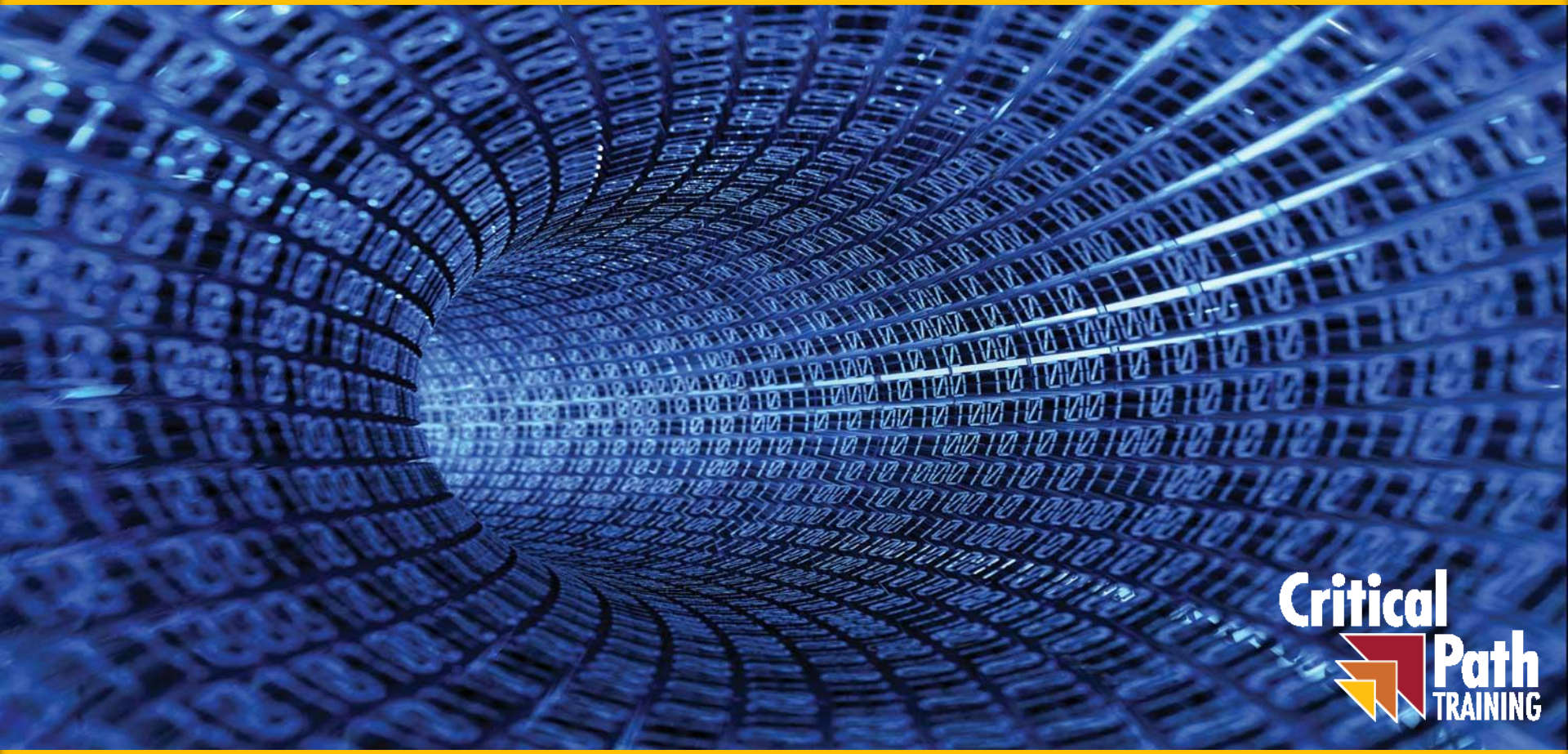
Getting Started with SharePoint 2016



Student Introductions

- Basic Info
  - What's your name?
  - Where do you work? (optional)
  - How long have you been a developer?
  - Have you used SharePoint? Which versions?

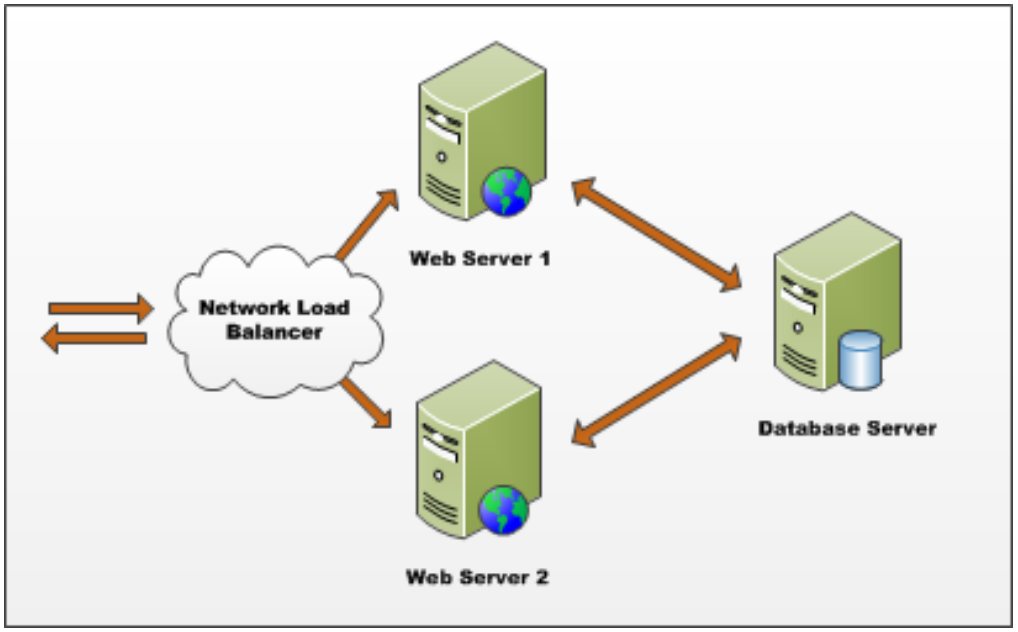
- List the skills with which you already feel comfortable
  - .NET programming in Visual Studio with C# or VB.NET
  - Development with ASP.NET and ASP.NET MVC
  - SharePoint solution development
  - JavaScript and jQuery
  - REST and OData

