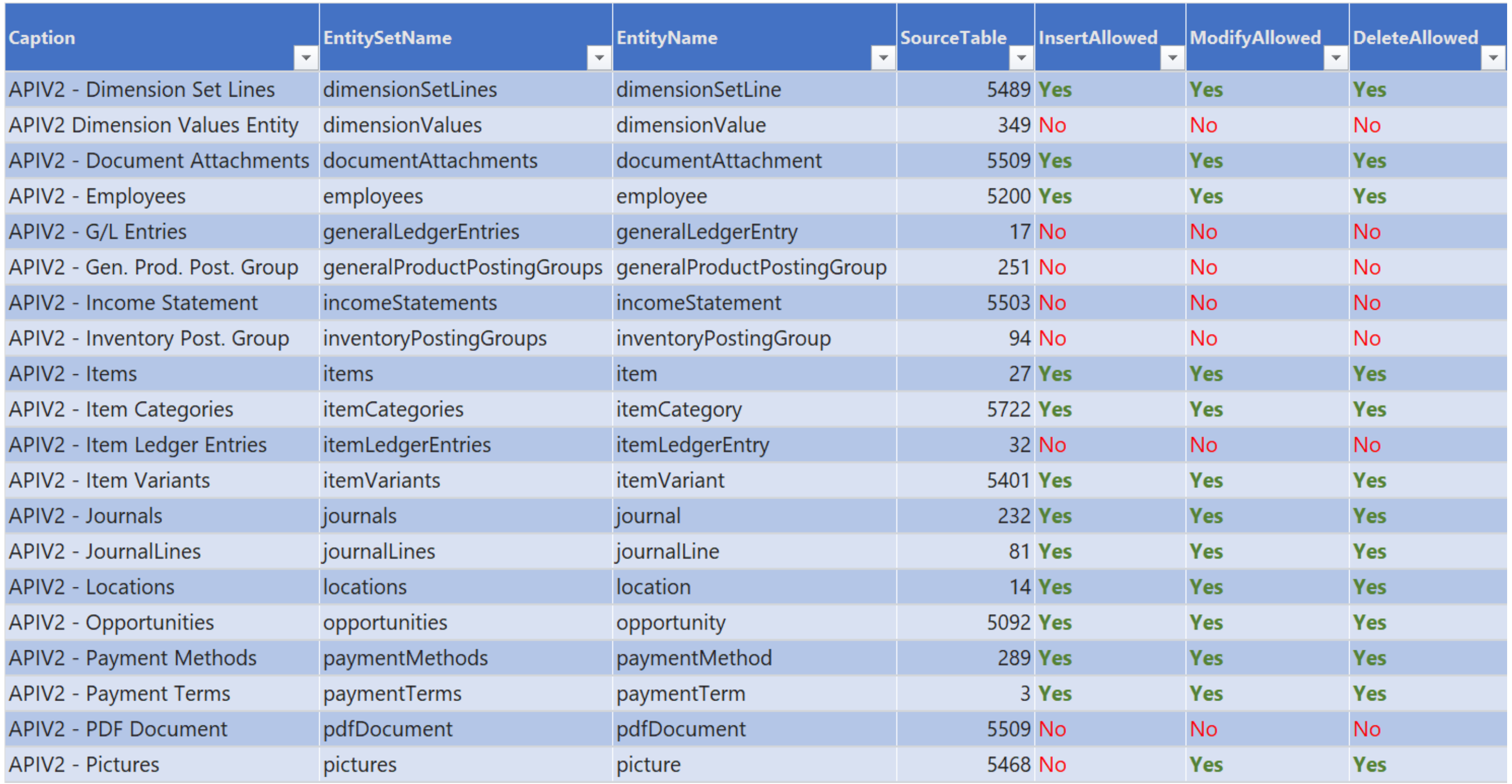


„Standard“ APIs



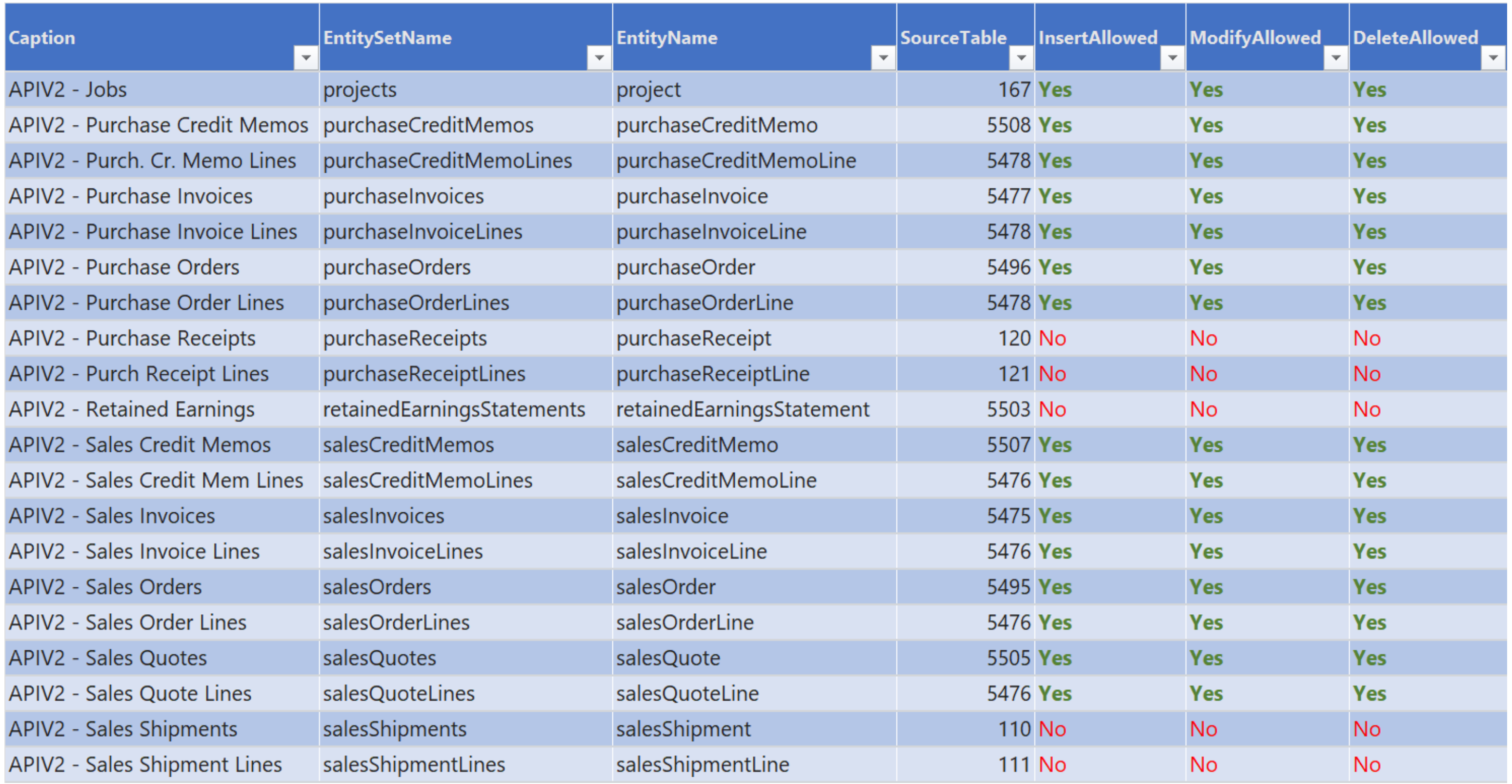
Caption	APIVersion	EntitySetName	EntityName	SourceTable	InsertAllowed	ModifyAllowed	DeleteAllowed
APIV2 - Accounts	v2.0	accounts	account	15	No	No	No
APIV2 - Aged AP	v2.0	agedAccountsPayables	agedAccountsPayable	5499	No	No	No
APIV2 - Aged AR	v2.0	agedAccountsReceivables	agedAccountsReceivable	5499	No	No	No
APIV2 - Apply Vendor Entries	v2.0	applyVendorEntries	applyVendorEntry	2250	No	Yes	No
APIV2 - Attachments	v2.0	attachments	attachment	5509	Yes	Yes	Yes
APIV2 - Balance Sheet	v2.0	balanceSheets	balanceSheet	5487	No	No	No
APIV2 - Bank Accounts	v2.0	bankAccounts	bankAccount	270	Yes	Yes	Yes
APIV2 - Cash Flow Statement	v2.0	cashFlowStatements	cashFlowStatement	5503	No	No	No
APIV2 - Company Information	v2.0	companyInformation	companyInformation	79	No	Yes	No
APIV2 - Contacts	v2.0	contacts	contact	5050	Yes	Yes	Yes
APIV2 - Contacts Information	v2.0	contactsInformation	contactInformation	5049	No	No	No
APIV2 - Countries/Regions	v2.0	countriesRegions	countryRegion	9	Yes	Yes	Yes
APIV2 - Currencies	v2.0	currencies	currency	4	Yes	Yes	Yes
APIV2 - Customers	v2.0	customers	customer	18	Yes	Yes	Yes
APIV2 - Cust Financial Details	v2.0	customerFinancialDetails	customerFinancialDetail	18	No	No	No
APIV2 - Customer Payments	v2.0	customerPayments	customerPayment	81	Yes	Yes	Yes
APIV2 - Cust. Paym. Journals	v2.0	customerPaymentJournals	customerPaymentJournal	232	Yes	Yes	Yes
APIV2 - Cust. Return Reasons	v2.0	customerReturnReasons	customerReturnReason	231	Yes	Yes	Yes
APIV2 - Default Dimensions	v2.0	defaultDimensions	defaultDimension	352	Yes	Yes	Yes
APIV2 - Dimensions	v2.0	dimensions	dimension	348	No	No	No

