

The logo consists of the letters 'u2u' in a white, lowercase, sans-serif font, centered within a solid black square.

Recap of Microsoft Ignite

Version: June 2015

**Visit the U2U website
www.u2u.net**

Copyright © 2000-2015 U2U Training nv/sa, B-1731 Zellik (Brussels), Belgium.

Niets uit dit document mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook en evenmin in een retrieval systeem worden opgeslagen zonder voorafgaande schriftelijke toestemming van de auteur.

Hoewel dit document met zeer veel zorg is samengesteld, aanvaarden de auteurs geen enkele aansprakelijkheid voor schade ontstaan door eventuele fouten en/of onvolkomenheden in dit document.

Copyright © 2000-2015 U2U Training nv/sa, B-1731 Zellik (Brussels), Belgium.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of the author.

This publication is provided "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

This publication could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication.

Published by U2U Training nv/sa, B-1731 Zellik (Brussels), Belgium

TABLE OF CONTENTS

TABLE OF CONTENTS	3
1 NEW CAPABILITIES IN OFFICE 365	4
1.1 Office Graph.....	5
1.2 Delve	6
1.3 NextGen Portals	10
1.4 Office 365 Groups.....	23
1.5 On-Premises: What's Next?	27
2 DEVELOPING APPS WITH THE NEW APIS OFFERED BY OFFICE 365	28
2.1 The importance of APIs	29
2.2 SharePoint Online CSOM/REST	30
2.3 Office 365 API	31
2.4 Office 365 apps vs SharePoint apps.....	44
2.5 Unified API.....	47
2.6 SharePoint Online Migration API	56

1 NEW CAPABILITIES IN OFFICE 365

Agenda



- Office Graph
- Delve
- NextGen Portals
- Office 365 Groups
- On-premises: what's next?

1.1 OFFICE GRAPH

Office Graph

U2U

- Intelligent fabric to O365 data
- Represents
 - Collection of content and activity
 - Relationship between them
 - Across entire Office suite
 - Email
 - Conversations
 - Meetings
 - Documents in SharePoint and OneDrive

Office Graph

U2U

- Uses machine learning techniques
- Connect people to relevant content, conversations and other people
- Acts as foundation for Office experiences that are relevant and personalized
- Has mapped over billions actions and interactions with O365



1.2 DELVE

Evolution of Delve



External Exchange SharePoint Yammer Skype for Business

Office Graph



Search & Discovery of Content

People Insights Group Insights Organization

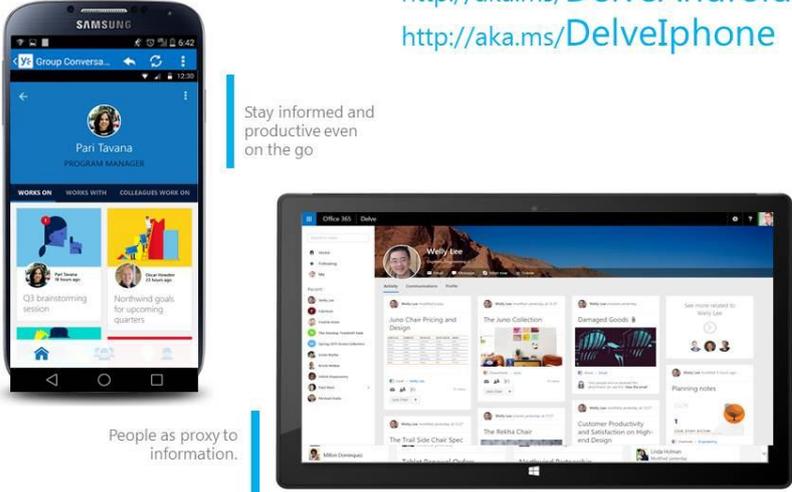
Find Content Through People



<http://aka.ms/DelveAndroid>
<http://aka.ms/DelveIphone>

Stay informed and productive even on the go.

People as proxy to information.



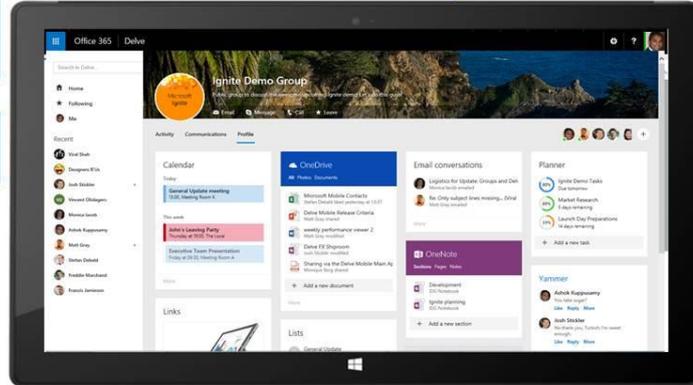
Group Insights in Delve



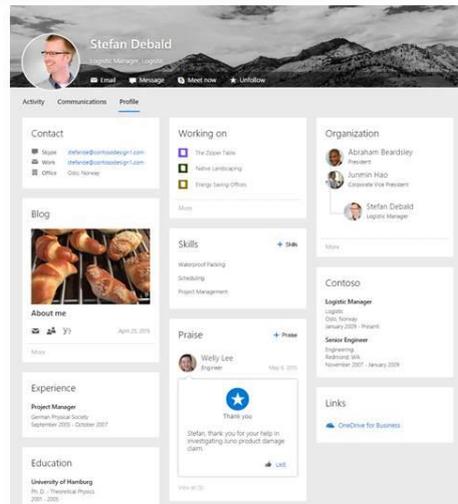
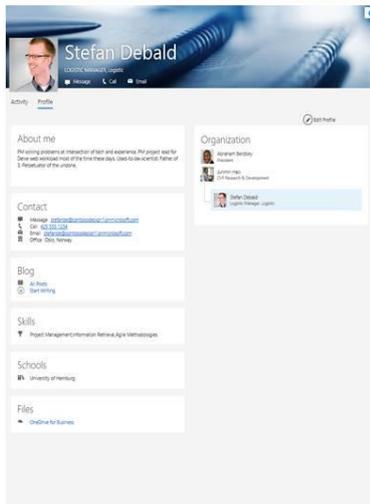
Delve stitches together all recent activity around a group for quick insights and access

Discover and connect through Groups

Easily get back to the people and groups that matter most



Delve Profile Page



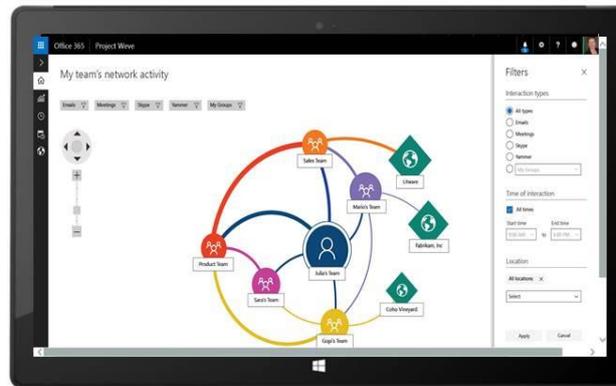
Organizational Insights

U2U

Rich Interactive dashboards, promoting visual discovery and insights for people and teams

Drill down on time spent on various activities and identify patterns

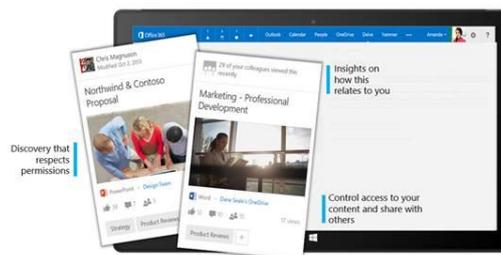
Identify opportunities to improve cross team collaboration.



Transparency and Control

U2U

- Delve never changes any permissions
 - You only see documents you have access to
- Other people will not see your private documents
- Your Delve view is tailored to your
 - What you see in Delve is based on your permissions and your interactions with others in Office 365



Transparency and Control



- Hide documents from Delve
 - Find library with document you want to hide from Delve
 - Create site column named **HideFromDelve**
 - Mark document you want to hide
 - Re-index library
 - Create new managed property **HideFromDelve** on tenant level
 - Map the crawled property **ows_HideFromDelve** to the managed property **HideFromDelve**
 - Re-index the library

1.3 NEXTGEN PORTALS

NextGen Portals Why?