Agenda

- SharePoint Architecture and Topology
- SharePoint Development Strategies
- SharePoint Developer Tools and Utilities
- Creating a SharePoint Development Environment

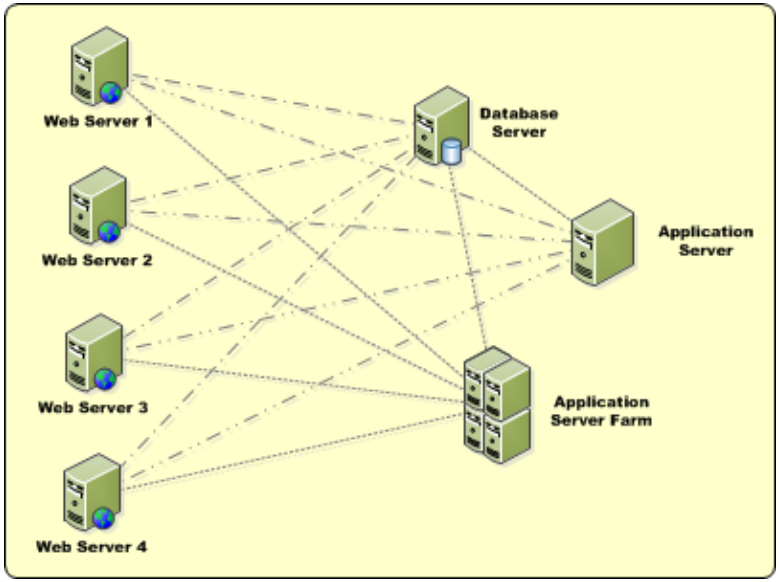
SharePoint Farms

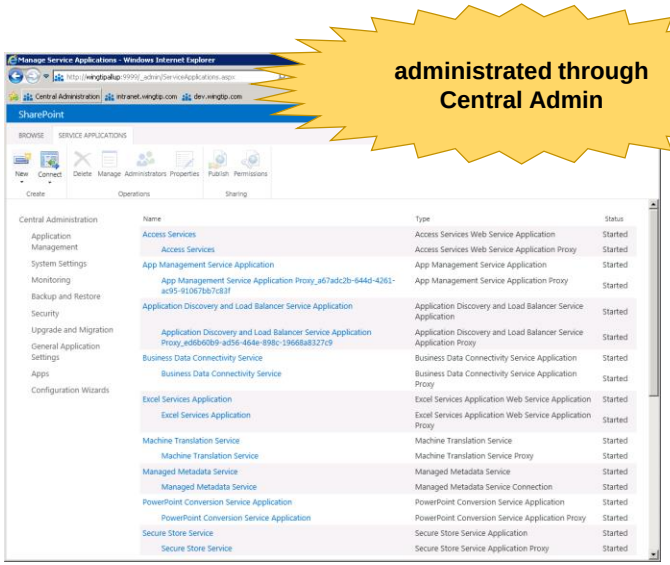
- SharePoint farms created for on-premises deployments
  - Farm requires Web server(s) and database server
  - Farm can be single server or multi-server
  - Each farm has exactly one configuration database
  - Single-server farm used for development environments



Service Applications

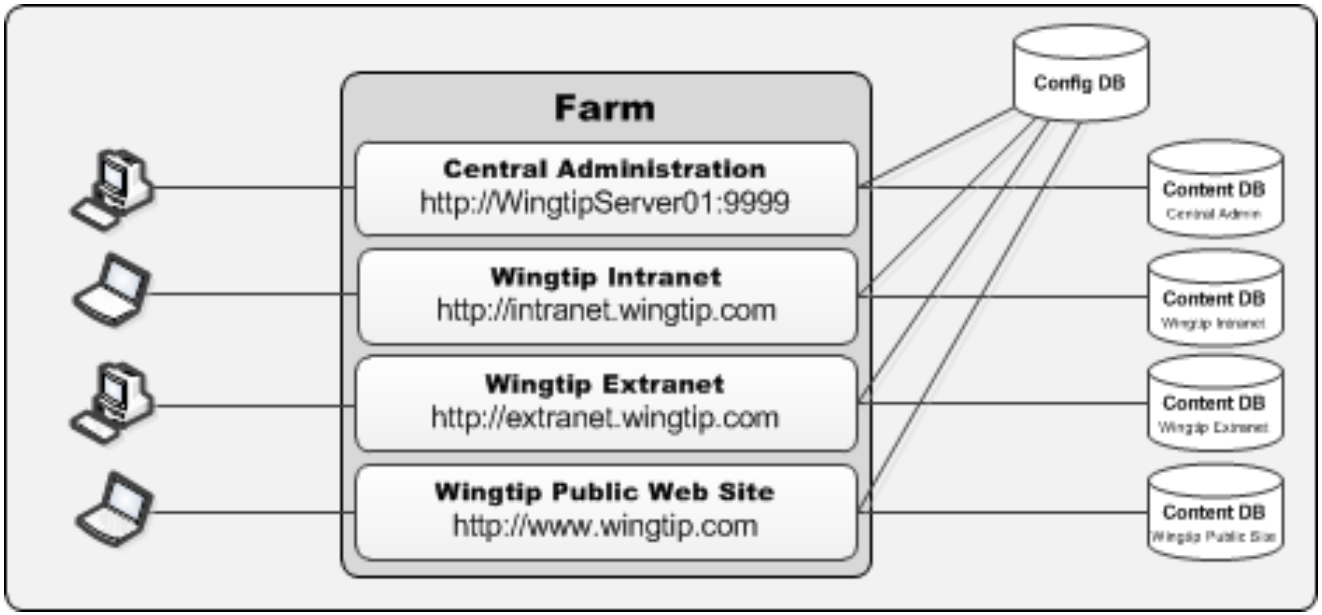
- Services applications facilitate resource sharing
- Service apps can run on WFE or Application Servers
- Service apps can be used across farms