„Standard” APIs



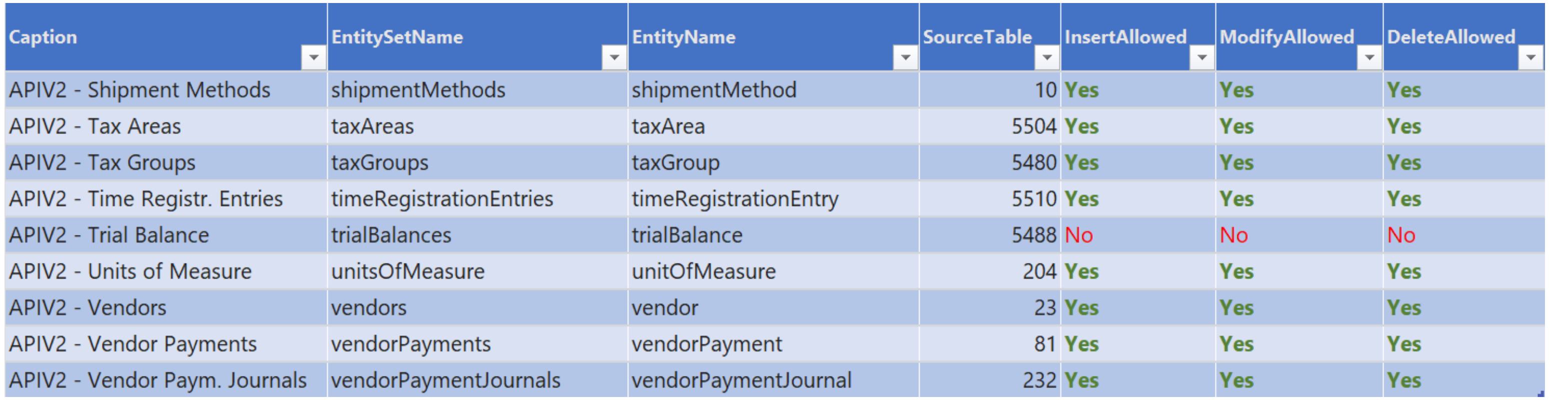
Caption	EntitySetName	EntityName	SourceTable	InsertAllowed	ModifyAllowed	DeleteAllowed
APIV2 - Dimension Set Lines	dimensionSetLines	dimensionSetLine	5489	Yes	Yes	Yes
APIV2 Dimension Values Entity	dimensionValues	dimensionValue	349	No	No	No
APIV2 - Document Attachments	documentAttachments	documentAttachment	5509	Yes	Yes	Yes
APIV2 - Employees	employees	employee	5200	Yes	Yes	Yes
APIV2 - G/L Entries	generalLedgerEntries	generalLedgerEntry	17	No	No	No
APIV2 - Gen. Prod. Post. Group	generalProductPostingGroups	generalProductPostingGroup	251	No	No	No
APIV2 - Income Statement	incomeStatements	incomeStatement	5503	No	No	No
APIV2 - Inventory Post. Group	inventoryPostingGroups	inventoryPostingGroup	94	No	No	No
APIV2 - Items	items	item	27	Yes	Yes	Yes
APIV2 - Item Categories	itemCategories	itemCategory	5722	Yes	Yes	Yes
APIV2 - Item Ledger Entries	itemLedgerEntries	itemLedgerEntry	32	No	No	No
APIV2 - Item Variants	itemVariants	itemVariant	5401	Yes	Yes	Yes
APIV2 - Journals	journals	journal	232	Yes	Yes	Yes
APIV2 - JournalLines	journalLines	journalLine	81	Yes	Yes	Yes
APIV2 - Locations	locations	location	14	Yes	Yes	Yes
APIV2 - Opportunities	opportunities	opportunity	5092	Yes	Yes	Yes
APIV2 - Payment Methods	paymentMethods	paymentMethod	289	Yes	Yes	Yes
APIV2 - Payment Terms	paymentTerms	paymentTerm	3	Yes	Yes	Yes
APIV2 - PDF Document	pdfDocument	pdfDocument	5509	No	No	No
APIV2 - Pictures	pictures	picture	5468	No	Yes	Yes

„Standard" APIs



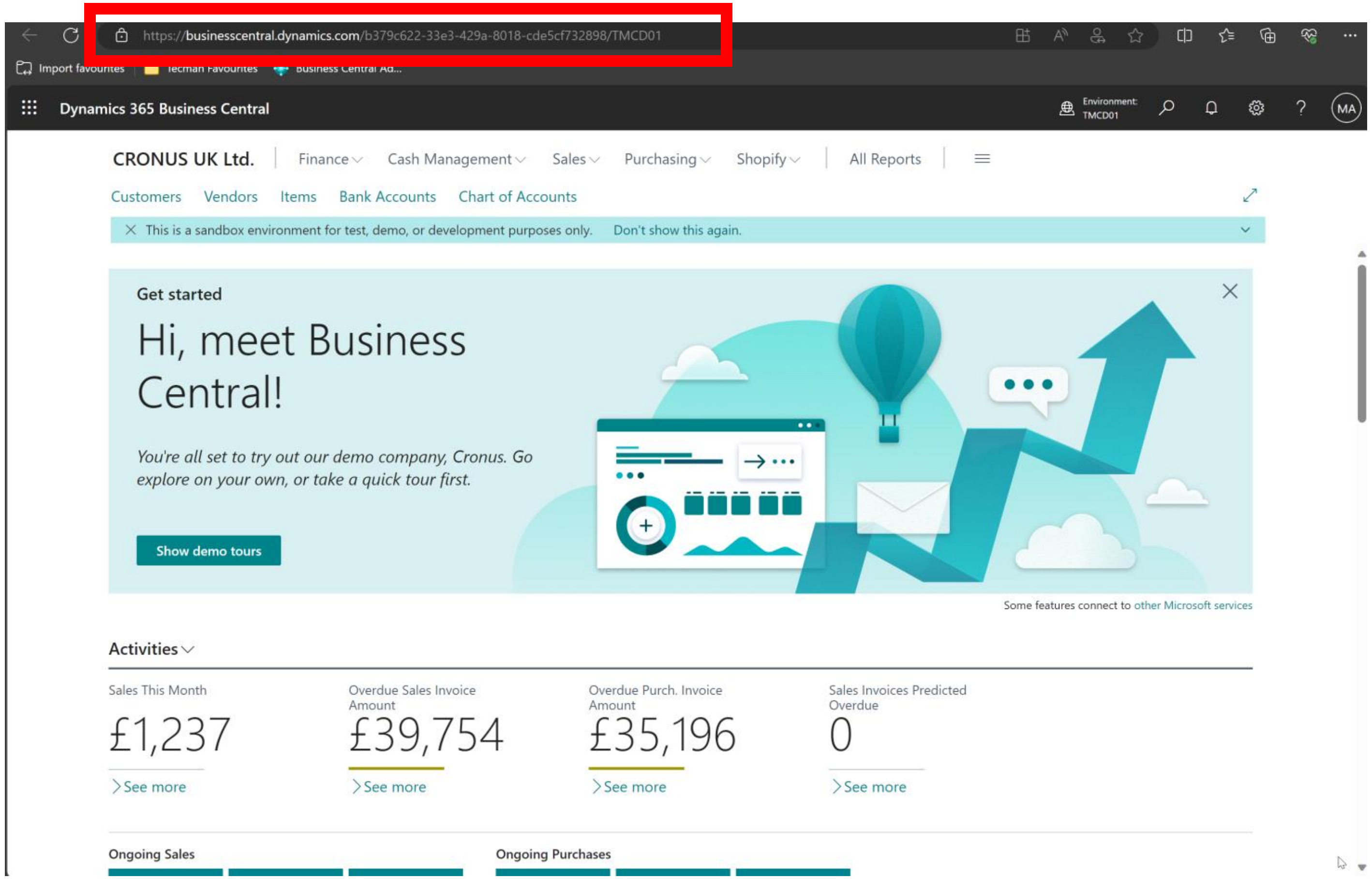
Caption	EntitySetName	EntityName	SourceTable	InsertAllowed	ModifyAllowed	DeleteAllowed
APIV2 - Jobs	projects	project	167	Yes	Yes	Yes
APIV2 - Purchase Credit Memos	purchaseCreditMemos	purchaseCreditMemo	5508	Yes	Yes	Yes
APIV2 - Purch. Cr. Memo Lines	purchaseCreditMemoLines	purchaseCreditMemoLine	5478	Yes	Yes	Yes
APIV2 - Purchase Invoices	purchaseInvoices	purchaseInvoice	5477	Yes	Yes	Yes
APIV2 - Purchase Invoice Lines	purchaseInvoiceLines	purchaseInvoiceLine	5478	Yes	Yes	Yes
APIV2 - Purchase Orders	purchaseOrders	purchaseOrder	5496	Yes	Yes	Yes
APIV2 - Purchase Order Lines	purchaseOrderLines	purchaseOrderLine	5478	Yes	Yes	Yes
APIV2 - Purchase Receipts	purchaseReceipts	purchaseReceipt	120	No	No	No
APIV2 - Purch Receipt Lines	purchaseReceiptLines	purchaseReceiptLine	121	No	No	No
APIV2 - Retained Earnings	retainedEarningsStatements	retainedEarningsStatement	5503	No	No	No
APIV2 - Sales Credit Memos	salesCreditMemos	salesCreditMemo	5507	Yes	Yes	Yes
APIV2 - Sales Credit Mem Lines	salesCreditMemoLines	salesCreditMemoLine	5476	Yes	Yes	Yes
APIV2 - Sales Invoices	salesInvoices	salesInvoice	5475	Yes	Yes	Yes
APIV2 - Sales Invoice Lines	salesInvoiceLines	salesInvoiceLine	5476	Yes	Yes	Yes
APIV2 - Sales Orders	salesOrders	salesOrder	5495	Yes	Yes	Yes
APIV2 - Sales Order Lines	salesOrderLines	salesOrderLine	5476	Yes	Yes	Yes
APIV2 - Sales Quotes	salesQuotes	salesQuote	5505	Yes	Yes	Yes
APIV2 - Sales Quote Lines	salesQuoteLines	salesQuoteLine	5476	Yes	Yes	Yes
APIV2 - Sales Shipments	salesShipments	salesShipment	110	No	No	No
APIV2 - Sales Shipment Lines	salesShipmentLines	salesShipmentLine	111	No	No	No

„Standard" APIs



Caption	EntitySetName	EntityName	SourceTable	InsertAllowed	ModifyAllowed	DeleteAllowed
APIV2 - Shipment Methods	shipmentMethods	shipmentMethod	10	Yes	Yes	Yes
APIV2 - Tax Areas	taxAreas	taxArea	5504	Yes	Yes	Yes
APIV2 - Tax Groups	taxGroups	taxGroup	5480	Yes	Yes	Yes
APIV2 - Time Registr. Entries	timeRegistrationEntries	timeRegistrationEntry	5510	Yes	Yes	Yes
APIV2 - Trial Balance	trialBalances	trialBalance	5488	No	No	No
APIV2 - Units of Measure	unitsOfMeasure	unitOfMeasure	204	Yes	Yes	Yes
APIV2 - Vendors	vendors	vendor	23	Yes	Yes	Yes
APIV2 - Vendor Payments	vendorPayments	vendorPayment	81	Yes	Yes	Yes
APIV2 - Vendor Paym. Journals	vendorPaymentJournals	vendorPaymentJournal	232	Yes	Yes	Yes

URL to connect



https://businesscentral.dynamics.com/b379c622-33e3-429a-8018-cde5cf732898/TMCD01

This is a sandbox environment for test, demo, or development purposes only. Don't show this again.

Get started

Hi, meet Business Central!

You're all set to try out our demo company, Cronus. Go explore on your own, or take a quick tour first.

