



Д. Г. Ермаков
Н. И. Тебайкина

Использование веб-служб MS SharePoint

Электронный текстовый ресурс

Для студентов бакалавриата всех форм обучения направления
38.03.05 «Бизнес-информатика»

Содержит теоретический и справочный материал для выполнения самостоятельной работы студентов, подготовки к промежуточной аттестации. Примеры программирования могут быть использованы в качестве отдельных учебных задач в процессе выполнения лабораторных работ

Подготовлено кафедрой «Анализ систем и принятия решений»

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
КРАТКИЕ СВЕДЕНИЯ О СИСТЕМЕ MS SHAREPOINT.....	5
SHAREPOINT 1.0 WAVE.....	5
SHAREPOINT 2010	5
SHAREPOINT 2013	10
SHAREPOINT 2016	12
SHAREPOINT 2019	20
ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ ПО СОДЕРЖИМОМУ РАЗДЕЛА «КРАТКИЕ СВЕДЕНИЯ О СИСТЕМЕ MS SHAREPOINT»	33
MICROSOFT POWERSHELL.....	34
ОБЩИЕ СВЕДЕНИЯ	34
PNP POWERSHELL OVERVIEW.....	38
АКТУАЛЬНЫЕ КОМАНДЛЕТЫ.....	39
СОЗДАНИЕ СПИСКОВ НА САЙТЕ SHAREPOINT	84
СКРИПТ СОЗДАНИЯ САЙТОВ И УЧЕТНЫХ ЗАПИСЕЙ СТУДЕНТОВ	97
ПОЛЕЗНЫЕ СКРИПТЫ ДЛЯ SHAREPOINT НА POWERSHELL.....	109
ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ ПО СОДЕРЖИМОМУ РАЗДЕЛА «MICROSOFT POWERSHELL»	123
ПРИЛОЖЕНИЕ 1.....	124
ПРИЛОЖЕНИЕ 2.....	129
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	130

ВВЕДЕНИЕ

В зависимости от назначения информационной системы было принято классифицировать тот или иной программный продукт. Возникли классы специализированных систем, например:

- системы электронного документооборота;
- системы управления проектами;
- системы управления персоналом;
- системы автоматизированного проектирования;
- системы управления ресурсами;
- системы управления содержимым и многие другие.

Но в ряде случаев, особенно в крупных и средних компаниях, существует потребность применения одновременно программных продуктов различных классов, в то же время нет необходимости использования дорогих специализированных систем с избыточной для данной компании функциональностью, тогда целесообразно применение коллекции программ и информационных технологий, объединенных единой платформой. К таким платформам относят систему SharePoint, созданную IT-компанией Microsoft. SharePoint также называют средой совместной работы пользователей.

Более 200000 организаций и 190 млн. человек используют интрасети, сайты групп и средства управления контентом SharePoint¹.

Так, например, о назначении SharePoint версии 2013 известная IT-компания на своем портале отзывається следующим образом: Microsoft SharePoint Server 2013 представляет собой интегрированный пакет корпоративных приложений, который предназначен для увеличения производительности труда, организации совместной работы сотрудников, решения таких важных бизнес-задач, как контроль информационных потоков, принятие взвешенных решений и управление рабочими процессами. Продукт Microsoft SharePoint Server 2013 на базе единой интегрированной платформы поддерживает выполнение корпоративных приложений в интранете, экстранете и в Интернете, что позволяет компаниям не зависеть от отдельных раздробленных систем. Встроенные функции, дополненные технологиями поиска, помогают реагировать на меняющиеся

¹ URL: <https://products.office.com/compare-all-microsoft-office-products?tab=2>

требования бизнеса. Microsoft SharePoint Server 2013 упрощает принятие решений, опирающихся на результаты анализа данных, оптимизирует внедрение комплексов, настраиваемых под индивидуальные потребности компании, как в масштабах всего предприятия, так и вне его. В версии Microsoft SharePoint 2013 сделан акцент на социальную составляющую, облака и мобильность. Microsoft SharePoint 2013 предлагает новые средства простого администрирования, эффективной защиты коммуникации и информации и гибкой совместной работы. Социальные возможности позволяют легко обмениваться идеями, отслеживать действия коллег, отыскивать экспертов и информацию и т. п.²

MS SharePoint считается наиболее экономичным, обладающим всеми необходимыми функциями достаточного качества, в классе аналогичных программных продуктов.