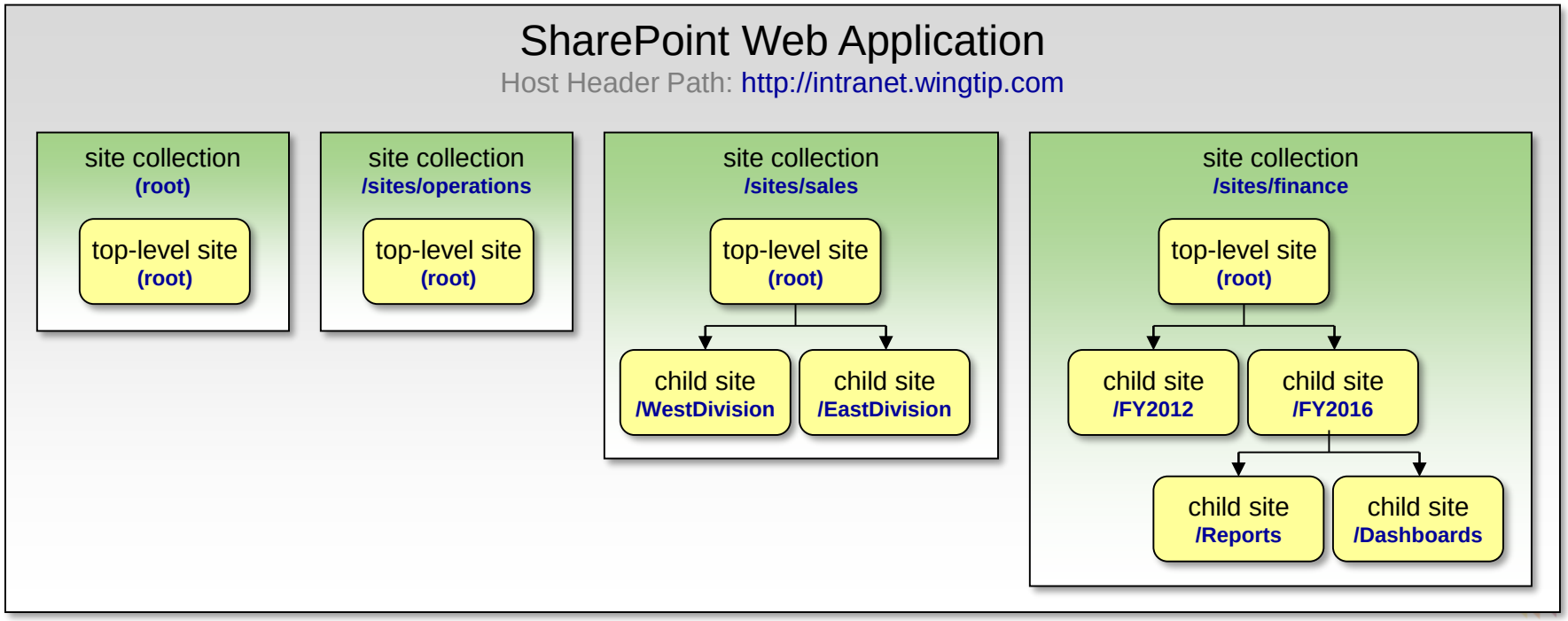
Web Applications

- Web Applications provide HTTP entry points
  - Web Applications based on IIS Web sites
  - Web Application defines one or more URL spaces
  - Web Application security configured independently



Site Collections and Sites

- Sites always created in scope of a site collection
  - Site collections created at web application scope in on-prem farm
  - Site collections created at tenancy scope in SharePoint on-line
  - User can be configured to be site collection administrator