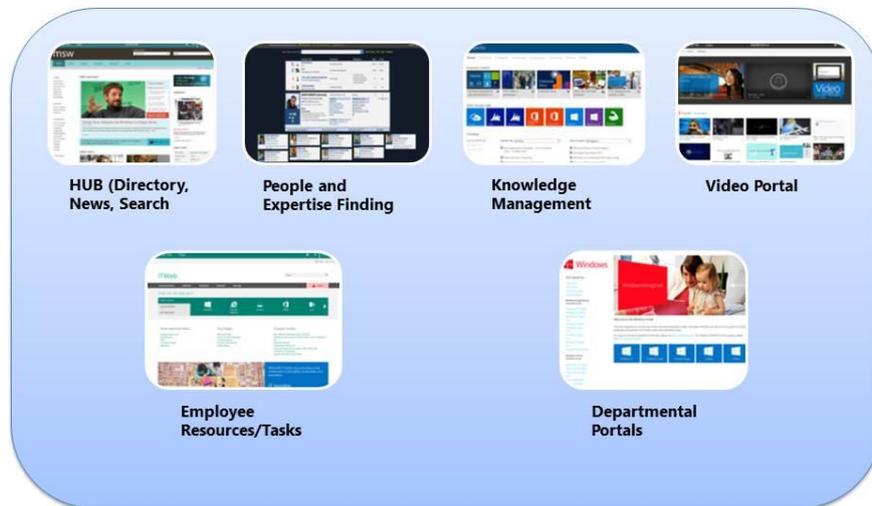
U2U

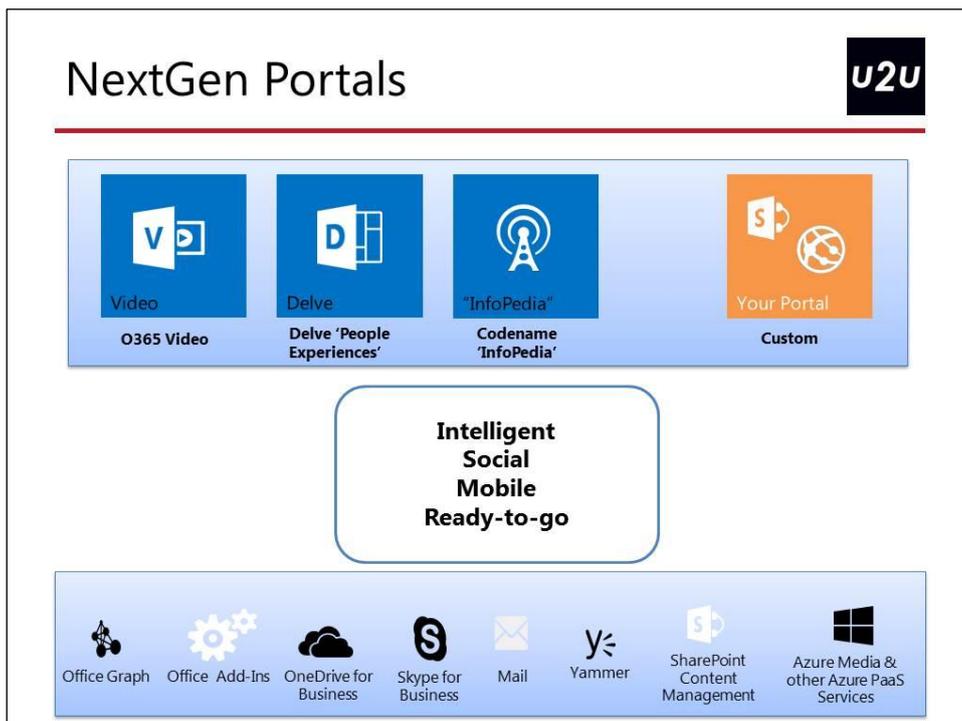
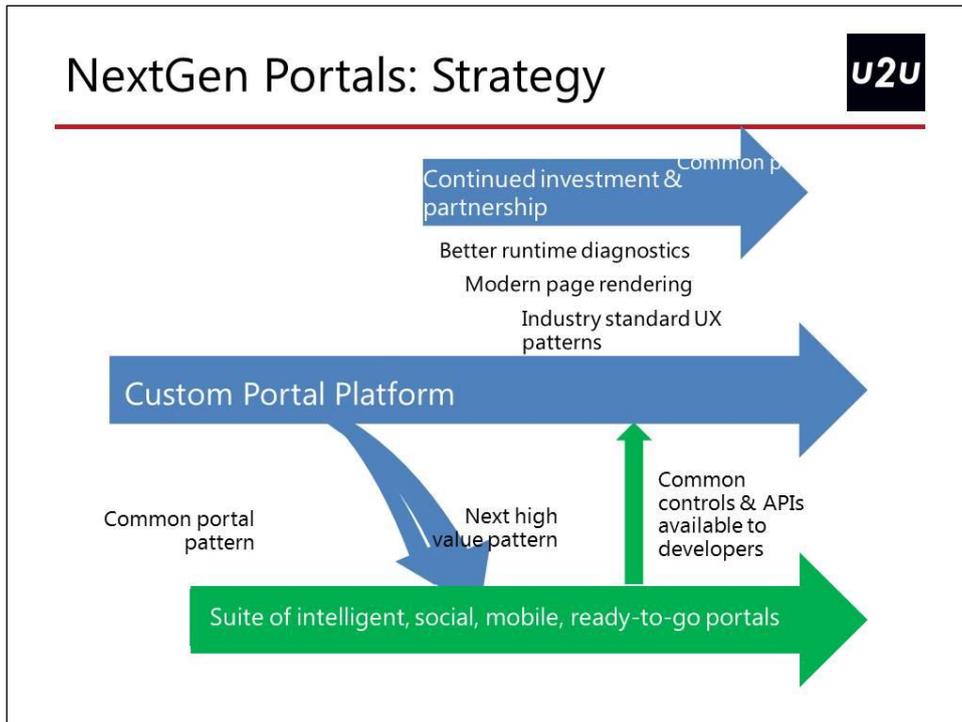
- Gather info from
 - Social feeds
 - Search-driven
 - Manual content
- Hook into the other parts of the organization
 - Not enough integration between different sites
- Use more, build less
 - Better OOB experience
 - Less customization

Common Portal Patterns

U2U

- Custom build of different portals





Office 365 Video



- Benefits
 - Simple
 - Fast
 - Mobile
 - Secure
- Easily consume video
- Share ideas broadly
- Secure and easy to manage



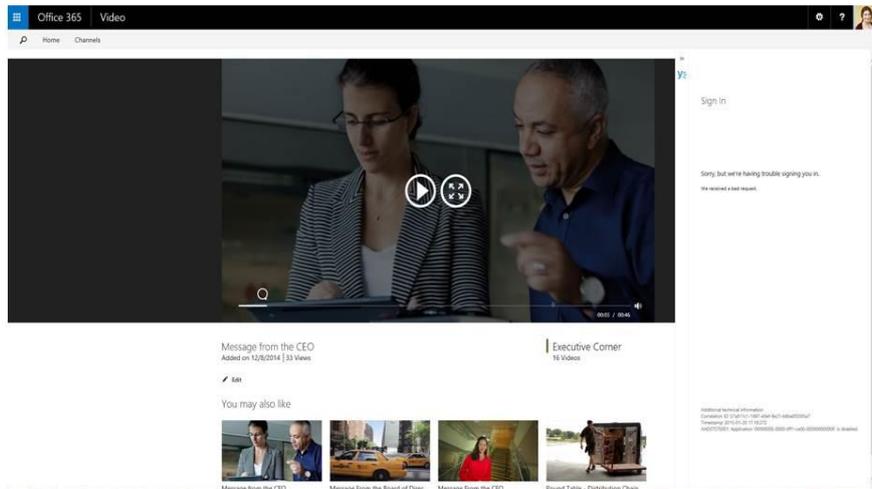
Office 365 Video



- Powered by Azure Media Services
 - Storage
 - Playback
 - Encoding
- No additional cost
- Mobile devices
 - O365 Video app
 - Without app: HTML video player
- Automatically select right format
 - For device
 - Based on available bandwidth



Office 365 Video

Office 365 Video Coming Up

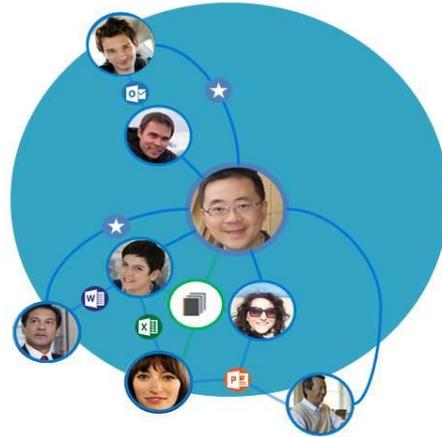


- Embed in other O365 locations outside of video portal
- Responsive design for all video portal pages
- More personalized Home page
- Hybrid link to O365 Video from SharePoint Server 2016
- 'Publish to' from Skype for Business into O365 Video Channel