Some features connect to other Microsoft services

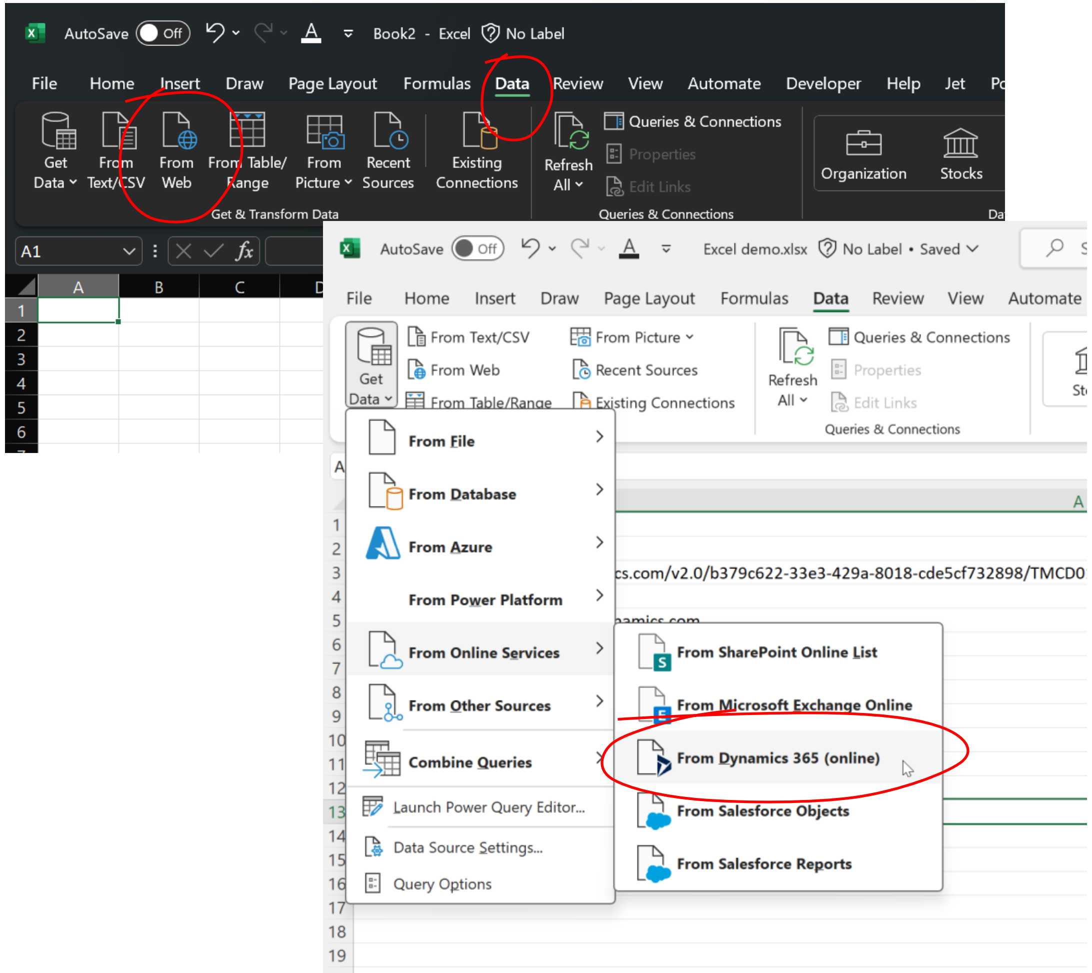
Sales This Month	Overdue Sales Invoice Amount	Overdue Purch. Invoice Amount	Sales Invoices Predicted Overdue
£1,237	£39,754	£35,196	0

URL to connect

https://api.businesscentral.dynamics.com/
v2.0/
<tenantId>/
<environmentName>/
api/v2.0/
companies(<companyId>)/
apiName

https://api.businesscentral.dynamics.com/
v2.0/
b379a6f2-33e3-429a-8018-cde5cf732898/
TMCD02/
api/v2.0/
companies(7368d553-0d5c-ee11-8df4-6045bdc89477)/
vendors

Demo – BC and Excel



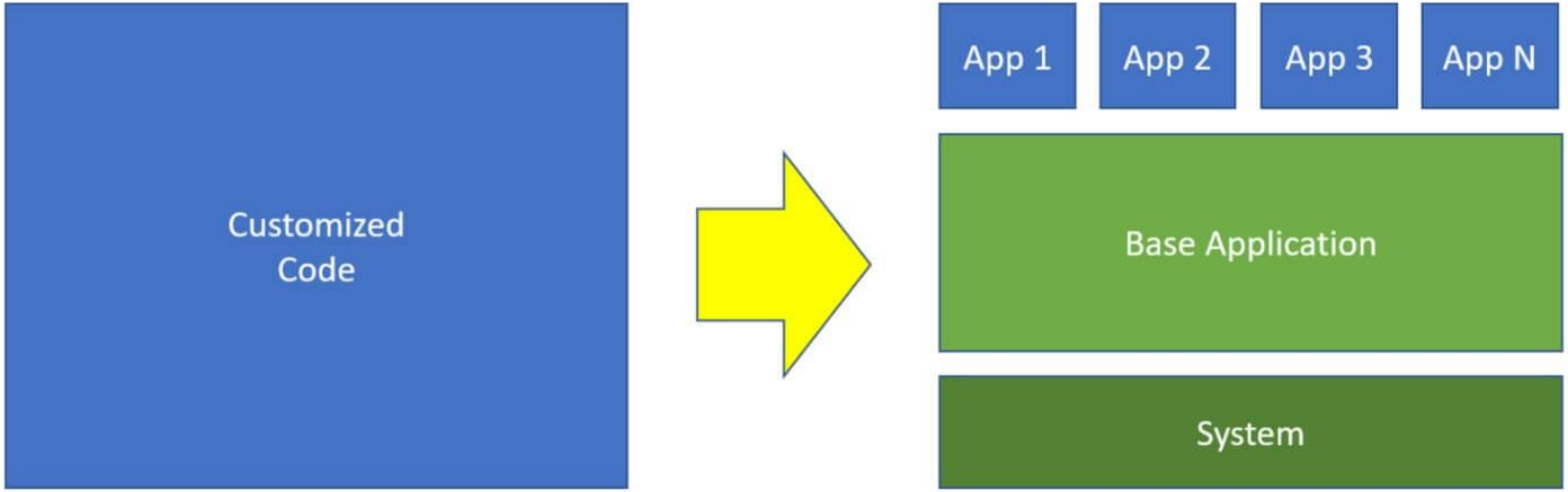
cs.com/v2.0/b379c622-33e3-429a-8018-cde5cf732898/TMCD0

amics.com

https://api.businesscentral.dynamics.com/v2.0/b379c622-33e3-429a-8018-cde5cf732898/TMCD01/api/v2.0/companies(7368d553-0c5c-ee11-8df4-6045bdc89477)/vendors

Extensions

Extensions


Customized Code
App 1
App 2
App 3
App N
Base Application
System

Extensions

Extensions are the way of building the Software as a Service (SaaS)

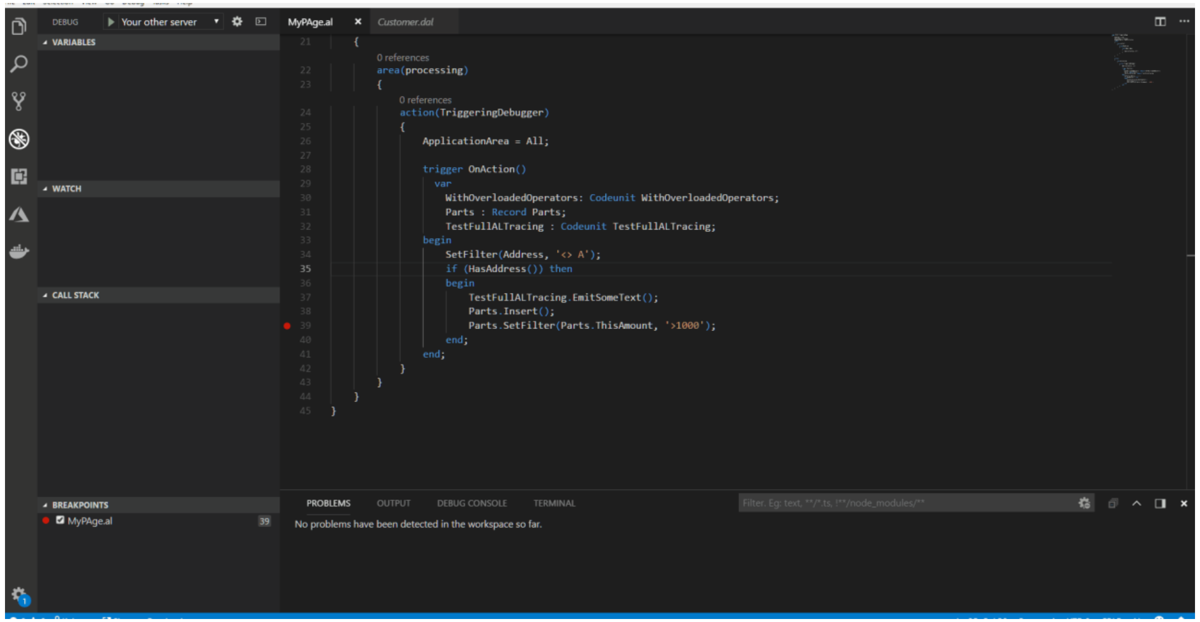
Does not alter the source code directly

Easier and low cost upgrades

Platform stability

Reduced time to deployment

Extensions in V2 (APPs)



```
    {
        0 references
        area(processing)
        {
            0 references
            action(TriggeringDebugger)
            {
                ApplicationArea = All;

                trigger OnAction()
                  var
                    WithOverloadedOperators: Codeunit WithOverloadedOperators;
                    Parts : Record Parts;
                    TestFullALTracing : Codeunit TestFullALTracing;
                begin
                    SetFilter(Address, '<> A');
                    if (HasAddress()) then
                    begin
                        TestFullALTracing.EmitSomeText();
                        Parts.Insert();
                        Parts.SetFilter(Parts.ThisAmount, '>1000');
                    end;
                end;
            }
        }
    }
}
```

No problems have been detected in the workspace so far.