В электронном текстовом ресурсе приводятся основные сведения о программной среде MS SharePoint 2013, 2016, 2019. Как инструмент рассматривается приложение PowerShell.

Данный материал призван ознакомить бакалавров направления 38.03.05 «Бизнес-информатика» с технологией программирования на основе использования WEB-служб в рамках дисциплины по выбору студента «Основы работы в MS SharePoint», предназначен для выполнения самостоятельной работы студентов, подготовки к промежуточной аттестации. Примеры использования командлетов могут быть использованы при выполнении отдельных задач лабораторных работ. Электронный текстовый ресурс может быть полезен и для магистрантов направления «Бизнес-информатика».

² URL: <http://www.interface.ru/home.asp?artId=33974>

КРАТКИЕ СВЕДЕНИЯ О СИСТЕМЕ MS SHAREPOINT

SharePoint 1.0 Wave

Первая коммерческая версия системы – это так называемая первая волна создания платформы, SharePoint 1.0 Wave (2001 г.), затем была создана система MS SharePoint 2003, состоящая из двух инструментов: Windows SharePoint Services (WSS) 2.0 – продукт для обеспечения командной работы с ограниченными возможностями управления документами и SharePoint Portal Server 2003 – порталный продукт, рассматривающийся как расширение функции общедоступных папок. Оба продукта второй волны основаны на механизме баз данных SQL Server³.

В 2006 году была выпущена следующая версия системы, а именно SharePoint 2007, состав: Microsoft Office SharePoint Server 2007 и Windows SharePoint Services 3.0. В этой версии уже было действительно исчерпывающее управление документами, поиск по предприятию, обеспечение совместной работы, и многие организации развернули ее для повышения эффективности и возможностей взаимодействия⁴. Недостаток системы – не хватало возможностей общего управления средами SharePoint, что приводило к порождению излишних данных.

SharePoint 2010

В начале 2010 года выпущена версия SharePoint 2010, в составе которой были SharePoint Foundation Server (в предыдущих версиях это была служба сервисов в предыдущих версиях) и Microsoft SharePoint Server 2010. Присутствовала тесная интеграция с линейкой продуктов Office 2010. Появилось много новых возможностей, например, списки SharePoint хранили не файлы, а данные, их можно использовать многими способами, список представляет собой автономную базу данных. Осуществлялось управление метаданными, которые встречаются в документах SharePoint. Появились сервисы для бизнес-аналитики Business

³ Ноэл М., Спенс К. Microsoft SharePoint 2010. Полное руководство, 2012, с.33

⁴ То же, с.35

Intelligence. На рисунке 1 отображена модель платформы SharePoint 2010.

На рисунке 2 представлен укрупненно функциональный состав платформы SharePoint.

Состав лицензий SharePoint 2010 представлен на рисунке 3.

Были задействованы следующие сценарии использования Microsoft SharePoint 2010:⁵

- корпоративный портал (информация о достижениях компании; информация о сотрудниках, например, фото, интересы, теги и др.; данные, необходимые для работы сотрудников; инструменты для общения сотрудников такие, как форумы, блоги; публикация новостей, быстрое создание форумов и опросов; интерактивная оргструктура; интегрированные источники информации – библиотеки фото- и видеоконтента; легкое редактирование сайтов, включая темы и оформление с помощью интерфейса «лента»; система подписки и оповещений о появлении новых документов; календари, синхронизируемые с Outlook; одновременное ведение нескольких проектов в компании – рабочие области проектов; у каждого сотрудника своя рабочая веб-область; у каждого департамента своя страница);

- бизнес-анализ (личный кабинет директора: все данные для принятия решений на одной странице; актуальность информации в любой момент времени; точность и скорость сбора данных от региональных офисов; отслеживание зависимости разных показателей, в том числе трендов увеличения или снижения продаж; выгрузка данных, быстрое построение отчетов в MS Excel; индикаторы продаж по сотрудникам, публикация на портале с помощью MS Visio; работа с отчетами в браузере; изменение параметров отчетов «на лету»; интерактивные индикаторы);