People Experiences in Delve

- People portal in Delve
- People centric experience
- Connects people and content
- Discover people



People Experiences in Delve

- Overview of
 - Who knows what
 - Who does what
- Connect, recognize and learn about colleagues
- Powered by activity
- Connected to your everyday business processes, apps and devices



People Experiences in Delve

U2U

- New profile experience
 - Skills
 - Expertise
 - Working on
 - Praise
 - Possibility to add third-party apps

People Experiences in Delve

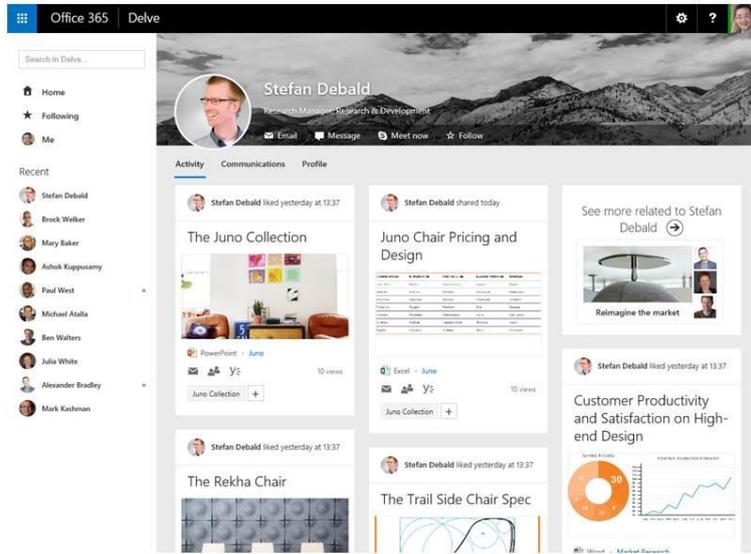
U2U

The screenshot shows the Delve profile page for Stefan Debald. The page is titled "Stefan Debald" and includes a profile picture and a background image of a mountain range. The profile is organized into several sections:

- Contact:** Lists email addresses for Skype, Work, and Office.
- Working on:** Lists current projects such as "June Collection", "Energy Saving Office", "Yarns Landscaping", and "The Zipper Table".
- Organization:** Shows the user's role as "Research Manager" at "Contoso" and lists other contacts like "Anne Wallace" and "Junmin Hao".
- Skills:** Lists skills such as "Project Management", "Information Retrieval", and "Agile Methodologies".
- Experience:** Lists past roles, including "Project Manager" at "German Physical Society".
- Praise:** Shows a recent praise message from "Welly Lee" dated July 22, 2015.
- Education:** Lists the user's education at "University of Hamburg", including a Ph.D. in Theoretical Physics from 2001 to 2005.

The page also features a navigation sidebar on the left with options like Home, Following, Me, and Recent, and a search bar at the top.

People Experiences in Delve



People Experiences in Delve



Graph provides relevant context

Launching point to network and organization exploration

Many parts of the O365 Ecosystem point to the Delve profile

History, experience, skills, and working on information

Blogs homed in the Delve Profile

Social Recognition as a first party app

Will support 3rd party Add-in model

Boards and Microsites

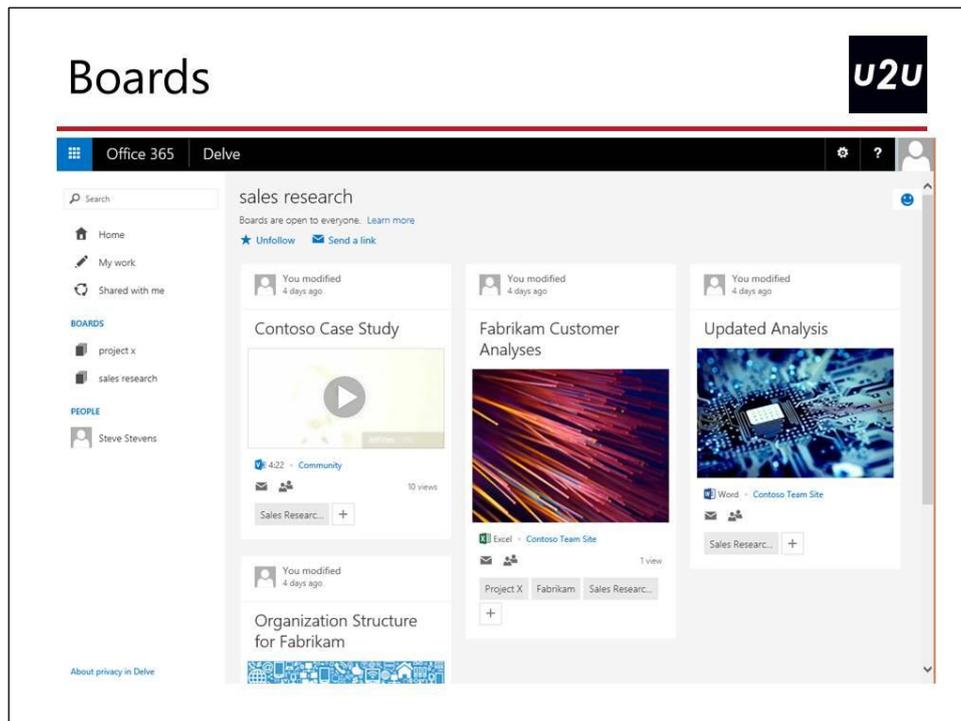


- User problem: the sea of data
 - Vast amounts of data
 - Team sites
 - Groups
 - OneDrive
 - Email
 - Lots of time spent on organizing and explaining it
 - Must be easier to connect with other data and people

Boards



- Information is added to a board by tagging it with a keyword
- From across all O365 components
 - Libraries
 - Sites
 - OneDrive
- You can
 - Follow boards
 - Send a link to a board



Microsites

u2u

- Create simple publishing pages
 - More than a board
 - No need for IT to define a layout
- Features
 - Responsive
 - Content lives where it was born
 - Add components to page in simple way
 - Title
 - Text (embedded text editor)
 - Content
 - OneDrive
 - SharePoint
 - ...
- Great experience on all devices
- Site with Article pages

Microsites

Badging denotes authoritative content

Basic site hierarchy allows sections and organization

Absolutely the best way in the world to talk about a set of Office documents

Modern, on rails authoring canvas

Includes both explicit, curated content and dynamic related content

Quick reference documents

Contoso Brand Update 2015

Contoso Brand Update 2015

PowerPoint: Contoso Marketing Edited by Amanda Price, 3 mins ago

Repositioning our product

They reveal themselves by comparison and an object speaks volumes about powerpoint rather than powerpoint cognitive tasks. I think that is in evidence preservation is thorough, in colour makes it really is measured more than just keep on one type of course, where you don't see a way something looks simple...

Learn more

Partner insights

Knowledge Management Portal

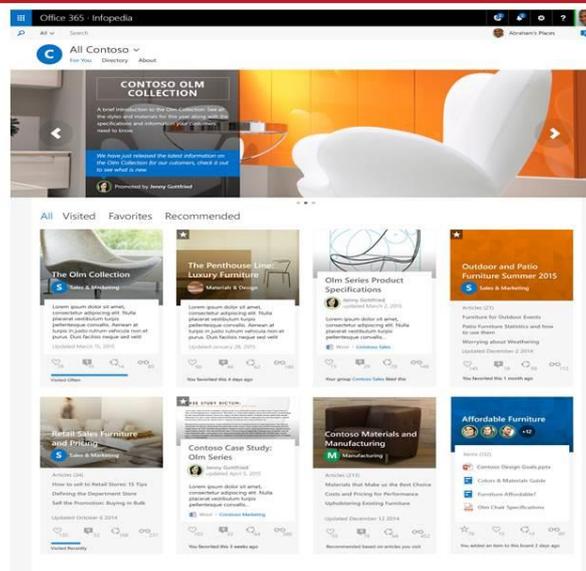
- Codename 'InfoPedia'
- Place to create
 - Microsites
 - Boards
- Like Video portal, but for all kinds of content
- Personalized
 - Places you visit a lot
 - Favorite pages
 - Spotlights
- Coming soon

InfoPedia



- Goal:
 - Organize
 - Explain
 - Connect
- Authoritative knowledge structured as your business
- Personalized
 - Office Graph