- Development is done in Visual Studio Code (VSCode) – this is not the same as Vistual Studio!
- File based (objects stored as text files)
- Deployment can be done in VS Code

AL - Application Language

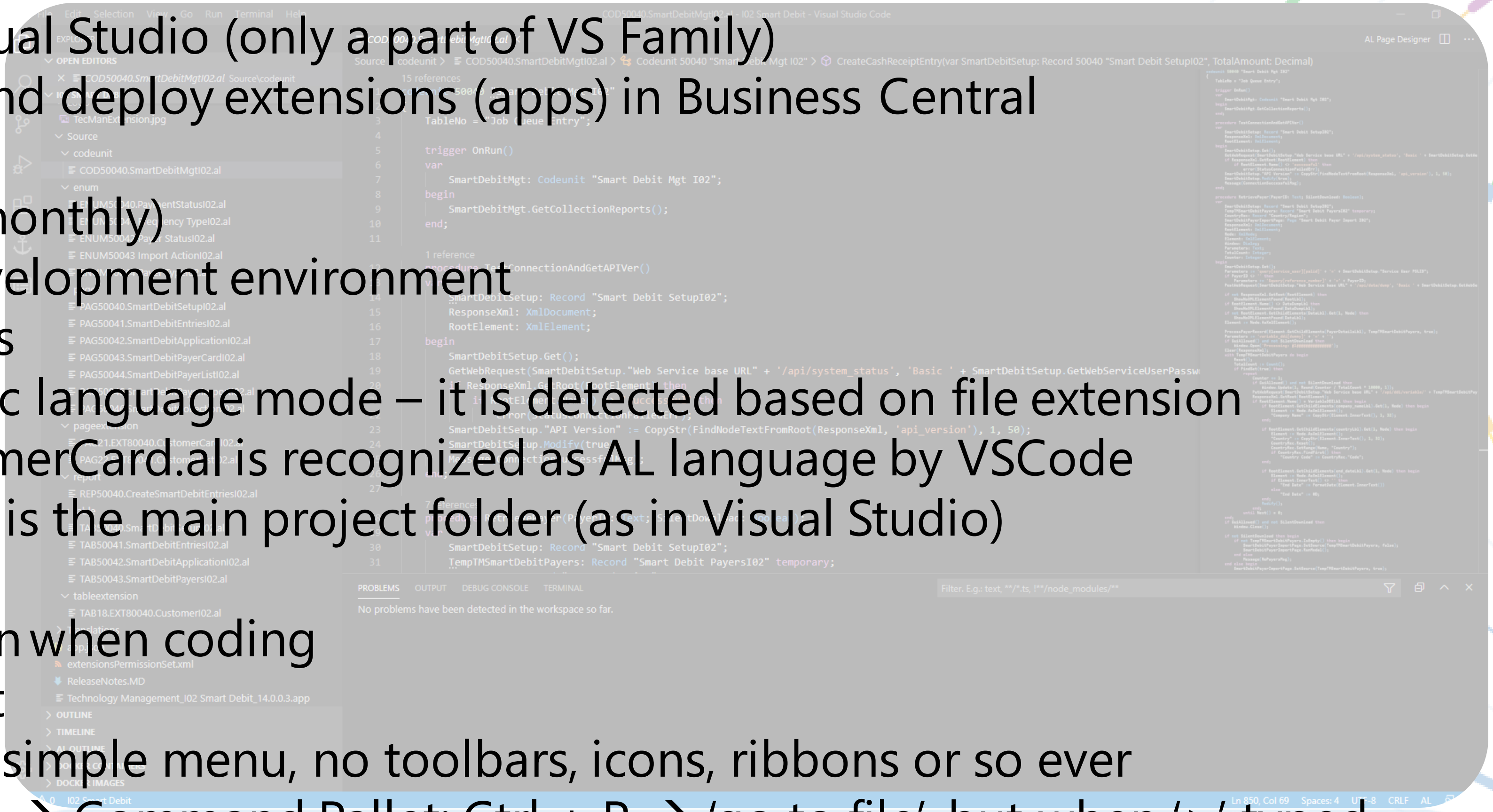
What can you do with extensions?

- Create new objects (tables, pages, reports, queries)
- Extend existing objects – except API objects
- Change standard system routine (if events available)
- New features are added on regular basis

VS Code

Visual Studio Code – VS Code

- VS Code is not Visual Studio (only a part of VS Family)
- Is used to create and deploy extensions (apps) in Business Central
- It is for free
- Always updated (monthly)
- Cross platform development environment
- Files in Workspaces
  - File is in specific language mode – it is detected based on file extension – i.e. a file CustomerCard.al is recognized as AL language by VSCode
  - Workspace – it is the main project folder (as in Visual Studio)
- Code Snippets
- Compiler validation when coding
- Native GIT support
- Command Pallet – simple menu, no toolbars, icons, ribbons or so ever
  - Ctrl + Shift + P → Command Pallet; Ctrl + P → 'go to file', but when '>' typed it open command pallet



Source > codeunit > COD50040.SmartDebitMgtI02.al > Codeunit 50040 "Smart Debit Mgt I02" > CreateCashReceiptEntry(var SmartDebitSetup: Record 50040 "Smart Debit SetupI02", TotalAmount: Decimal)