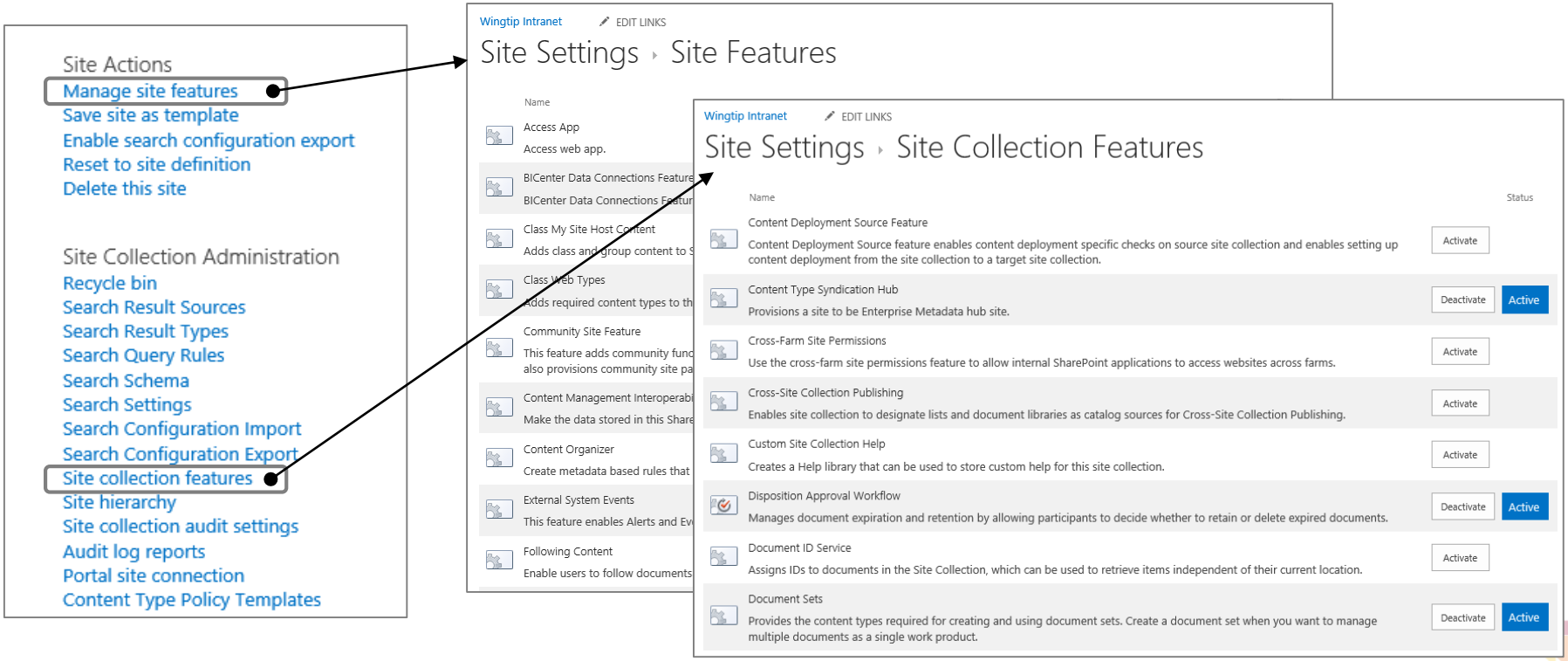


Managing SharePoint

- Central Administration
  - Available in On-Premises deployments
  - Manage servers, services, jobs, etc.
  - Create Web Applications, site collections
- Site Settings
  - Available in On-Premises & Hosted deployments
  - Manage site features, lists, users, permissions, etc.
  - Dual-purpose management site for sites & site collections
    - Manage site collection from top-level site's site settings page
    - When in a non-top-level site, only managing current site

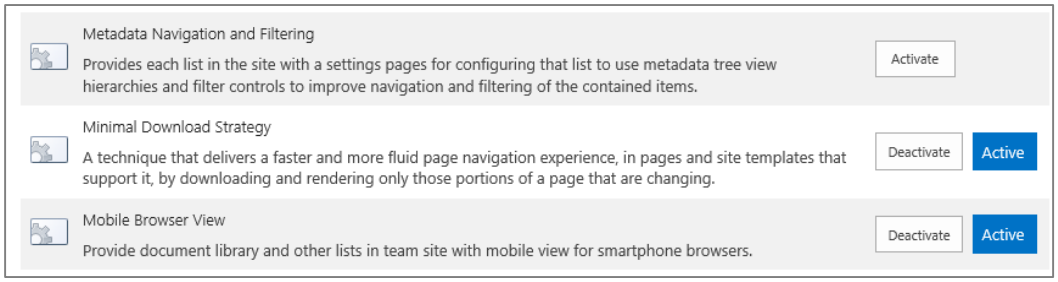
Managing Features

- Site collection administrator can activate features
  - Some features activate at site (aka web) level
  - Other features activate at site collection level

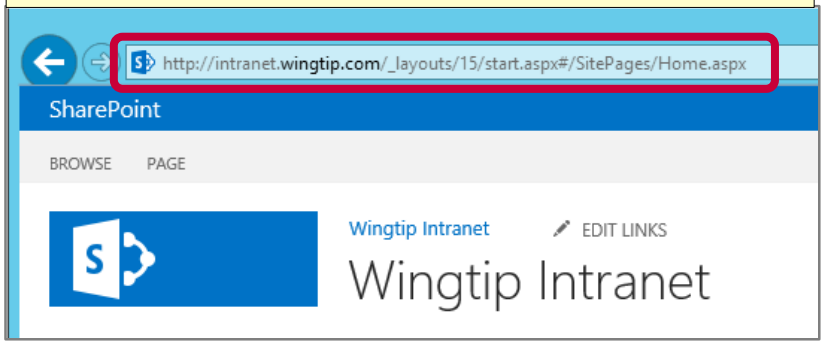


Minimal Download Strategy (MDS) Feature

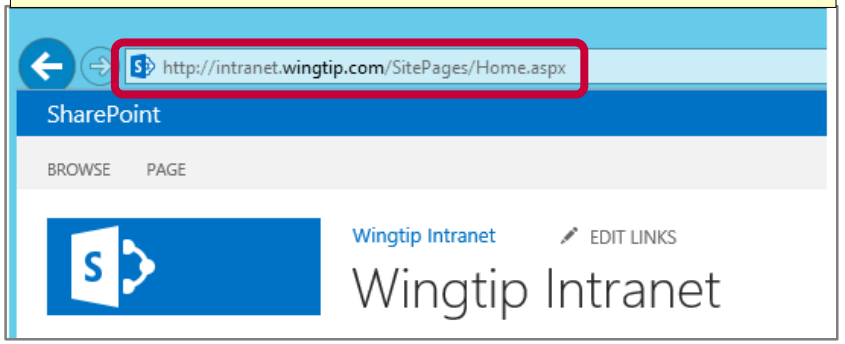
- MDS features used to smooth page transitions
  - Implemented with site-scoped feature
  - MDS features is activated in Team Site by default
  - MDS feature is disabled (and not supported) in Publishing Sites



URL structure to site page with MDS feature activated



URL structure to site page with MDS feature deactivated




DEMO
Creating a Team Site using Central Administration

Agenda

- ✓ SharePoint Architecture and Topology
- ➢ SharePoint Development Strategies
- SharePoint Developer Tools and Utilities
- Creating a SharePoint Development Environment

SharePoint Environments

- SharePoint On-Premises Farms
  - SharePoint installed and managed by company
  - Access to 100% of SharePoint's features & capabilities
- Office 365 and SharePoint Online
  - SharePoint installed and managed by Microsoft in cloud
  - Some on-premises features not available in the cloud
- Hybrid Environments
  - Mix of the two other environments
  - Very scenario driven on customer-by-customer basis