- архив документов;

⁵ Стаханов Е. Краткий обзор возможностей MS SharePoint (компания Terrelink) стр.6

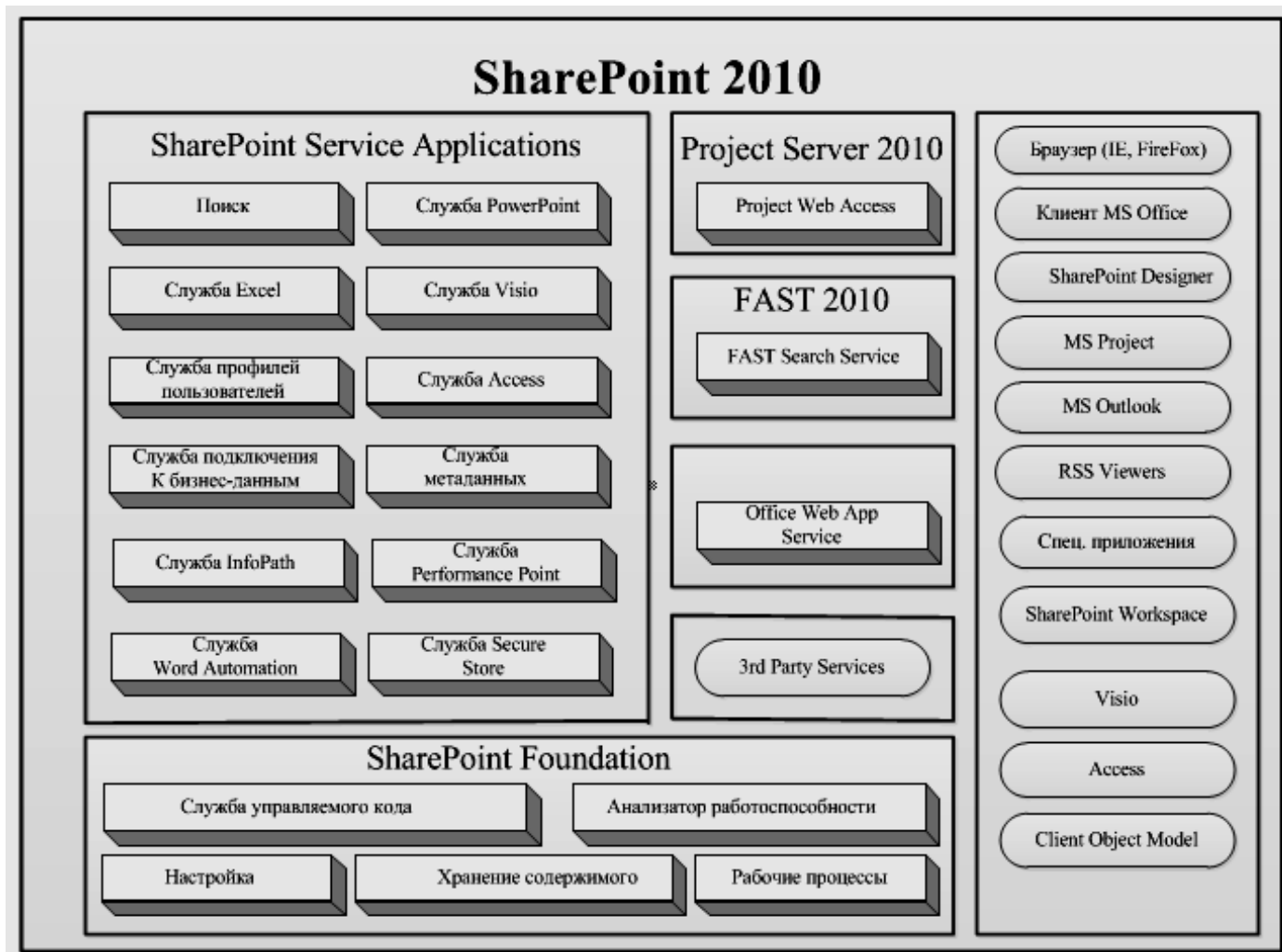


Рисунок 1 – Платформа SharePoint 2010⁶

⁶ URL: <http://iedaddy.com/2010/05/sharepoint-2007-vs-sharepoint-2010/amp/>



Единая платформа SharePoint

Рисунок 2 – Функциональный состав⁷

– документооборот (мониторинг процессов согласования; автоматические уведомления участников согласования; хранение истории согласования; структурированное хранилище – карточки документа; версионность документов; наборы документов; использование Office в качестве клиента; использование готовых шаблонов таких, как договор с клиентом, соглашение; автозаполнение однотипных данных; поиск по классификаторам, по облаку тегов – по клиенту, по городу, по сумме; настройка правил архивации и сроков хранения; регистрация входящей / исходящей корреспонденции);

– поисковая система (простой интерфейс поиска, как «в Интернете»; поиск не только по данным портала – большой объем информации, по которой нужно осуществлять поиск, а сама информация может содержаться в различных источниках: сетевых папках, электронной почте, в различных информационных системах; предварительный просмотр документов; подсказки для поиска; визуальные наиболее подходящие элементы; контекстный поиск; уточнение результатов поиска – навигаторы; индикатор присутствия в реальном времени);

– управление продажами (типовые процессы подготовки документов для клиентов; управление задачами – контроль исполнения поручениями; хранение истории общения с

⁷ Стаханов Е. Краткий обзор возможностей MS SharePoint (компания Terrelink) стр.4