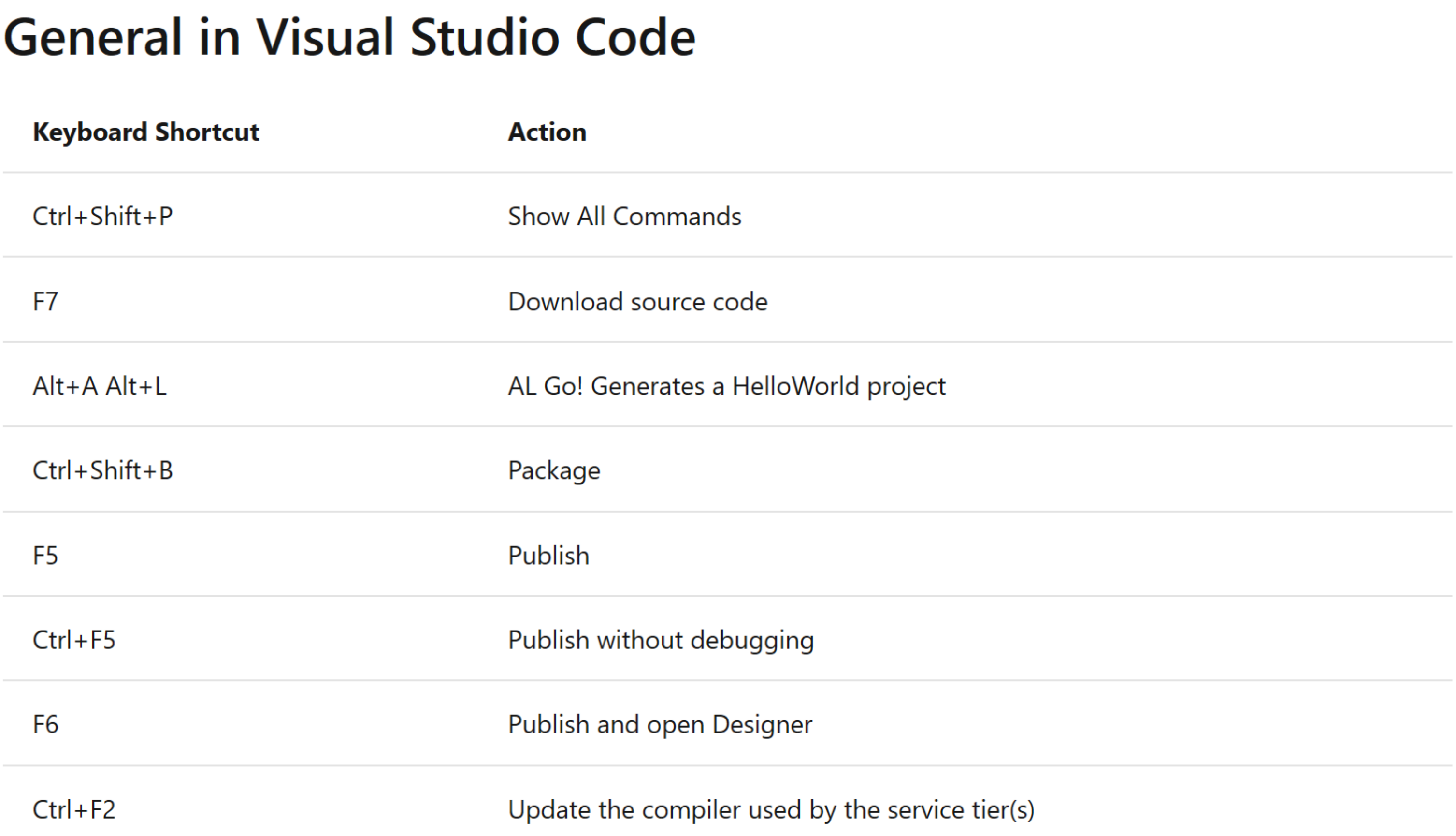
```
    TableNo = "Job Queue Entry";

    trigger OnRun()
    var
        SmartDebitMgt: Codeunit "Smart Debit Mgt I02";
    begin
        SmartDebitMgt.GetCollectionReports();
    end;

    procedure TestConnectionAndGetAPIVer()
    var
        SmartDebitSetup: Record "Smart Debit SetupI02";
        ResponseXml: XmlDocument;
        RootElement: XmlElement;
    begin
        SmartDebitSetup.Get();
        GetWebRequest(SmartDebitSetup."Web Service base URL" + '/api/system_status', 'Basic ' + SmartDebitSetup.GetWebServiceUserPassw
        if ResponseXml.GetRoot(RootElement) then
            [illegible]
                Error(StatusConnectionFailedErr);
        SmartDebitSetup."API Version" := CopyStr(FindNodeTextFromRoot(ResponseXml, 'api_version'), 1, 50);
        SmartDebitSetup.Modify(true);
        [illegible]
    end;

    [illegible]
    var
        SmartDebitSetup: Record "Smart Debit SetupI02";
        TempTMSmartDebitPayers: Record "Smart Debit PayersI02" temporary;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

No problems have been detected in the workspace so far.

VS Code - DEMO

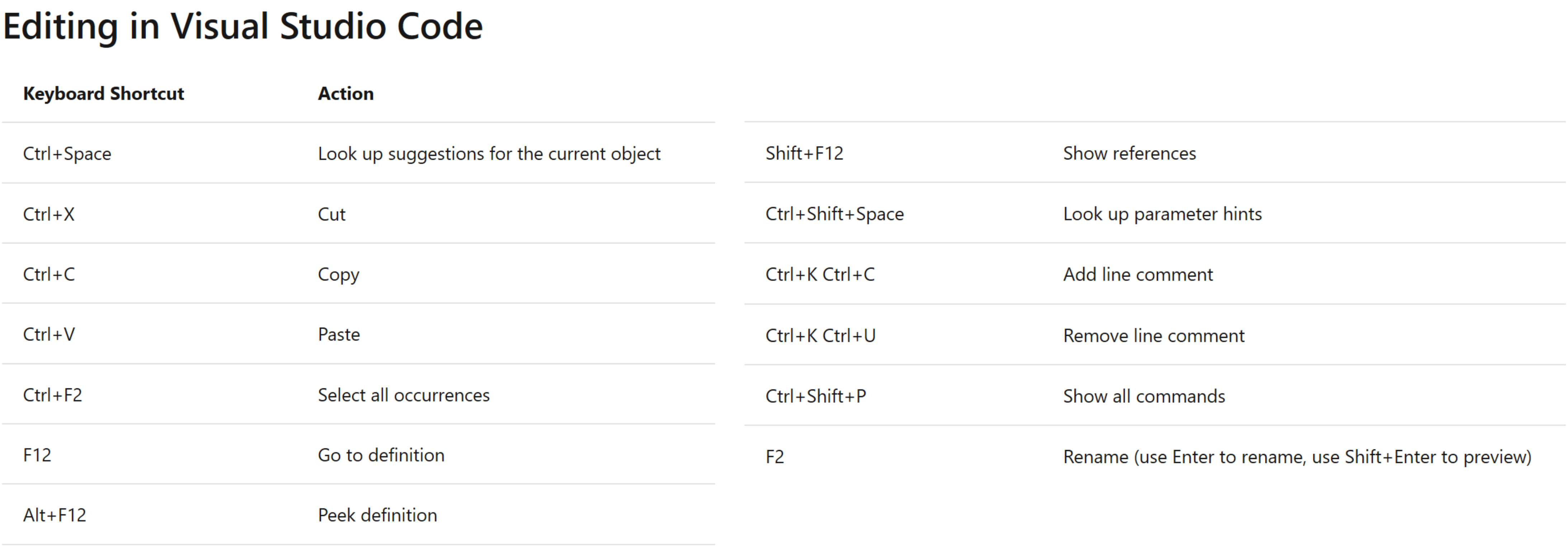
VSCode – useful keyboard shortcuts 1/3



General in Visual Studio Code

Keyboard Shortcut	Action
Ctrl+Shift+P	Show All Commands
F7	Download source code
Alt+A Alt+L	AL Go! Generates a HelloWorld project
Ctrl+Shift+B	Package
F5	Publish
Ctrl+F5	Publish without debugging
F6	Publish and open Designer
Ctrl+F2	Update the compiler used by the service tier(s)

VSCode – useful keyboard shortcuts 2/3



Editing in Visual Studio Code

Keyboard Shortcut	Action
Ctrl+Space	Look up suggestions for the current object
Ctrl+X	Cut
Ctrl+C	Copy
Ctrl+V	Paste
Ctrl+F2	Select all occurrences
F12	Go to definition
Alt+F12	Peek definition
Shift+F12	Show references
Ctrl+Shift+Space	Look up parameter hints
Ctrl+K Ctrl+C	Add line comment
Ctrl+K Ctrl+U	Remove line comment
Ctrl+Shift+P	Show all commands
F2	Rename (use Enter to rename, use Shift+Enter to preview)

VSCode – useful keyboard shortcuts 3/3

Errors in Visual Studio Code

Keyboard Shortcut	Action
F8	Move to the next error or warning
Shift+F8	Move to the previous error or warning

Compile in Visual Studio Code

Keyboard Shortcut	Action
Ctrl+Shift+B	Compile and build the solution
Ctrl+F5	Build and deploy

Debugging in Visual Studio Code

Keyboard Shortcut	Action
F5	Start debugging session
Ctrl+Alt+F5	Start RAD publishing without debugging.
Alt+F5	Start RAD with debugging.

AL Language Extension for VS Code

- Create Objects for Business Central
  - Write and interpreting AL Code
  - Syntax Highlighting (different syntax has different colour)
  - Intellisense (giving a hint what are the available options)
  - Code snippets for AL (templates of code for building objects and its parts)
- Compilation errors and warnings – you can pick the code analyser and exclude some rules (i.e. warnings)
- Deployment straight to Business Central (for dev/test/sandbox environment – for live use PowerShell)
- Automatic creation of workspace when using AL:Go! (from Command Pallet).

VSCode – Configuration files

App.json

- This is the application specific config
- Extension name & version
- Info about the Publisher
- Dependencies on other apps

Launch.json

- Server Location

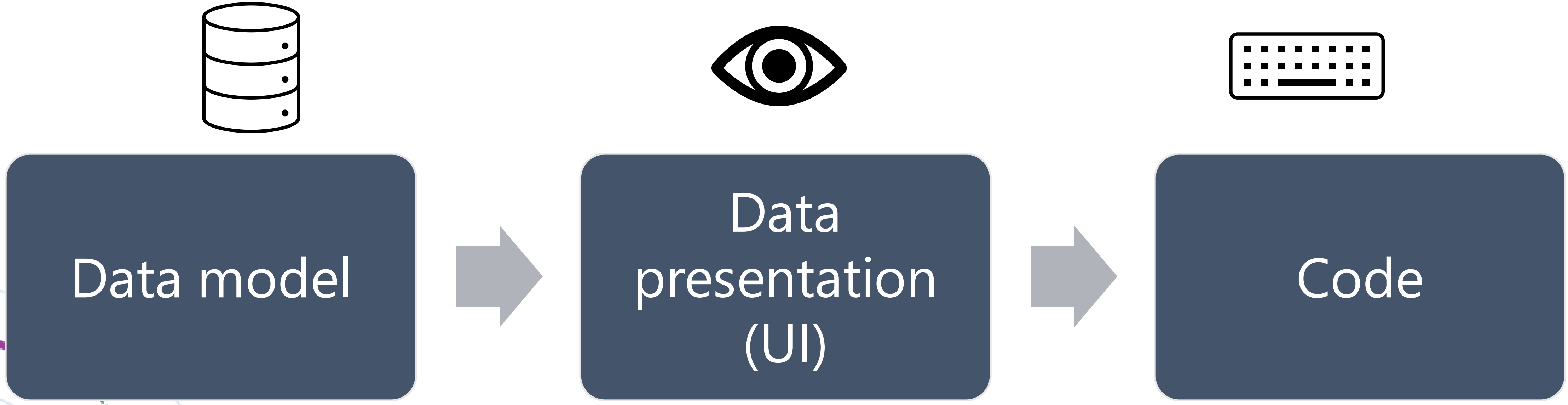
Settings.json

- Workspace settings – config for other VSCode extensions, Code Analyzer settings etc.

App.json and Launch.json are autogenerated when creating a new project with AL: Go!

Developing the extension

Developing the extension – main steps



Data model

Data presentation (UI)

Code

Data presentation

Page
(& page extension)

Report

XmlPort

Query

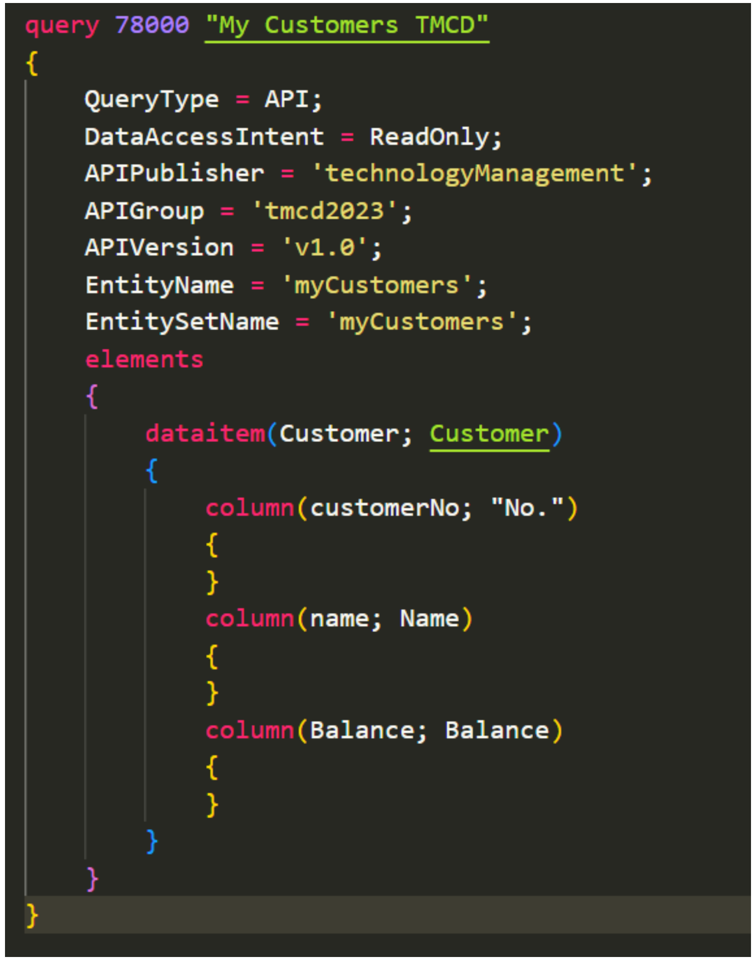
API Queries

API Queries

- Usually used for simple SQL queries
- Can be used to join data from different sources i.e. Customers and Sales Invoices, yet cannot have 2 datasets on same level (unions are not supported) i.e. Posted Sales Invoices and Posted Sales Shipments in one query
- Queries can also perform simple calculations on data, such as sum or average of all values in a column of the dataset
- You can pre-filter the dataset i.e. to show only non-blocked Customers, but it does not work on flowfields (i.e. inventory)

Queries

- Description: Define new query object for the extension
- File name example: QUE78000.MyCustomersTMCD.al
- Snippets:
  - tqueryapiwaldo (requires ext)
  - tdataitem
  - tcolumn
- Declaration:
  - Properties
  - Layout
    - elements, dataitem(table), column(field)