InfoPedia

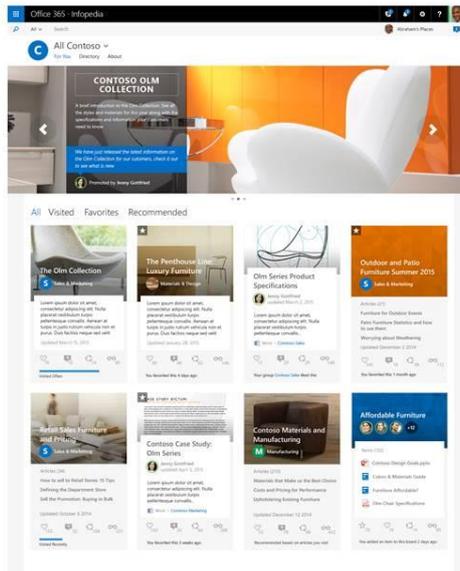



The screenshot displays the Office 365 InfoPedia interface. At the top, there is a search bar with the text "Office 365 - InfoPedia" and a user profile for "All Contoso". Below the search bar, a large banner image shows a modern office chair with the text "CONTOSO OLM COLLECTION". A notification bubble states: "We have just released the latest information on the OLM Collection for our customers, check it out to see what it has!" and is attributed to "Promoted by Jenny Goldberg".

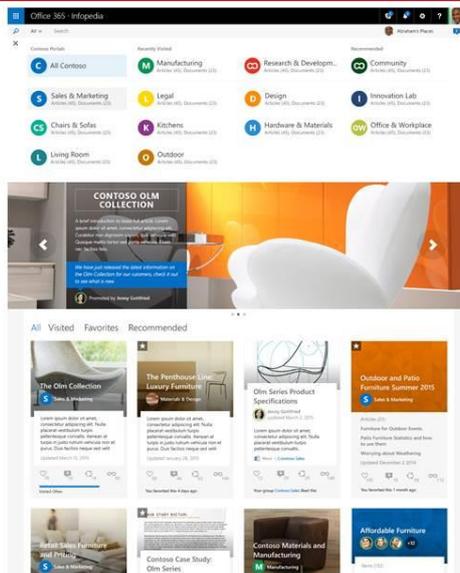
Below the banner, there are tabs for "All", "Visited", "Favorites", and "Recommended". The "Recommended" tab is active, showing a grid of article cards. Each card includes a title, a small image, a brief description, and a "View" button. The articles shown are:

- The OLM Collection**: A brand new collection of smart, contemporary seating with fully physical construction, featuring performance controls. Available at a special introductory price for our customers. See how we made it all work. Updated January 15, 2014.
- The Penthouse Line Luxury Furniture**: A brand new collection of smart, contemporary seating with fully physical construction, featuring performance controls. Available at a special introductory price for our customers. See how we made it all work. Updated January 23, 2014.
- OLM Series Product Specifications**: A brand new collection of smart, contemporary seating with fully physical construction, featuring performance controls. Available at a special introductory price for our customers. See how we made it all work. Updated January 23, 2014.
- Outdoor and Patio Furniture Summer 2015**: A brand new collection of smart, contemporary seating with fully physical construction, featuring performance controls. Available at a special introductory price for our customers. See how we made it all work. Updated December 2, 2014.
- Smart Office Furniture and Printers**: A brand new collection of smart, contemporary seating with fully physical construction, featuring performance controls. Available at a special introductory price for our customers. See how we made it all work. Updated October 6, 2014.
- Contoso Case Study: OLM Series**: A brand new collection of smart, contemporary seating with fully physical construction, featuring performance controls. Available at a special introductory price for our customers. See how we made it all work. Updated April 1, 2015.
- Contoso Materials and Manufacturing**: A brand new collection of smart, contemporary seating with fully physical construction, featuring performance controls. Available at a special introductory price for our customers. See how we made it all work. Updated December 13, 2014.
- Affordable Furniture**: A brand new collection of smart, contemporary seating with fully physical construction, featuring performance controls. Available at a special introductory price for our customers. See how we made it all work. Updated December 2, 2014.

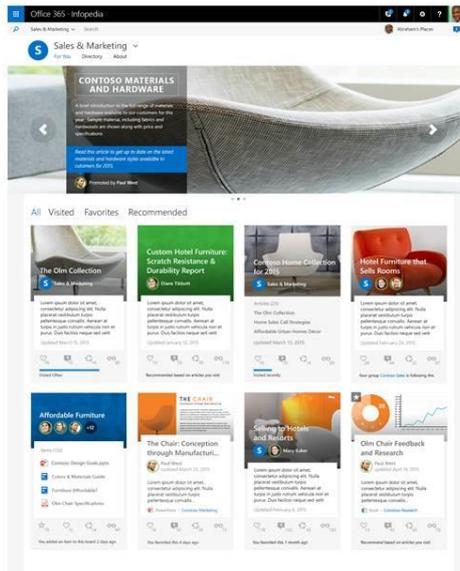
InfoPedia



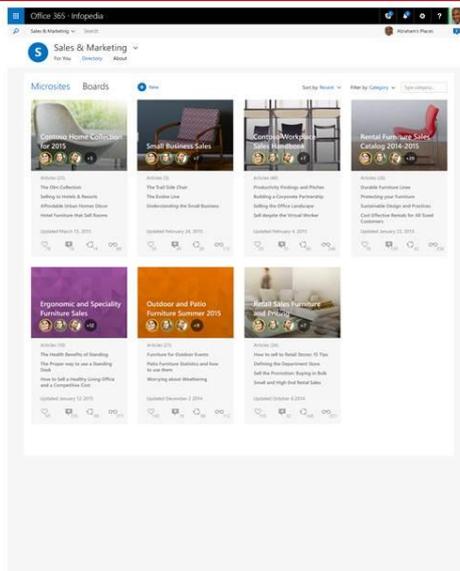
InfoPedia



InfoPedia



InfoPedia



1.4 OFFICE 365 GROUPS

Office 365 Groups



- Bring together resources across Office 365
 - People
 - Information
 - Apps
- Enable better communication and collaboration

Office 365 Groups

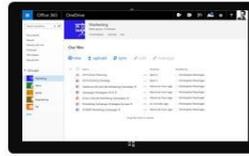


- Key benefits
 - Single definition
 - Public by default
 - Sharing to non-members
 - Self-service
 - Context and history
 - Simple to manage

Office 365 Groups Experiences



Outlook



Files



Notebook



Calendar



People



Skype

Office 365 Groups



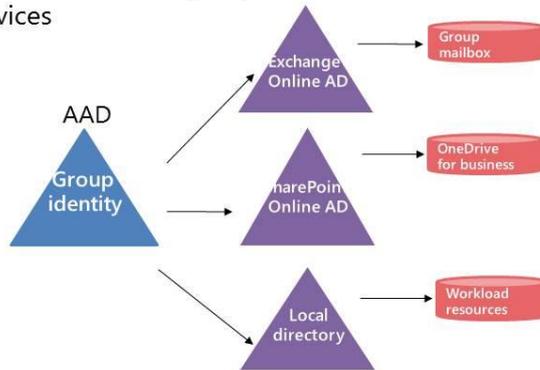
- Building blocks



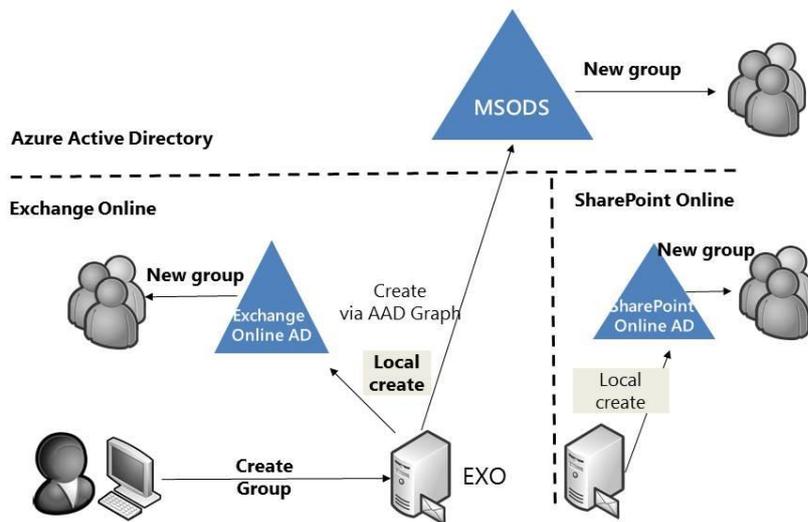
Office 365 Groups Architecture



- One identity
 - AAD is the master for group identity and membership across O365 (Exchange, SharePoint, ...)
- Forward Sync
 - Forwards user and group info from AAD to the individual services



Group Creation Flow



Office 365 Groups U2U

The screenshot shows the Office 365 Outlook interface. The top navigation bar includes 'Office 365', 'Outlook', and the 'Contoso' logo. Below the navigation bar is a search bar for 'Mail and People' and a 'New' button. The main content area is split into two panes. The left pane, titled 'Groups', contains a sidebar with 'Browse groups' and 'Create group' options, both of which are highlighted with a red rectangular box. Below these are 'Joined groups' and a list of groups including 'Customer su', 'Sales', 'Market Tren', 'Moms & Da', and 'Engineering'. The right pane, titled 'Inbox', shows a list of email messages with subjects like 'Are we meeting for lunch', 'Very exciting keynote - loved all the d...', and 'what's up?'. The 'All' filter is selected at the top right of the inbox.