клиентами; быстрый сбор отчетности; Dashboard эффективности работы каждого сотрудника с критериями для их поощрения или взыскания с них; электронный архив документов по клиентам и партнерам; карточки документов и клиентов; задачи, уведомления, напоминания; аналитика в Excel и Visio; публикация отчетов на портале);

- HR-портал (взаимодействие HR и сотрудников – заявки, опросы и др.; единая база сотрудников, информация об открытых вакансиях; отображение данных из других систем; постановка целей и обучение, актуальная информация о тренингах; формы для заполнения заявок; публикация анонсов и новостей; опросы и форумы; KPI для постановки целей – процесс оценки эффективности работы);

- маркетинговый портал;
- портал для работы с клиентами;
- HelpDesk;
- база знаний;
- внешний сайт компании.

Лицензии: Enterprise, Standard and Foundation



Рисунок 3 – Состав лицензий SharePoint 2010⁸

⁸ Стаханов Е. Краткий обзор возможностей MS SharePoint (компания Terrelink) стр. 19

SharePoint 2013

2012 – год выпуска версии SharePoint 2013. Состав системы – SharePoint Foundation Server 2013 и Microsoft SharePoint Server 2013. Присутствует возможность работы на мобильном устройстве, планшетном ПК. Усилены возможности бизнес-аналитики. Имеется большой набор инструментов разработки рабочих процессов, многоязычный интерфейс. Появился так называемый глобальный поиск (Global Search), возможность поиска не только данных, но и людей. Поддерживается работа в облаке и с Office 365, SharePoint Online. Присутствует возможность использования гибридных топологий системы на платформе SharePoint 2013.

Сервер SharePoint используется в различных системных комплексах, например, в информационной системе управления проектами (ИСУП) наряду с серверами MS Project и MS SQL имеется сервер SharePoint. Пример конфигурации ИСУП приведен на рисунках 4, 5.