```
query 78000 "My Customers TMCD"
{
    QueryType = API;
    DataAccessIntent = ReadOnly;
    APIPublisher = 'technologyManagement';
    APIGroup = 'tmcd2023';
    APIVersion = 'v1.0';
    EntityName = 'myCustomers';
    EntitySetName = 'myCustomers';
    elements
    {
        dataitem(Customer; Customer)
        {
            column(customerNo; "No.")
            {
            }
            column(name; Name)
            {
            }
            column(Balance; Balance)
            {
            }
        }
    }
}
```

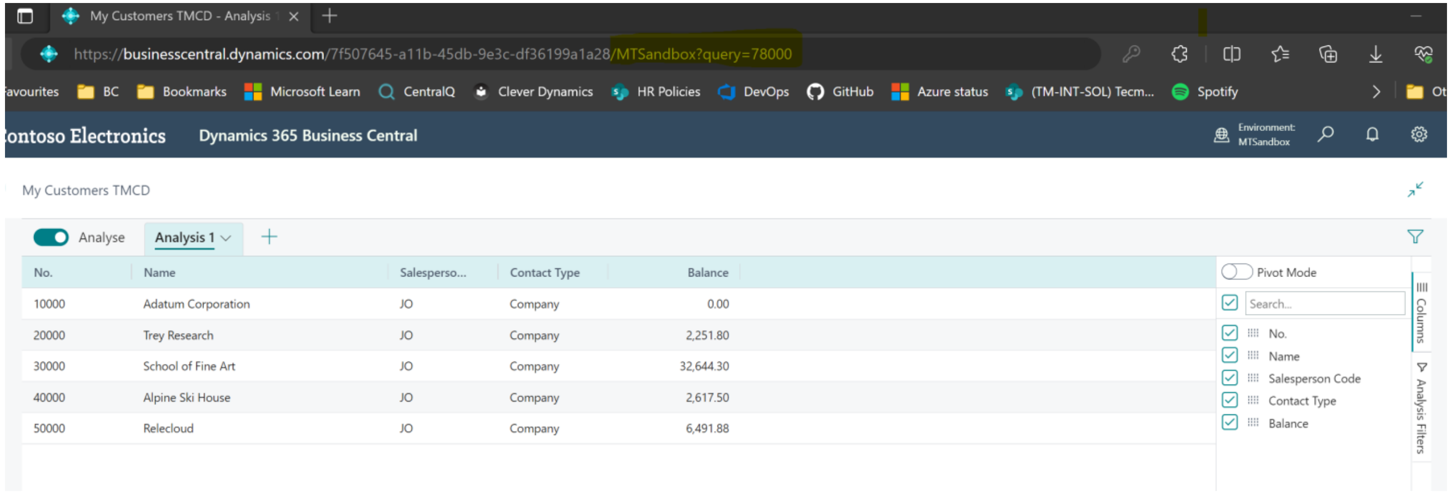
API – naming conventions

- camelCase - for naming attributes, tables, as well as APIPublisher, APIGroup, EntityName, and EntitySetName.
- Alphanumeric characters allowed (A-Z+a-z+0-9) in above elements.
- APIVersion follows the pattern vX.Y or beta.

View your API data

- Via browser:
  - Append "?<objectType>=<objectNo>" to the end of the base URL:

www.businesscentral.dynamics.com/7f507645-a11b-45db-9e3c-df36199a1a28/MTSandbox?query=78000

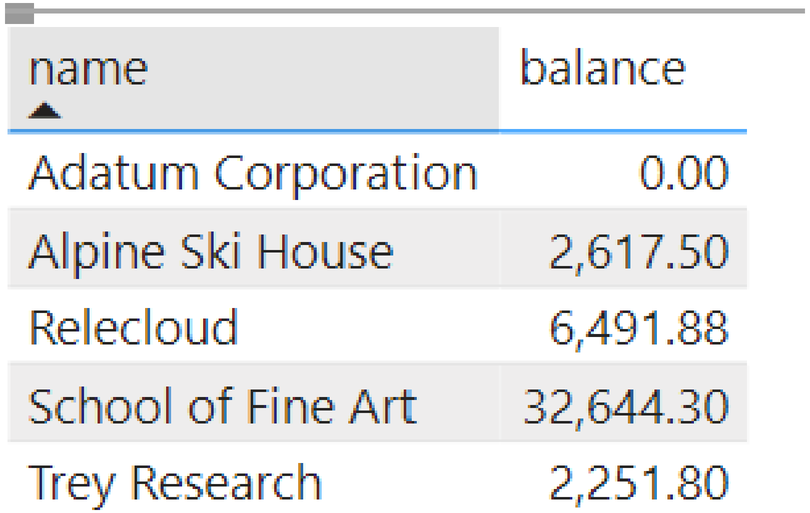


No.	Name	Salesperso...	Contact Type	Balance
10000	Adatum Corporation	JO	Company	0.00
20000	Trey Research	JO	Company	2,251.80
30000	School of Fine Art	JO	Company	32,644.30
40000	Alpine Ski House	JO	Company	2,617.50
50000	Relecloud	JO	Company	6,491.88

☑ No.
☑ Name
☑ Salesperson Code
☑ Contact Type
☑ Balance

View your API data

- Power BI
  - Connect Power BI to your BC tenant

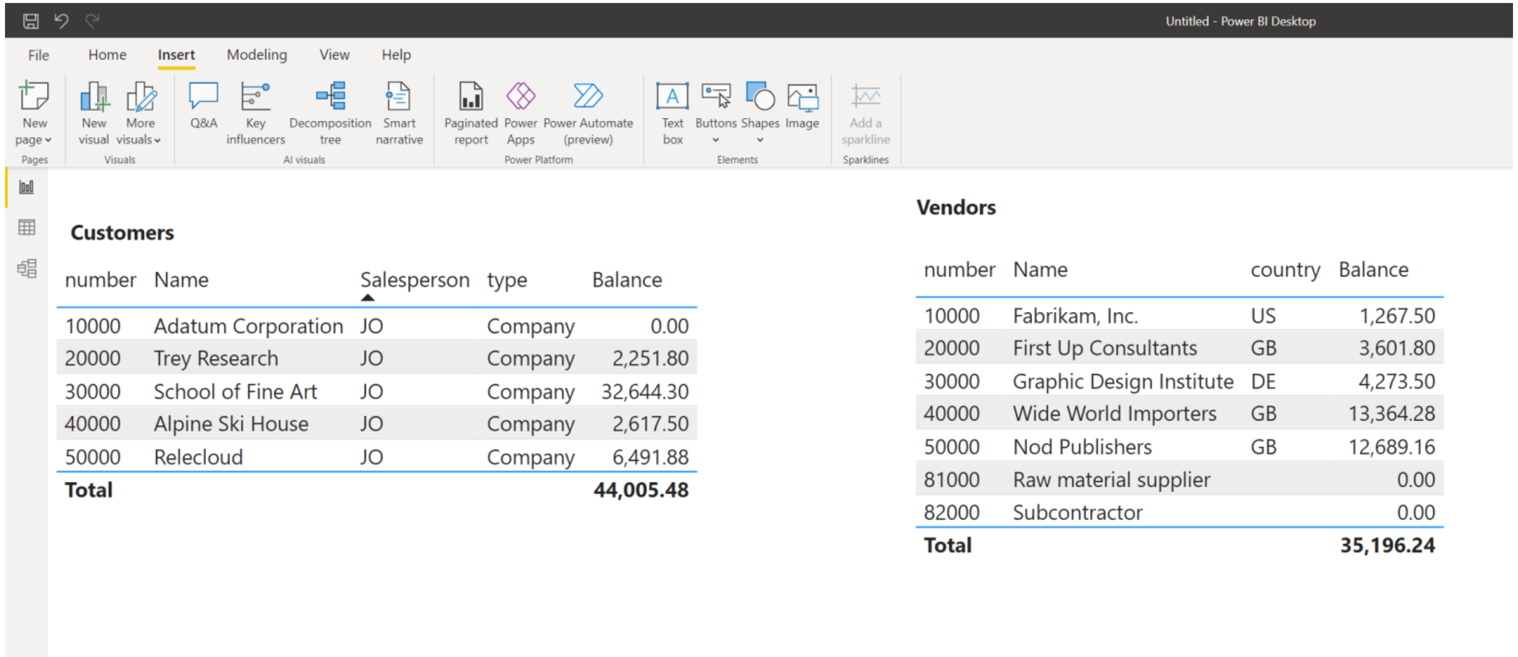


name ▲	balance
Adatum Corporation	0.00
Alpine Ski House	2,617.50
Relecloud	6,491.88
School of Fine Art	32,644.30
Trey Research	2,251.80

Creating an API query

Exercise 1 – My Customers and Vendors

Create an API Page for Customers and API Query for Vendors so the user can create a report in PowerBI like this:



Customers

number	Name	Salesperson	type	Balance
10000	Adatum Corporation	JO	Company	0.00
20000	Trey Research	JO	Company	2,251.80
30000	School of Fine Art	JO	Company	32,644.30
40000	Alpine Ski House	JO	Company	2,617.50
50000	Relecloud	JO	Company	6,491.88
Total				**44,005.48**

Vendors

number	Name	country	Balance
10000	Fabrikam, Inc.	US	1,267.50
20000	First Up Consultants	GB	3,601.80
30000	Graphic Design Institute	DE	4,273.50
40000	Wide World Importers	GB	13,364.28
50000	Nod Publishers	GB	12,689.16
81000	Raw material supplier		0.00
82000	Subcontractor		0.00
Total			**35,196.24**

API Pages

API Pages

- Usually used for more complex data integration
- Structures data similar to that of Business Central pages, including fields, data types, and relations
- Pre-filter the dataset i.e. to show only non-blocked Customers, and it works on flowfields
- Perform complex calculations on data, using the triggers to manipulate variables to display data

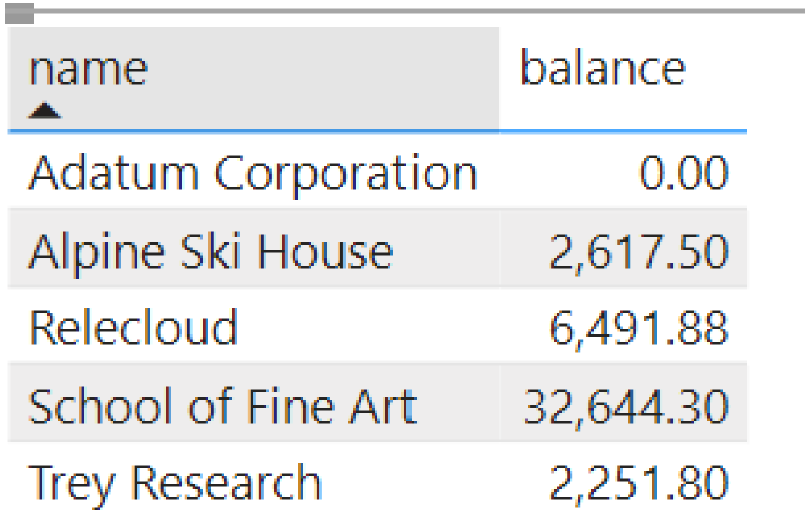
API Page – operations

- Endpoints which allow:
    - Read
    - Create, in order to disallow use InsertAllow = false
    - Update, in order to disallow use ModifyAllow = false
    - Delete, in order to disallow use DeleteAllow = false
    - Run actions

- You can also:
    - Filter records
    - Apply sorting

View your API data

- Power BI
  - Connect Power BI to your BC tenant

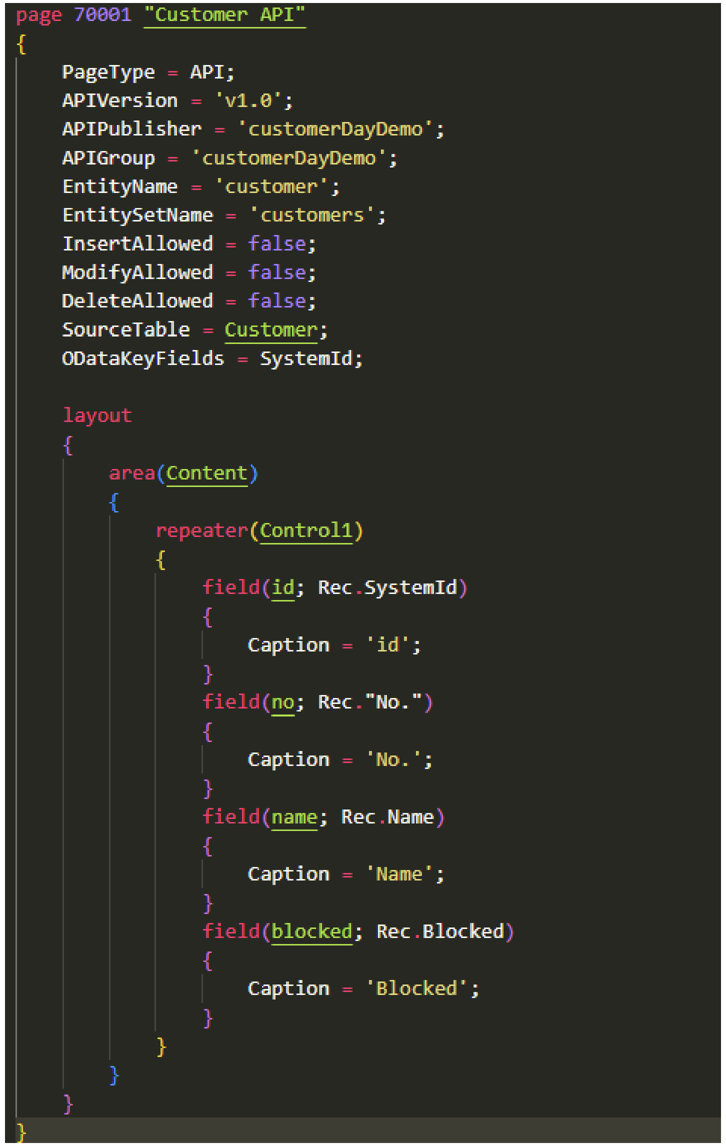


name ▲	balance
Adatum Corporation	0.00
Alpine Ski House	2,617.50
Relecloud	6,491.88
School of Fine Art	32,644.30
Trey Research	2,251.80

Pages (page)

- Description: Define new page object for the extension
- File name example: PAG50000.CustomerCardM01.al
- Snippets:
  - tpageapiwaldo (requires ext)
  - tfieldpage
  - ttrigger
- Declaration:
  - Variables (local and globals)
  - Properties
  - Layout
    - Areas, groups, parts, fields
  - Actions
  - Code

API Page - example