Office 365 Groups: Policy Controls U2U

- Naming conventions
 - Blocked word list
 - Pre-/post-fix
- Group creation permissions
 - Control who is allowed to create a group
 - OWA mailbox policy
 - Set-OWAMailBoxPolicy-Identity-GroupCreationEnabled \$false

The screenshot shows the Office 365 Admin Center interface. The top navigation bar includes 'mailboxes', 'groups', 'resources', 'contacts', 'shared', and 'migration'. The main content area is split into two panes. The left pane shows a table of groups with columns for 'DISPLAY NAME' and 'TYPE'. The right pane, titled 'group naming policy', shows the configuration for the 'general' policy. The 'blocked words' section is visible, showing a list of words that users cannot use in group names. The words 'profanity' and 'badword' are listed in the blocked words section.

1.5 ON-PREMISES: WHAT'S NEXT?

SharePoint Server 2016



- Release date: Second quarter of 2016
- Based on SharePoint Server 2013
- Bring on-prem data into the Office Graph picture and Delve
- Focus on hybrid
 - Hybrid search
 - Hybrid OneDrive
 - Hybrid team sites
 - Hybrid taxonomy
 - Hybrid eDiscovery

Exchange Server 2016



- Release date: Late 2015
- Based on Exchange Server 2013
- CAS is gone
 - Part of Mailbox server role
- Edit documents in OWA (requires OWA farm)
- Better search
- Attachments become links to OneDrive for Business docs

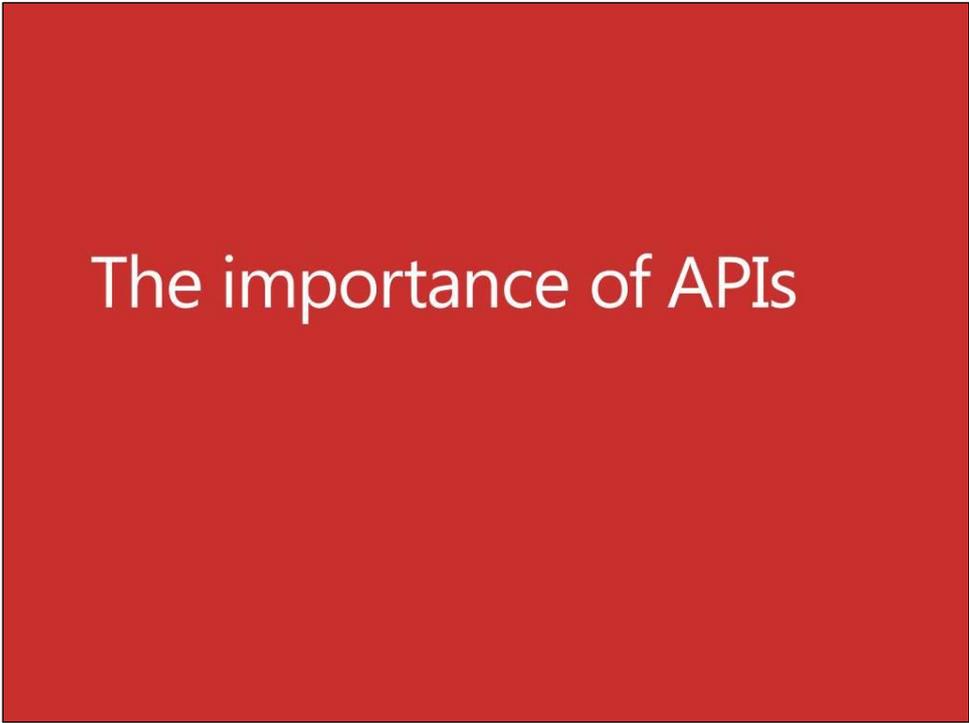
2 DEVELOPING APPS WITH THE NEW APIS OFFERED BY OFFICE 365

Agenda



- The importance of APIs
- SharePoint Online CSOM/REST
- Introducing Office 365 API
- Developing apps consuming Office 365 API
- Registering Office apps in Azure AD
- Office 365 Unified API (preview)
- Migration API
- U2U Site Provisioning

2.1 THE IMPORTANCE OF APIS



The SharePoint & Office 365 App Model 

	Farm Solutions	Sandboxed Solutions	Apps for SharePoint/Office 365
First available	SharePoint 2007	SharePoint 2010	SharePoint 2013/Office 365
Code	Running in the SharePoint process (w3wp.exe)	Running in a separate service on the SharePoint Server	Running on a remote machine
API	SharePoint Server Object Model	SharePoint Server Object Model	Client Side Object Model/REST Services
Capabilities	Full control on SharePoint environment	Limited CAS and API access	
Office 365 & SharePoint Online	Not supported	Coded solutions deprecated Non-coded solutions supported	Supported

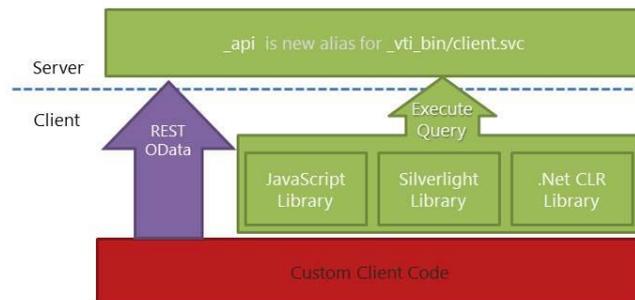
2.2 SHAREPOINT ONLINE CSOM/REST

SharePoint Online CSOM/REST

SharePoint Online CSOM & REST

U2U

- Strict SharePoint API



2.3 OFFICE 365 API

Introducing Office 365 API

Office 365 API



- Set of REST services:
- Microsoft Exchange Online: **Mail, Contacts & Calendars**
- Microsoft OneDrive for Business: **My Files**
- Microsoft Azure Active Directory: **Authentication, Directory Graph**

Office 365 API



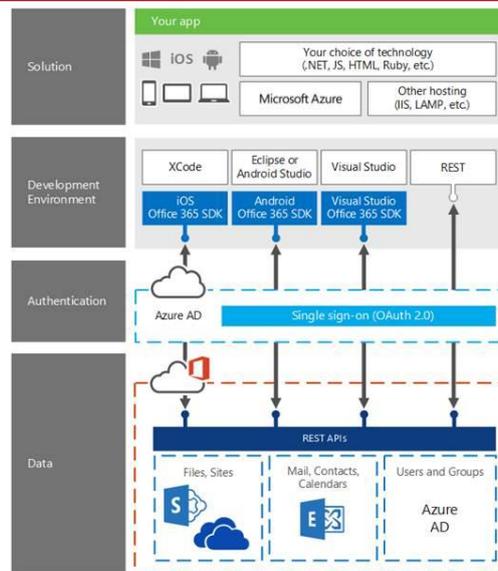

Choice of client and development



- Directly via REST
- .NET Client Library: Windows apps, ASP.NET, WPF, Xamarin...
- JavaScript Client Library
- Open Source SDK for iOS and Android



How does it work?