SharePoint Development Strategies

- Farm Solutions (aka Full Trust Solutions)
  - Packaged and deployed using farm solution packages
  - Deployment requires farm administrator
  - Works in on-premises farms but not Office 365
  - Heavily used since SharePoint 2007
- Sandboxed Solutions
  - Introduced in SharePoint 2010 with very limited adoption
  - Deprecated by Microsoft in SharePoint 2013
- SharePoint ~~Apps~~ Add-ins
  - Introduced with SharePoint 2013
  - Designed for Office 365 and on-premises farms
  - Does not allow server-side code to run in SharePoint
  - Requires breadth of client-side development skills

SharePoint Server-Side Object Model

- Accessible through **Microsoft.SharePoint.dll**
  - In-process Assembly DLL for ..NET clients
  - Oldest & most mature API for SharePoint
  - Available in solution packages but not SharePoint apps
  - Farm solutions have full access to server-side API
  - Sandbox solutions can access only a subset

Client-Side Object Model (CSOM)

- CSOM provides client-side API for SharePoint
  - Introduced in SharePoint 2010
  - Accessible using .NET, Silverlight and JavaScript
- CSOM expanded in SharePoint 2013
  - Search
  - Managed Metadata
  - User Profiles and Social Feeds
  - Business Connectivity Service (BCS)
  - Workflow
  - Publishing

SharePoint REST API

- SharePoint 2016 provides REST API
  - Great alternative to CSOM when coding in JavaScript
  - Can also be called from server-side .NET code
  - Accessible from non-Windows platforms as well
  - Unlike CSOM, SharePoint REST API accessible to all

- Creating URIs for the SharePoint REST API
  - URIs created based on principles of REST and ODATA
  - [Target Site URL] + _api + [Target SharePoint object]

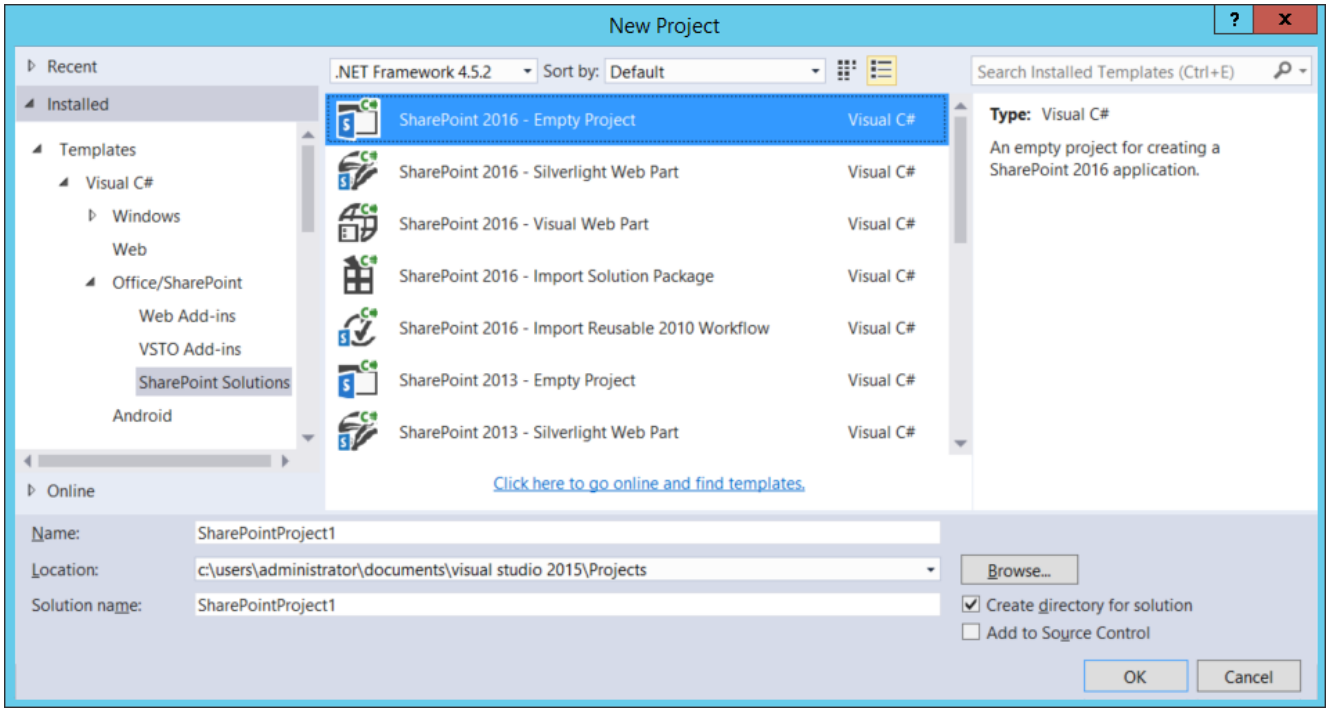
```
http://intranet.wingtip.com/_api/web/lists/getByTitle('Customers')
```

Agenda

- ✓ SharePoint Architecture and Topology
- ✓ SharePoint Development Strategies
- ➢ SharePoint Developer Tools and Utilities
- Creating a SharePoint Development Environment

Visual Studio 2015

- Visual Studio supports SharePoint development
  - Project templates for SharePoint solutions and apps
  - Use Visual Studio 2015 with Update 2 or later
  - Install latest Visual Studio updates for SharePoint 2016



Web Essentials

- Web Essentials 2015
  - Visual Studio 2015 Extension in Online Gallery
  - Additional IntelliSense for CSS3
  - Warnings & helpers for browser compatibility issues
  - Selector IntelliSense for HTML elements, classes, IDs
  - Web Essentials includes **JSHint**
  - Includes JSHint which detects problems in JavaScript