```
page 70001 "Customer API"
{
    PageType = API;
    APIVersion = 'v1.0';
    APIPublisher = 'customerDayDemo';
    APIGroup = 'customerDayDemo';
    EntityName = 'customer';
    EntitySetName = 'customers';
    InsertAllowed = false;
    ModifyAllowed = false;
    DeleteAllowed = false;
    SourceTable = Customer;
    ODataKeyFields = SystemId;

    layout
    {
        area(Content)
        {
            repeater(Control1)
            {
                field(id; Rec.SystemId)
                {
                    Caption = 'id';
                }
                field(no; Rec."No.")
                {
                    Caption = 'No.';
                }
                field(name; Rec.Name)
                {
                    Caption = 'Name';
                }
                field(blocked; Rec.Blocked)
                {
                    Caption = 'Blocked';
                }
            }
        }
    }
}
```

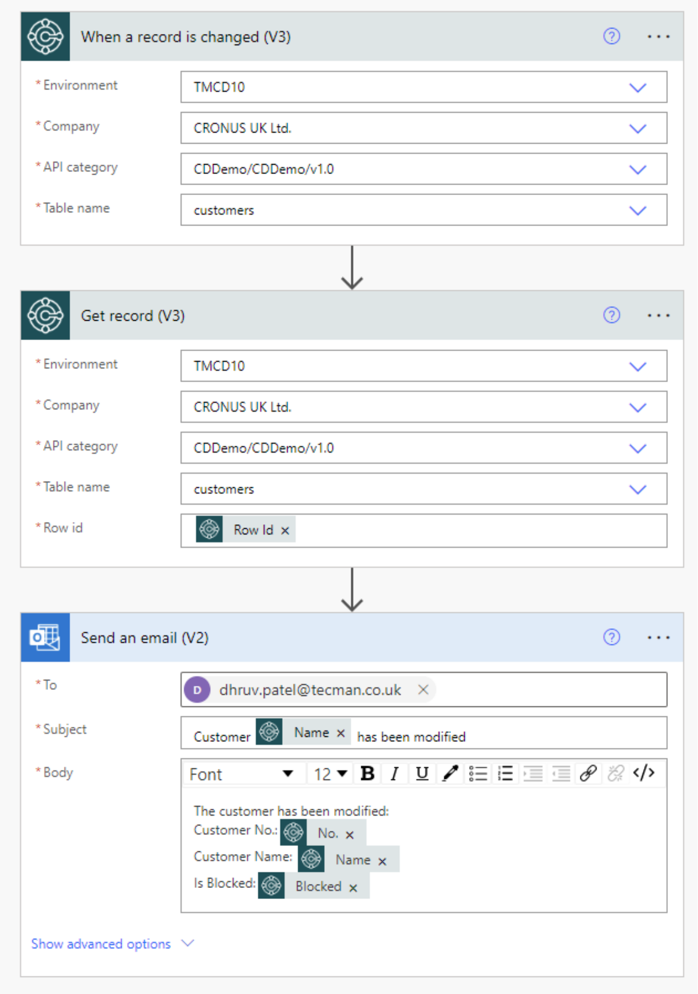
- Things to remember:
  - Setting InsertAllowed, ModifyAllowed, and DeleteAllowed to false will heavily improve performance
  - Using filters on the page to narrow down results will also improve performance (instead of filtering after loading the data such as on Power BI)
  - SystemId field of the table is required as a page field
  - ODataKeyFields must be set to SystemId

    (This is not required on queries)
  - Naming convention is the same in both API Pages and Queries

Creating an API page

Demo – Modified Customer

- Create a Power Automate flow to email the user when a Customer has been modified



When a record is changed (V3)

- *Environment: TMCD10
- *Company: CRONUS UK Ltd.
- *API category: CDDemo/CDDemo/v1.0
- *Table name: customers

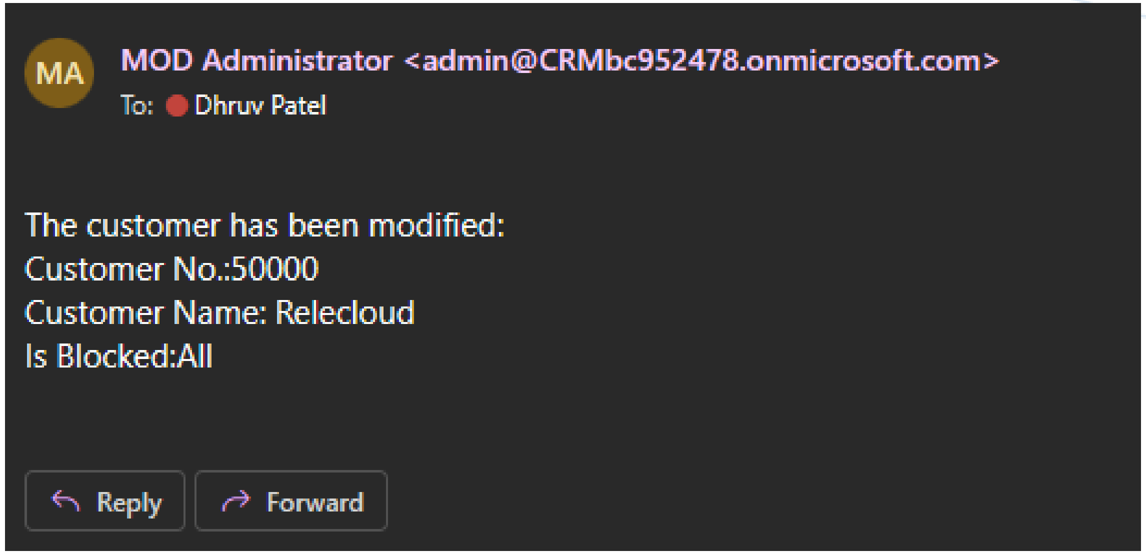
Get record (V3)

- *Environment: TMCD10
- *Company: CRONUS UK Ltd.
- *API category: CDDemo/CDDemo/v1.0