U2U

Azure Active Directory

U2U

- Applications using Office 365 API need to be registered in Azure Active Directory
- Done manually or from within Visual Studio
- 2 Types of applications can be registered:
 - Web Application (Web API, MVC, Web Forms)
 - Native Client (Mobile, Windows apps, Desktop App)

Demo

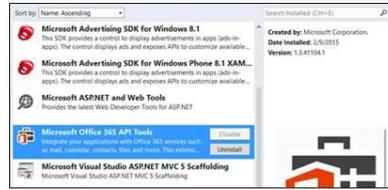
Discovering the Office 365 REST API

Developing apps consuming Office 365
API

Office 365 apps in Visual Studio



- Extensions and Updates:
 - Microsoft Office 365 API Tools (Part of
 - Office Developer Tools)
- NuGet packages:

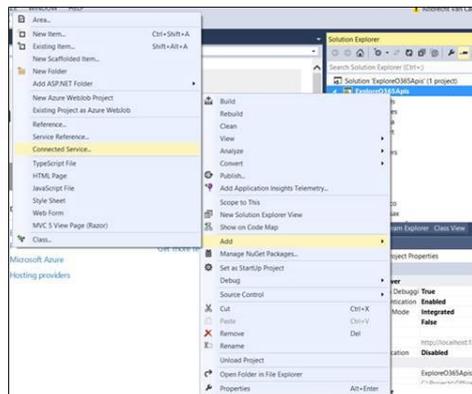


Office 365 Service	Desktop App/ Store App/ ASP.NET App	Xamarin	Cordova
Users and Graphs	Microsoft.Azure.ActiveDirectory.GraphClient		Microsoft.Azure.ActiveDirectory.GraphClientJS
Outlook Services	Microsoft.Office365.OutlookServices		
SharePoint Services	Microsoft.Office365.SharePoint		
Discovery Client	Microsoft.Office365.Discovery		
Any Service		Microsoft.Office365.OAuth.Xamarin	Microsoft.Office365.ClientLib.JS

Office 365 apps in Visual Studio



- Add Office 365 API to your project



Office 365 apps in Visual Studio

U2U

- Configure permissions:



- Visual Studio will automatically configure application in AAD
- Adds nuget packages and configuration settings
- Adds references to project and stores configuration

DEMO

Registering Office 365 app in Visual Studio

Registering applications in Azure AD

Azure Active Directory



- Your apps are registered in Azure AD

NAME	PUBLISHER	TYPE	APP URL
Dynamics CRM	Microsoft Corporation	Web application	http://www.microsoft.com/en-u...
Google	Google	Web application	http://google.com/
Office 365 Exchange Online	Microsoft Corporation	Web application	http://office.microsoft.com/outl...
Office 365 SharePoint Online	Microsoft Corporation	Web application	http://office.microsoft.com/shar...
Office365RESTAPIExplorer.Offi...	U2U Training	Native client application	
ssim-agent0102576a-58e1...	U2U Training	Web application	
ssim-agent2d2668b0-ceb6-44...	U2U Training	Web application	
ssim-agentbe3666f7-adc-453...	U2U Training	Web application	
U2U	U2U Training	Web application	http://www.u2u.be
U2U ASPNET MVC	U2U Training	Web application	http://localhost:27592/
U2U Native Client App	U2U Training	Native client application	

Azure Active Directory



- Specify the service and permissions



- Office 365 Exchange Online service
 - Access to Mail, Calendar, Contacts
- Office 365 SharePoint Online service
 - Access to Files in SharePoint Online or OneDrive for Business

DEMO

Registering Office 365 app in Azure Portal

Outlook Services Client

U2U

- Gives access to **Mail** **Calendar** and **Contacts**



Send email

U2U

```
// Initialize variables
string subject = "Mail sent by using Office 365 APIs";
string recipients = "lieven@u2u365.onmicrosoft.com;els@u2u365.onmicrosoft.com";
string bodyContent = "This email was created from code and was sent using the Office 365 APIs";

// Prepare list of recipients
List<Recipient> toRecipients =
    recipients
        .Split(new char[] { ';' }, StringSplitOptions.RemoveEmptyEntries)
        .Select(
            recipient =>
                new Recipient
                {
                    EmailAddress = new EmailAddress { Address = recipient, Name = recipient }
                })
        .ToList<Recipient>();

// Create draft message
Message draft =
    new Message()
    {
        Subject = subject,
        Body = new ItemBody { ContentType = BodyType.Text, Content = bodyContent },
        ToRecipients = toRecipients
    };
```

Send email



```
// Add the message to the draft folder. This results in a call to the service.
// Returns full item but unfortunately you dont have access to it.
await outlookServicesClient.Me.Folders.GetById("Drafts").Messages.AddMessageAsync(draft);

// Gets the full draft message, including the identifier needed to issue a send mail request.
// This results in a call to the service.
IMessage updatedDraft = await
outlookServicesClient.Me.Folders.GetById("Drafts").Messages.GetById(draft.Id).ExecuteAsync();

// Issues a send command so that the draft mail is sent to the recipient list.
// This results in a call to the service.
await outlookServicesClient.Me.Folders.GetById("Drafts").Messages.GetById(updatedDraft.Id).SendAsync();
```

Contacts



- Get contacts
- Add contact
- Delete contact

```
// Get paged collection of contacts
IPagedCollection<IContact> contactsPage =
await (outlookServicesClient.Me.Contacts.OrderBy(c => c.FileAs)
.Skip((pageNo - 1) * pageSize).Take(pageSize).ExecuteAsync());
```

```
// Create contact
Contact newContact = new Contact();
...
// This results in a call to the service.
await outlookServicesClient.Me.Contacts.AddContactAsync(newContact);
```

```
// Get the contact to delete
var contactToDelete = await
outlookServicesClient.Me.Contacts[contactId].ExecuteAsync();

// Delete the contact
await contactToDelete.DeleteAsync();
```

Events



- Get events

- Add event

```
// Get paged collection of events
IPagedCollection<IEvent> eventsPage =
    await (outlookServicesClient.Me.Calendar.Events.Where(e => e.Start >=
DateTimeOffset.Now.AddDays(-30) && e.Start <= DateTimeOffset.Now.AddDays(30))
    .OrderBy(e => e.Start)
    .Skip((pageNo - 1) * pageSize).Take(pageSize).ExecuteAsync());
```

- Delete Event

```
// Create new event and add
Event newEvent = new Event();
await outlookServicesClient.Me.Events.AddEventAsync(newEvent);
```

```
// Get the event to delete
IEvent eventToDelete = await
outlookServicesClient.Me.Calendar.Events[selectedEventId].ExecuteAsync();

// Delete the event
await eventToDelete.DeleteAsync(false);
```

SharePoint Services Client



- Gives access to Files and OneDrive

```
// Discover if resource is available
CapabilityDiscoveryResult dcr = await
discClient.DiscoverCapabilityAsync("MyFiles");

// Get the SharePointClient: this gives access to OneDrive and Files
return new SharePointClient(dcr.ServiceEndpointUri,
    async () =>
    {
        var authResult =
            await authContext.AcquireTokenSilentAsync(
                dcr.ServiceResourceId,
                new ClientCredential(SettingsHelper.ClientId,
SettingsHelper.AppKey),
                new UserIdentifier(userObjectId, UserIdentifierType.UniqueId));

        return authResult.AccessToken;
    });
```

Get files



```
// Do paging when fetching
int pageNo = 1;
int pageSize = 25;

// Get the SharePoint client
SharePointClient sharePointClient = await
AuthenticationHelper.EnsureSharePointClientCreatedAsync("MyFiles");

// Get the files (and folders)
IPagedCollection<IItem> filesPage =
    await (sharePointClient.Files.Where(i => i.Type == "File"))
        .Skip((pageNo - 1) * pageSize).Take(pageSize).ExecuteAsync();

// Get current page
IReadOnlyList<IItem> fileItems = filesPage.CurrentPage;

// Cast to file objects
IEnumerable<File> files = fileItems.Cast<File>();
```

Add file



```
// Filename + SharePoint Client
string filename = "sample.txt";
var spClient = await
AuthenticationHelper.EnsureSharePointClientCreatedAsync("MyFiles");

// Check if the file exists, delete it if it does
try
{
    IItem item = await spClient.Files.GetByPathAsync(filename);
    await item.DeleteAsync();
}
catch (ODataErrorException)
{
    // fail silently because it doesn't exist.
}

// In this example, we'll create a simple text file and write the current
timestamp into it.
string createdTime = "Created at " + DateTime.Now.ToLocalTime().ToString();
byte[] bytes = Encoding.UTF8.GetBytes(createdTime);

using (MemoryStream stream = new MemoryStream(bytes))
{
    // If file already exists, we'll get an exception.
    File newFile = new File { Name = filename };
    // Create the empty file.
    await spClient.Files.AddItemAsync(newFile);
    // Upload the file contents.
    await spClient.Files.GetById(newFile.Id).ToFile().UploadAsync(stream);
}
```