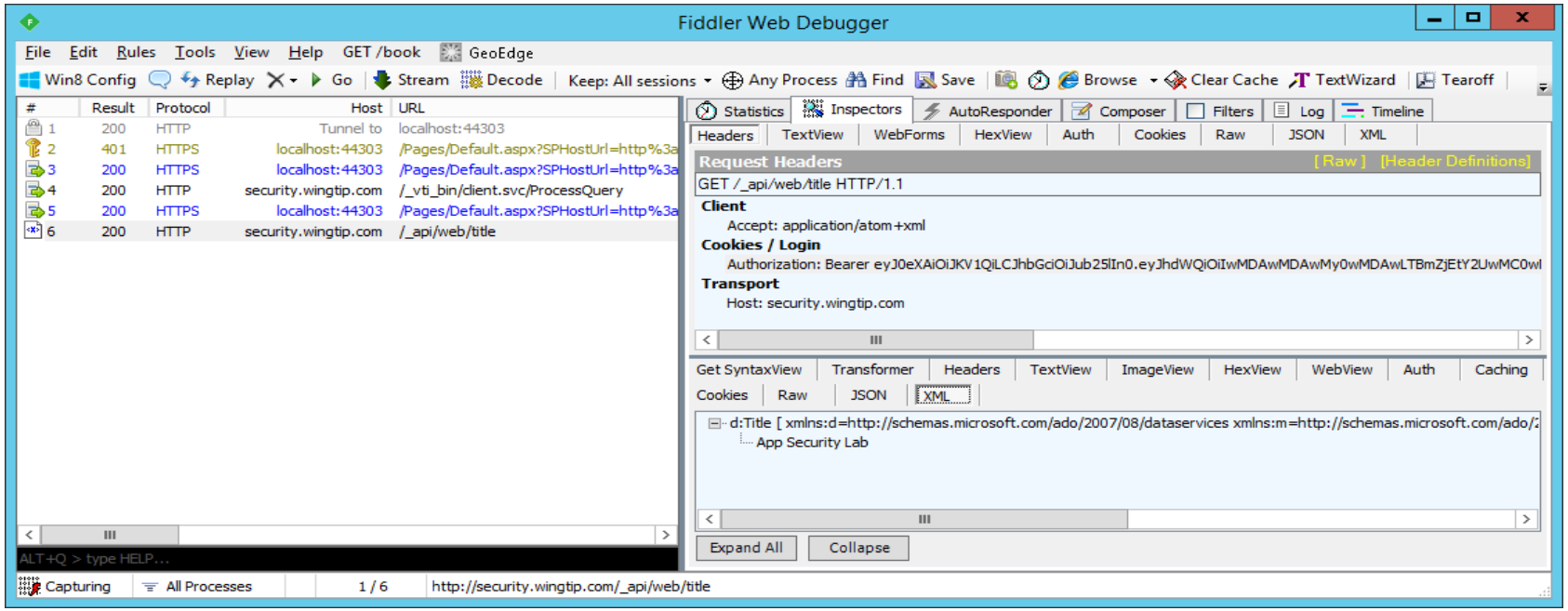
Developer Tools provided by the Browser

- Browser-based development tools
  - View HTML
  - View & modify CSS
  - View & debug JavaScript

- **Internet Explorer**
  - Developer Tools: included
- **FireFox**
  - FireBug: extra download
- **Google Chrome**
  - Developer Tools: included

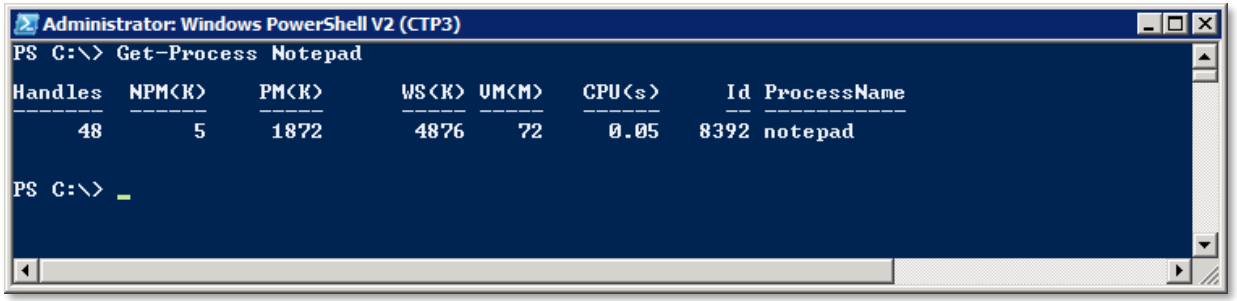
Debugging HTTP request with Fiddler

- Fiddler is a HTTP debugging proxy
  - It helps you inspect HTTP request & response
  - Useful in debugging client-side JavaScript code
  - Useful in debugging SharePoint Workflows



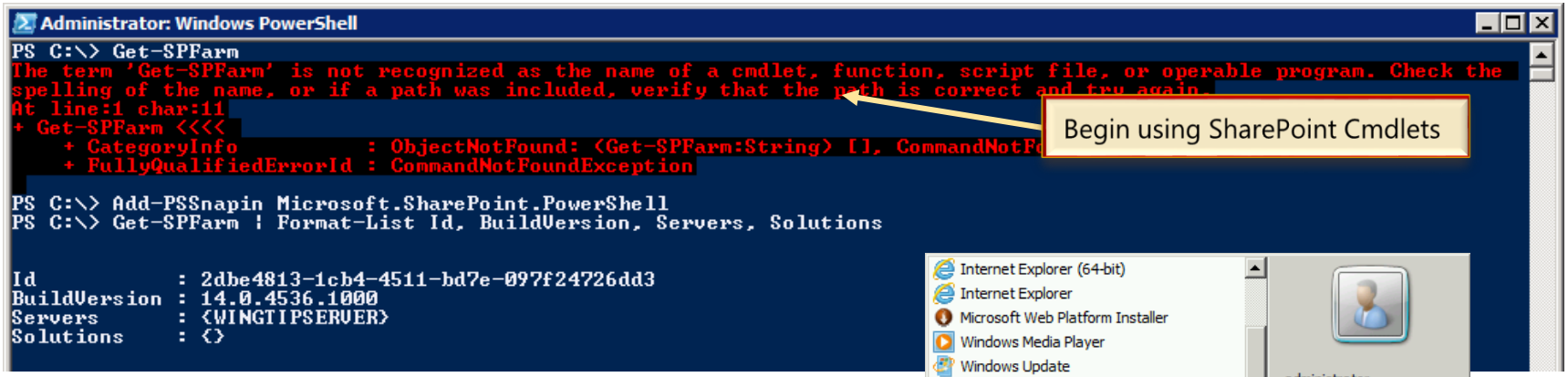
Working with Windows PowerShell

- SharePoint developers should learn PowerShell
  - Scripting environment for SharePoint administration
  - Used to create and manage test sites
  - Used to configure SharePoint environment
- Windows PowerShell fundamentals
  - Cmdlets (e.g. `Get-Process` and `Stop-Process`)
  - Pipelining and formatting features
  - Provider-based model for accessing resources