- *Table name: customers
- *Row id: Row Id

Send an email (V2)

- *To: dhruv.patel@tecman.co.uk
- *Subject: Customer Name has been modified
- *Body:

The customer has been modified:
Customer No.: No.
Customer Name: Name
Is Blocked: Blocked



MA MOD Administrator <admin@CRMbc952478.onmicrosoft.com>
To: Dhruv Patel

The customer has been modified:
Customer No.:50000
Customer Name: Relecloud
Is Blocked:All

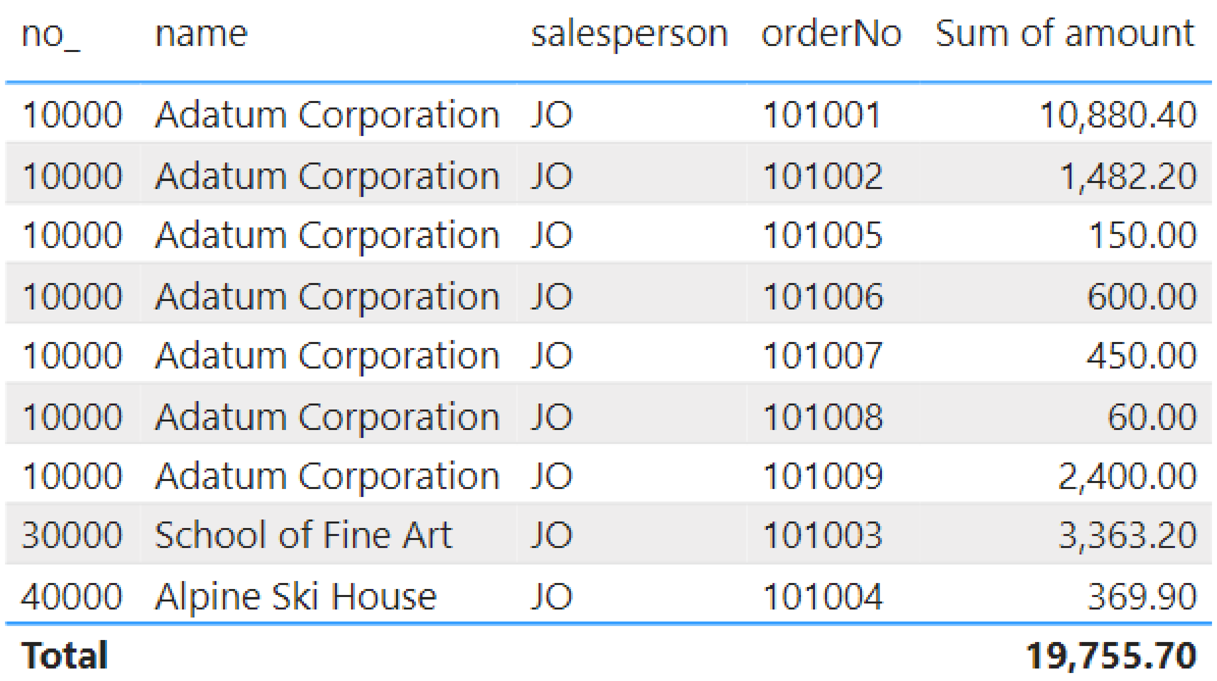
Exercise 2 – Item Modifications

- A user has requested they are notified by email when any Item is modified
- Create a Page API and power automate flow to send an email to the users Outlook

How about more practice?

Exercise 3 – My Customer Orders

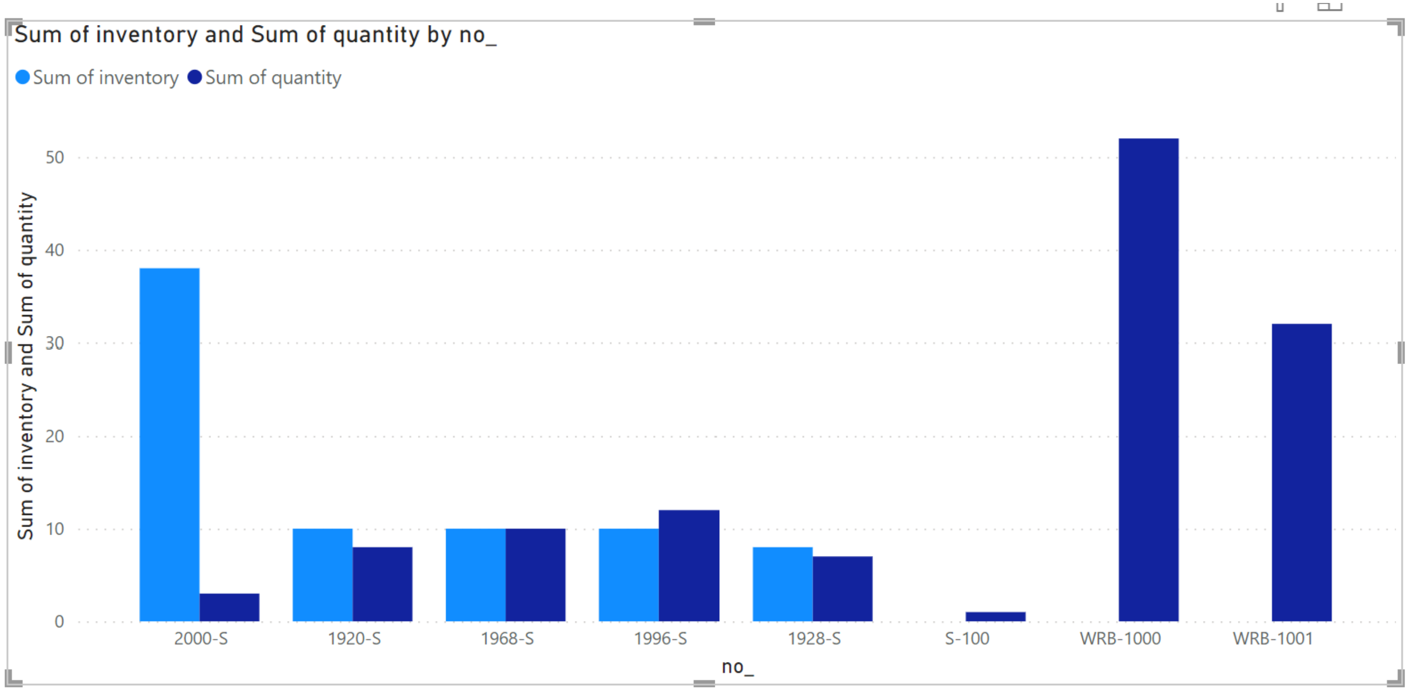
Create an API Query that will allow to prepare a list of Customers and their Sales Orders.



no_	name	salesperson	orderNo	Sum of amount
10000	Adatum Corporation	JO	101001	10,880.40
10000	Adatum Corporation	JO	101002	1,482.20
10000	Adatum Corporation	JO	101005	150.00
10000	Adatum Corporation	JO	101006	600.00
10000	Adatum Corporation	JO	101007	450.00
10000	Adatum Corporation	JO	101008	60.00
10000	Adatum Corporation	JO	101009	2,400.00
30000	School of Fine Art	JO	101003	3,363.20
40000	Alpine Ski House	JO	101004	369.90
Total				**19,755.70**

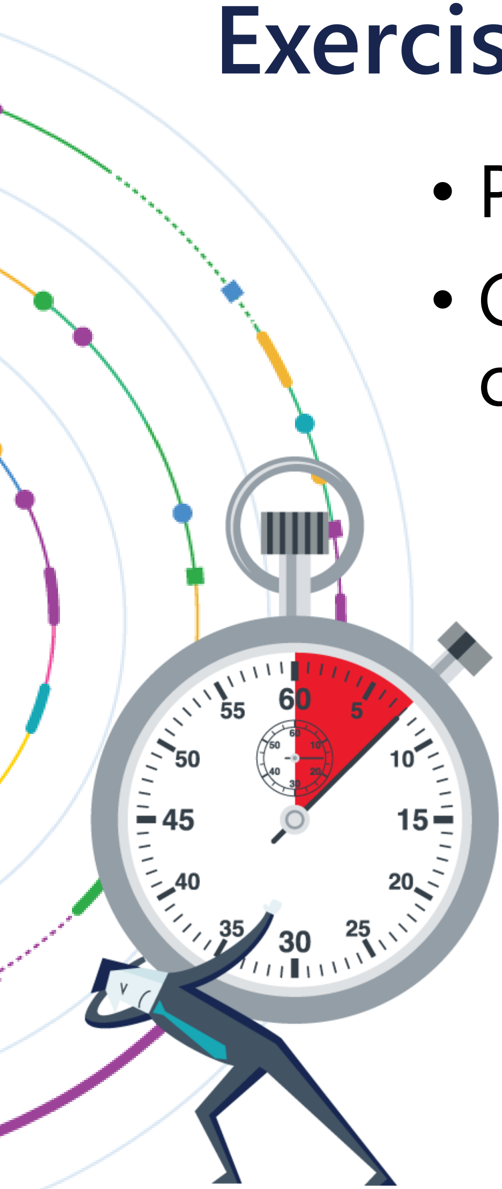
Exercise 4 – Compare Inventory with Orders

Create a query to show, for Items, Inventory and Sum of quantity on Sales Order in Power BI



Exercise 5 – My Customers

- Part 1: A user wants to view all customers
- Create an API Page for Customers so a list of customers can be viewed in Power BI

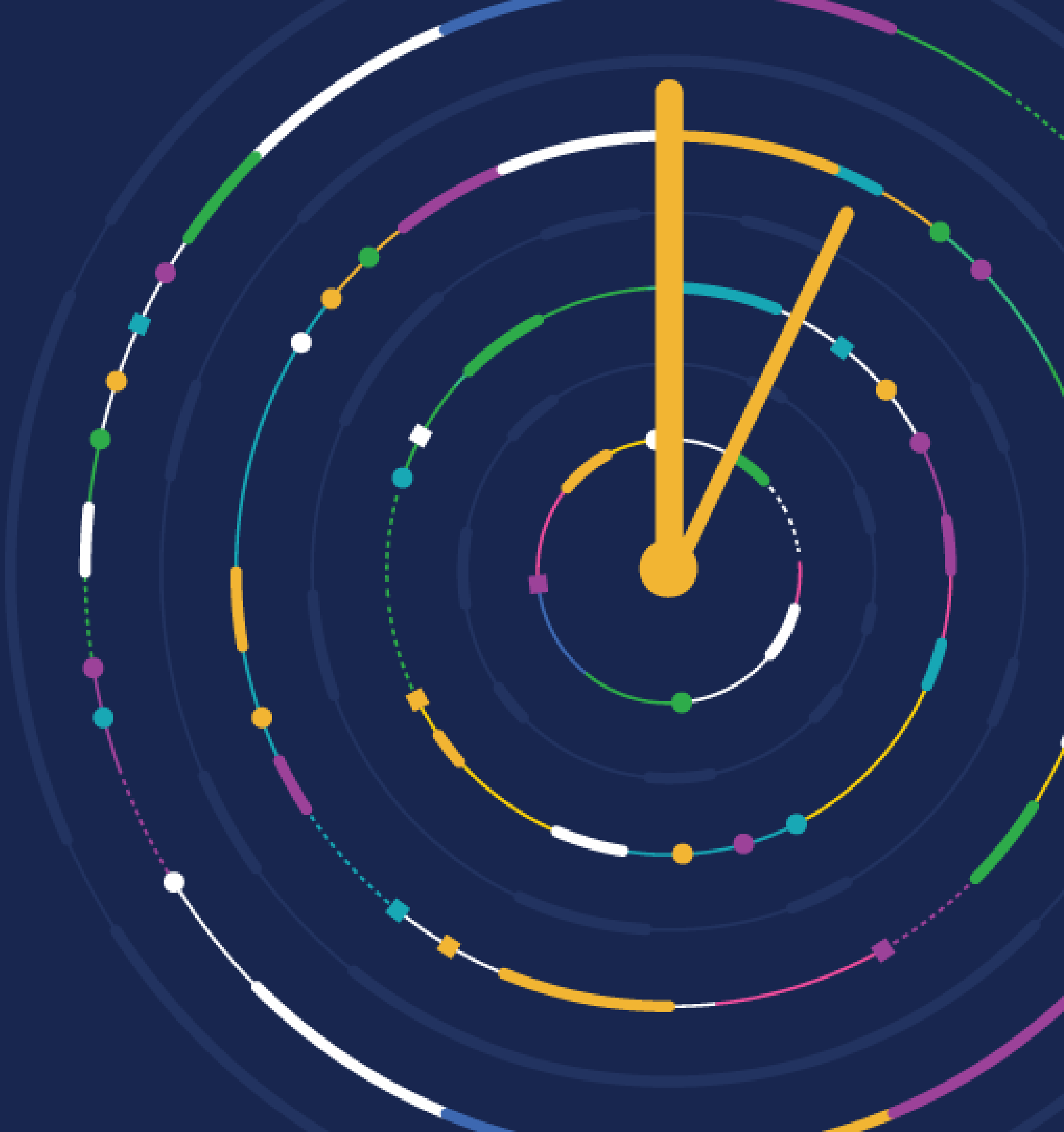


- Part 2: Now the user only needs to view customers that have the "Blocked" field set to "All"
- Add filters to the API page and view the data in Power BI again

QUESTIONS ?

tecman

THANK YOU



CUSTOMER DAY 2023