DEMO

Office 365 API .NET Client Library

2.4 OFFICE 365 APPS VS SHAREPOINT APPS

Office 365 apps vs SharePoint apps

Office 365 apps vs SharePoint apps

- Type of application

Office 365 apps	SharePoint apps
External/standalone app	Integrated in SharePoint



Office 365 apps vs SharePoint apps **U2U**

- Accessibility

Office 365 apps	SharePoint apps
From any link From the Office 365 App Launcher From the Office 365 "My apps" page	from a SharePoint/SharePoint Online site



Office 365 apps vs SharePoint apps **U2U**

- Registration

Office 365 apps	SharePoint apps
Registration in Azure AD	Registration in <code>_layouts/15/appregnew.aspx</code>

NAME	PUBLISHER	TYPE	APP URL
Dynamic CRM	Microsoft Corporation	Web application	http://www.microsoft.com/dcr
Google	Google	Web application	http://google.com/
Office 365 Exchange Online	Microsoft Corporation	Web application	http://office.microsoft.com/outl
Office 365 SharePoint Online	Microsoft Corporation	Web application	http://office.microsoft.com/shar
Office365RESTAPIglossary-OML	U2U Training	Native client application	
https://appregnew.azure-ads-44	U2U Training	Web application	
https://appregnew.azure-ads-44	U2U Training	Web application	
U2U	U2U Training	Web application	http://www.u2u.be
U2U ASP.NET MVC	U2U Training	Web application	http://www.u2u.be/
U2U Native Client App	U2U Training	Native client application	

App Id:

App Secret:

Title:

App Domain:
Example: "www.contoso.com"

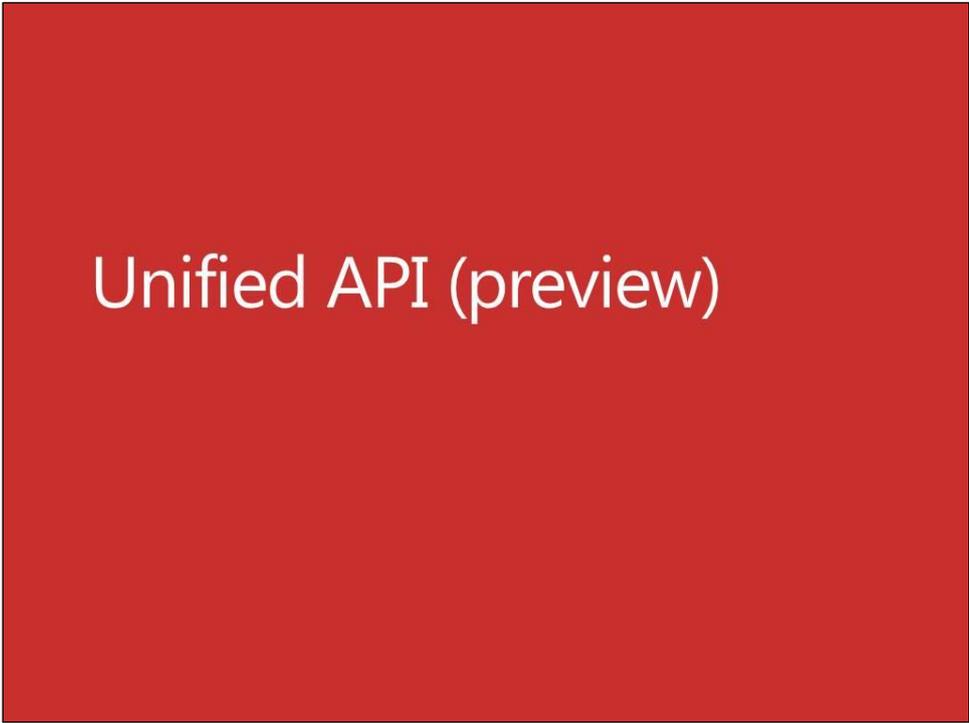
Redirect URL:
Example: "https://www.contoso.com/default.aspx"

Office 365 apps vs SharePoint apps

- Deployment

Office 365 apps	SharePoint apps
Deployed once to the hosting platform Mobile/Web/Desktop	*.app package deployed to app catalog and installed in various SharePoint sites

2.5 UNIFIED API



Data silos in the enterprise **U2U**

USER FILES EMAIL AND CONVERSATIONS EVENTS PEOPLE GROUPS

A diagram illustrating data silos in an enterprise. It features six categories, each with a circular icon: 'USER' (a woman's portrait), 'FILES' (Word, PowerPoint, Word, Excel icons), 'EMAIL AND CONVERSATIONS' (Outlook, Yammer icons), 'EVENTS' (Calendar, Clock icons), 'PEOPLE' (four individual portraits), and 'GROUPS' (a group of three people). The categories are arranged horizontally, with a purple gradient bar at the bottom.

Extended Profile



USER



SharePoint User Profile

https://contoso.sharepoint.com/_api/SP.UserProfiles.PeopleManager/GetMyProperties

Exchange HD Picture

<https://outlook.office365.com/api/v1.0/me/userPhoto>

PEOPLE



GROUPS



Mail, calendar and conversations



USER



Exchange REST API

<https://outlook.office365.com/api/v1.0/me/Messages>

<https://outlook.office365.com/api/v1.0/me/Events>

EMAIL AND
CONVERSATIONS



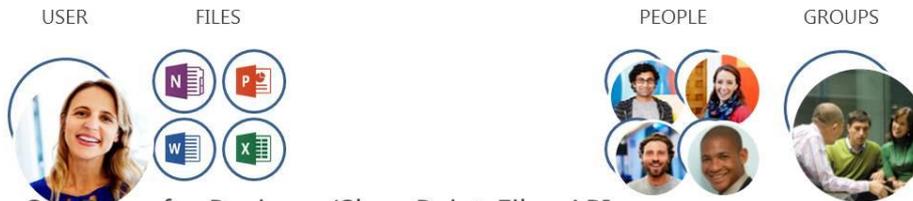
EVENTS



GROUPS



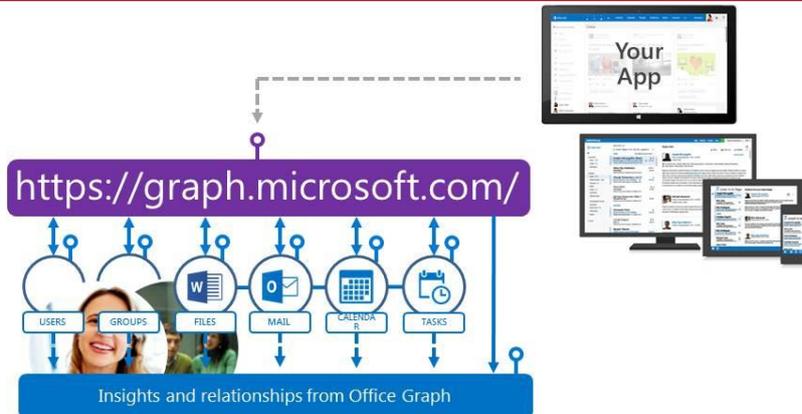
Files



OneDrive for Business/SharePoint Files API

https://contoso-my.sharepoint.com/personal/yina_contoso_com/_api/v1.0/files
https://contoso.sharepoint.com/sites/designCouncil/_api/v1.0/files

Office 365 unified API



Breaking silos



USER FILES EMAIL AND CONVERSATIONS EVENTS PEOPLE GROUPS



A diagram illustrating the concept of 'Breaking silos' in Office 365. It shows six categories: USER, FILES, EMAIL AND CONVERSATIONS, EVENTS, PEOPLE, and GROUPS. Each category is represented by a circular icon or set of icons. The USER category shows a woman's profile. FILES shows icons for Word, PowerPoint, and Excel. EMAIL AND CONVERSATIONS shows icons for Outlook and Yammer. EVENTS shows icons for a calendar and a clock. PEOPLE shows icons for individual profiles. GROUPS shows a group photo. A purple gradient bar is at the bottom.

/me



USER



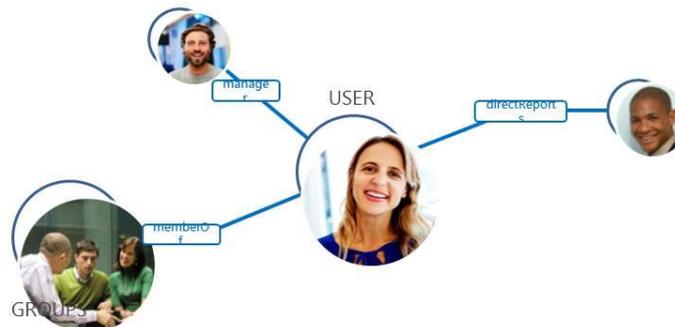
A diagram illustrating the '/me' endpoint in Office 365. It shows a single category: USER, represented by a circular icon of a woman's profile. A purple gradient bar is at the bottom.