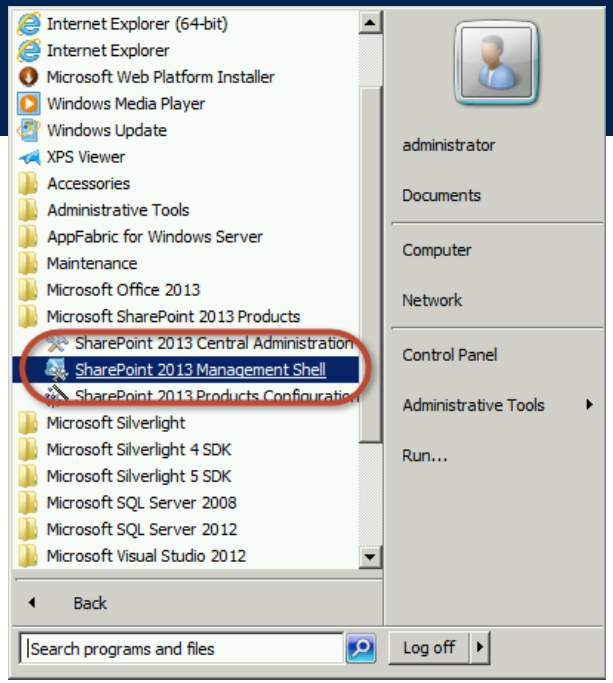


The Microsoft.SharePoint.Powershell Snapin

- Explicitly load SharePoint Windows PowerShell snap-in from console or script

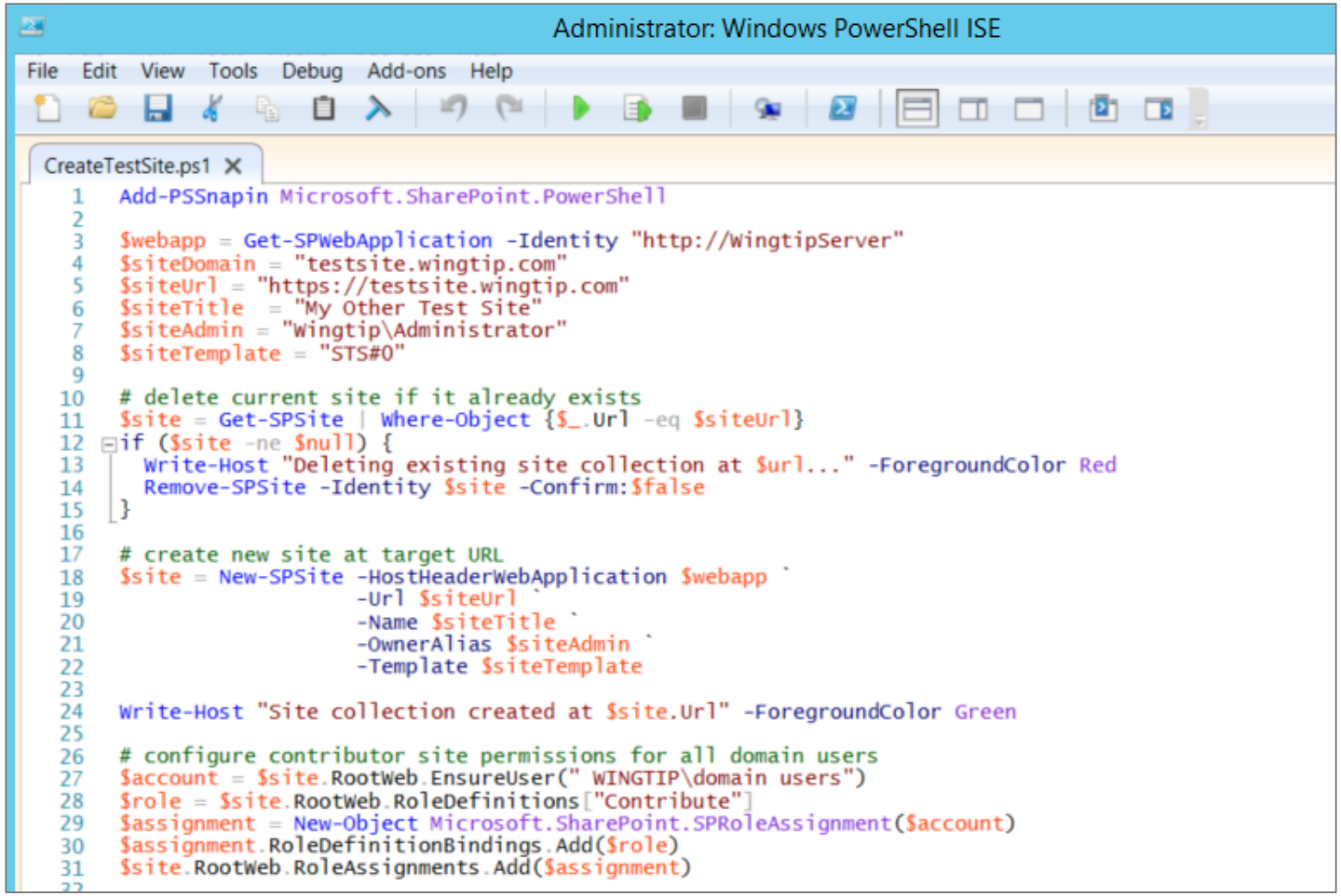


- Implicitly load snap-in by using link in Windows Start menu



Windows PowerShell ISE

- Supports color-coding, IntelliSense and debugging

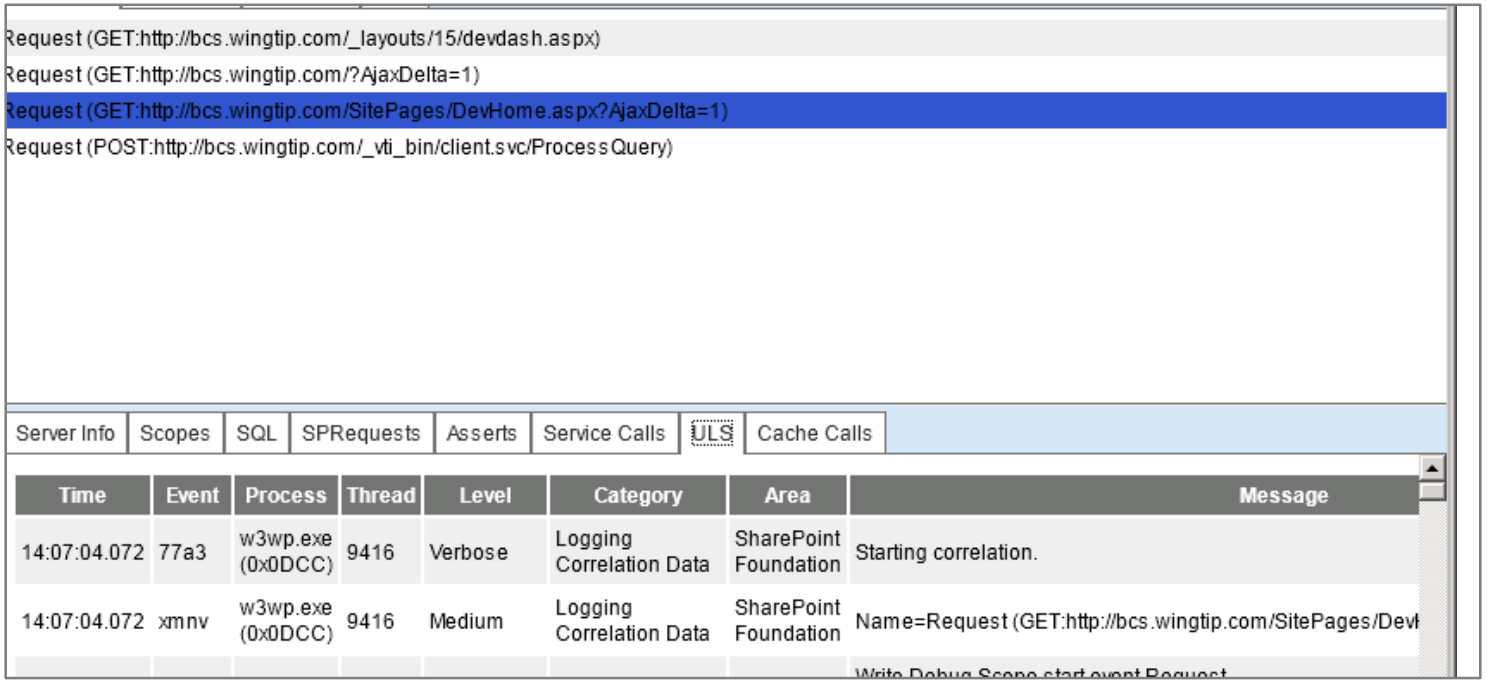


Troubleshooting Errors with ULS

- ULS is Unified Logging Service
  - SharePoint's log files located at `..\15\LOGS`
  - Configure level of logging for different categories:
    - Central Administration → Monitoring → Configure Diagnostic Logging
- Developer Tools for inspecting ULS logs
  - Merge-SPLogFile cmdlet in PowerShell
  - ULS Log Reader Utility (ULSViewer.exe)
  - Developer Dashboard

Developer Dashboard

- A utility for inspecting per-request diagnostics
  - Introduced in SharePoint 2010
  - Much improved in SharePoint 2013
  - Shows requests from start of dashboard session



Enabling & Using Developer Dashboard

- Developer Dashboard must be enabled
  - Typically enabled using PowerShell script

ToggleDeveloperDashboard.ps1

```
Add-PSSnapin Microsoft.SharePoint.PowerShell

$contentService = [Microsoft.SharePoint.Administration.SPWebService]::ContentService
$dashboardSettings = $contentService.DeveloperDashboardSettings