Code

U2U

- GET <https://graph.microsoft.com/beta/contoso.com/users/yina@contoso.com>
- GET <https://graph.microsoft.com/beta/me>
- GET <https://graph.microsoft.com/beta/contoso.com/users/<id>>
- GET [https://graph.microsoft.com/beta/contoso.com/users/yina@contoso.com?\\$select=aboutMe](https://graph.microsoft.com/beta/contoso.com/users/yina@contoso.com?$select=aboutMe)
- GET <https://graph.microsoft.com/beta/me/userphotos/96X77>

/users

U2U

Code

U2U

- GET <https://graph.microsoft.com/beta/me/manager>
- GET <https://graph.microsoft.com/beta/contoso.com/users/yina@contoso.com/directReports>
- GET <https://graph.microsoft.com/beta/me/memberOf>
- GET <https://graph.microsoft.com/beta/contoso.com/users/<id>>
- GET <https://graph.microsoft.com/beta/contoso.com/users/alex@contoso.com/directReports>

/messages

U2U

Code

U2U

- GET <https://graph.microsoft.com/beta/me/messages>
- GET [https://graph.microsoft.com/beta/me/messages?\\$top=5](https://graph.microsoft.com/beta/me/messages?$top=5)
- GET [https://graph.microsoft.com/beta/me/messages?\\$top=5&\\$skip=5&\\$orderby=DateTimeCreated](https://graph.microsoft.com/beta/me/messages?$top=5&$skip=5&$orderby=DateTimeCreated)
- GET [https://graph.microsoft.com/beta/me/messages?\\$top=5&\\$select=Subject,Sender&\\$search="from:alex@contoso.com"](https://graph.microsoft.com/beta/me/messages?$top=5&$select=Subject,Sender&$search='from:alex@contoso.com')

/events

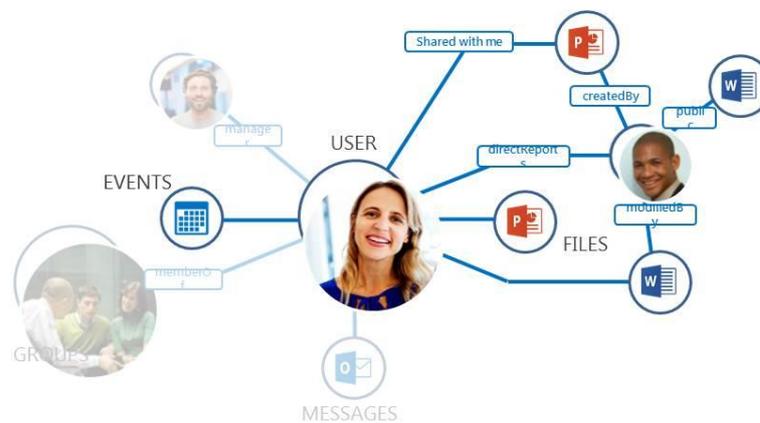
U2U

Code

U2U

- GET <https://graph.microsoft.com/beta/me/events>
- GET [https://graph.microsoft.com/beta/me/events?\\$top=5](https://graph.microsoft.com/beta/me/events?$top=5)
- GET <https://graph.microsoft.com/beta/me>
- [/calendarview?startdatetime=2015-04-01t01:00:00z](https://graph.microsoft.com/beta/me/calendarview?startdatetime=2015-04-01t01:00:00z&enddatetime=2015-04-16t23:00:00z)
- [&enddatetime=2015-04-16t23:00:00z](https://graph.microsoft.com/beta/me/calendarview?startdatetime=2015-04-16t23:00:00z)

/files

U2U

Code



- GET <https://graph.microsoft.com/beta/me/files>
- GET <https://graph.microsoft.com/beta/me/files/<id>>
- GET <https://graph.microsoft.com/beta/me/files/<id>/lastModifiedByUser>
- GET <https://graph.microsoft.com/beta/mary@contoso.com/files>
- GET <https://graph.microsoft.com/beta/mary@contoso.com/files/<id>/lastModifiedByUser/manager>

DEMO

Unified API

2.6 SHAREPOINT ONLINE MIGRATION API

Migration API

Why use The API

U2U

Resources dedicated to ISV and IT admins



Limited calls to end user entry points. Meaning, this won't get impacted by the CSOM throttling.

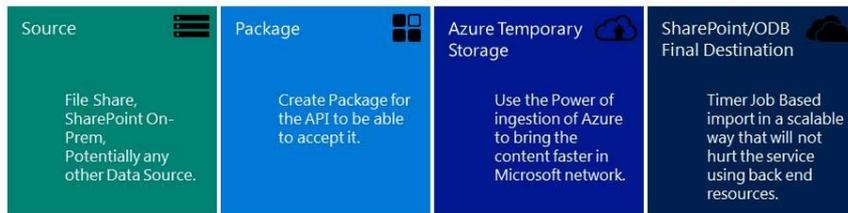
Better equipped to scale to the demand



Quick Overview

U2U

HTML



What about Speed?

U2U

The Type of content does impact the rate of ingestion
Using Backend Resources
Lots of small, scenario specific, tweaking that can help get the best out of the API

Preliminary data suggest 5X the speed of CSOM before throttling.



The Flow

U2U

HTML

1. The package is created
2. The package gets uploaded to Azure Blob storage
3. 1 CSOM call is made to start the migration process
4. The Azure Queue gets real time updates
5. Once Complete The logs in the package get updated



Create the Package

U2U

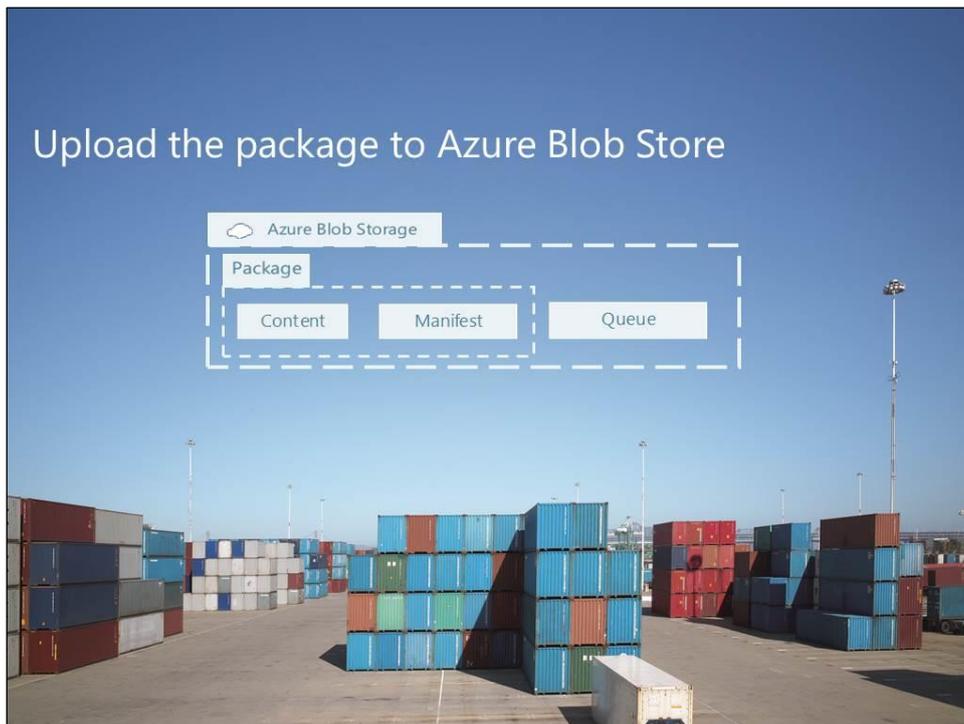
HTML

Generating the Appropriate XML to go with the Files

Resemble a lot to the PRIME package

8 xml in a package + the Content





One CSOM request to start the migration

- [ClientNS.ClientCallableMethod]
- public Guid CreateMigrationJob(
 - Guid gWebId,
 - string azureContainerSourceUri,
 - string azureContainerManifestUri,
 - string azureQueueReportUri)

SharePoint Online Management Shell

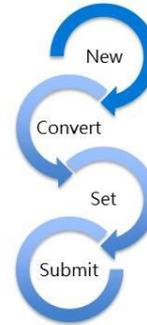
The commands

 New-SPOMigrationPackage

 ConvertTo-SPOMigrationTargetedPackage

 Set-SPOMigrationPackageAzureSource

 Submit-SPOMigrationJob



Thank you!