if ($dashboardSettings.DisplayLevel -eq "OnDemand"){
   # developer dashboard is on - turn it off
   $dashboardSettings.DisplayLevel = "Off"
   $dashboardSettings.Update()
   Write-Host "Developer dashboard disabled." -ForegroundColor Gray
} else {
   # developer dashboard is off - turn it on
   $dashboardSettings.DisplayLevel = "OnDemand"
   $dashboardSettings.Update()
   Write-Host "Developer dashboard enabled." -ForegroundColor Gray
}
```

Agenda

- ✓ SharePoint Architecture and Topology
- ✓ SharePoint Development Strategies
- ✓ SharePoint Developer Tools and Utilities
- ➢ Creating a SharePoint Development Environment

The SharePoint 2016 VM Setup Guide

- Use the SharePoint 2016 VM Setup Guide
  - https://github.com/CriticalPathTraining/SharePoint2016VmSetupGuide
- Guide has docs and scripts to build student VM
  - Windows server 2012 R2
  - Active Directory Domain Services
  - SQL Server 2016
  - SharePoint Server 2016
  - Support for SharePoint 2013 Workflows
  - SharePoint Designer 2013
  - Visual Studio 2015 with SharePoint 2016 Support

Student files for GSA2016

- Student files for GSA2016 are kept in GitHub
  - https://github.com/CriticalPathTraining/GSA2016

Summary

- ✓ SharePoint Architecture and Topology
- ✓ SharePoint Development Strategies
- ✓ SharePoint Developer Tools and Utilities
- ✓ Creating a SharePoint Development Environment