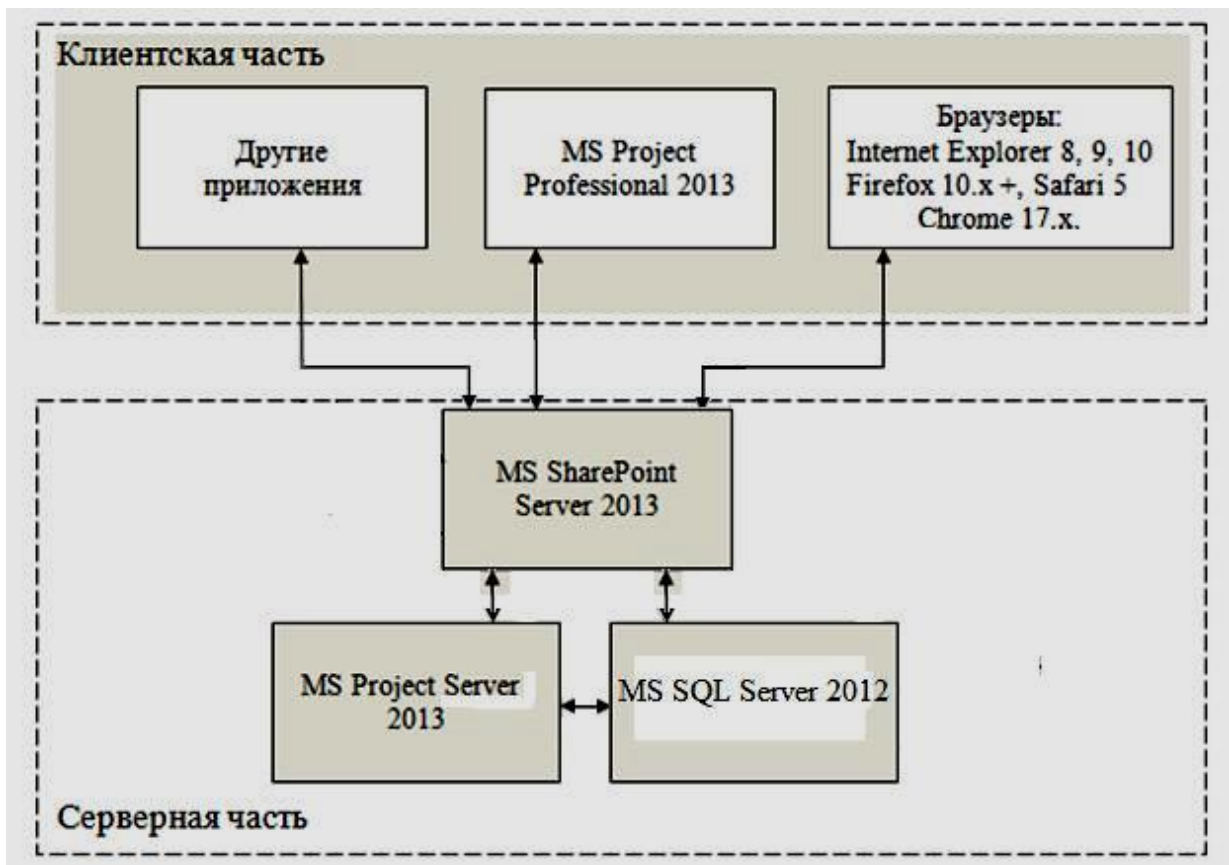


Рисунок 4 – Схема ИСУП

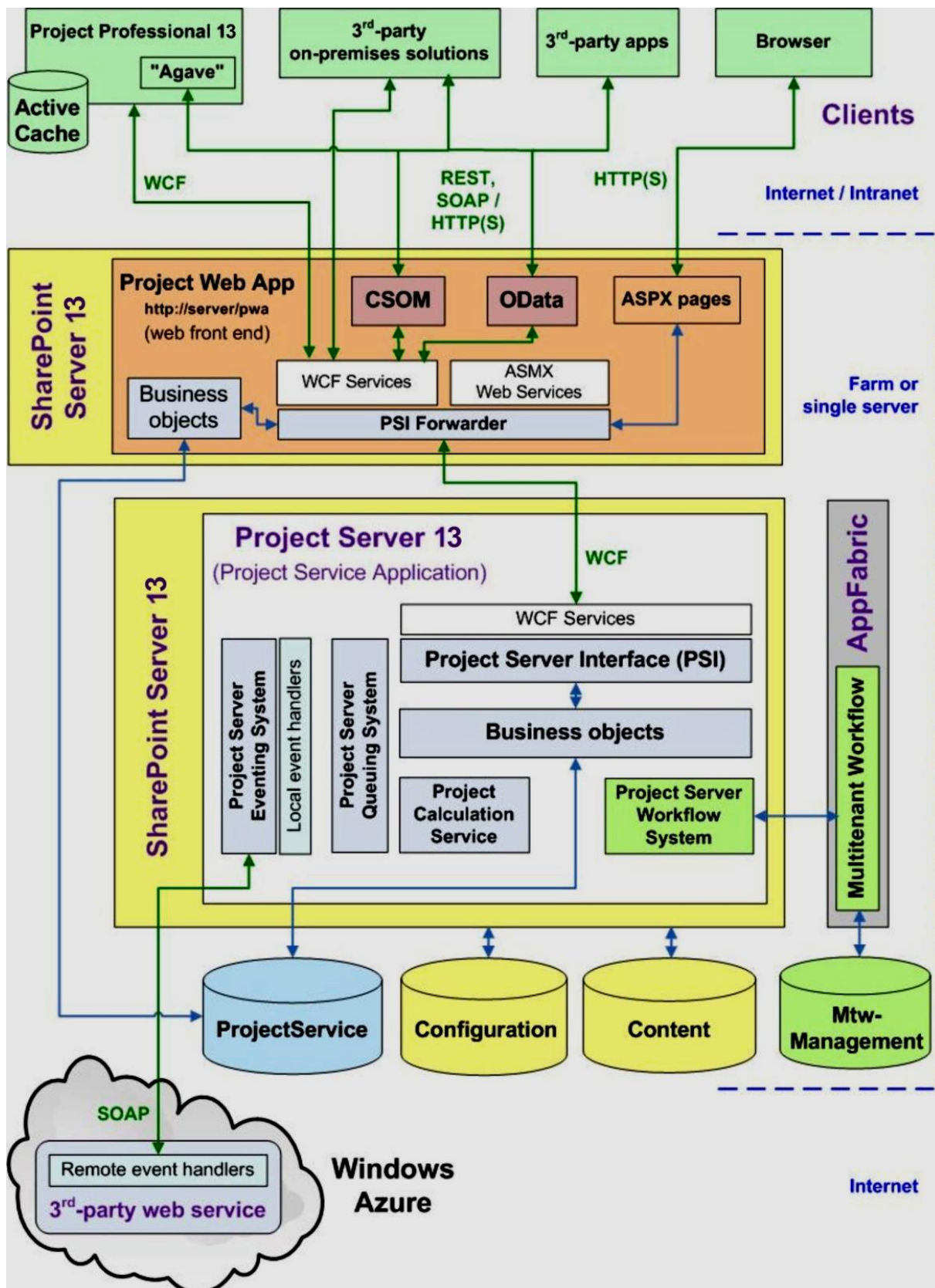


Рисунок 5 – Конфигурация ИСУП⁹

⁹ URL: <https://docs.microsoft.com/ru-ru/office/client-developer/project/project-server-2013-architecture?cid=kerryherger>

SharePoint 2016

В 2016 году вышла версия SharePoint Server 2016. Продолжилось улучшение и внедрение гибридной архитектуры системы – различные сочетания с облачными технологиями.

В SharePoint Server 2016 доступны новые функции для гибридных решений. Функции гибридных сайтов обеспечивают интегрированную среду для пользователей при использовании сайтов SharePoint Server и SharePoint Online:

- пользователи могут подписываться на сайты SharePoint Server и SharePoint Online и объединять их в один список;

- у пользователя есть единый профиль в Office 365, где хранятся все данные профиля;

- функции гибридных сайтов используются вместе с гибридной службой OneDrive для бизнеса, появившейся в SharePoint Server 2013 с пакетом обновления 1 (SP1);

- пользователи могут синхронизировать файлы с Office 365 и предоставлять другим пользователям доступ к ним;

- пользователи могут получать доступ к своим файлам непосредственно через Office 365 с любого устройства;

- облачная служба гибридного поиска предоставляет следующие возможности:

- а) отправляя запросы к индексу поиска Office 365, пользователи получают объединенные результаты поиска, включающие локальный контент и содержимое Office 365;

- б) поиск конфиденциального содержимого в SharePoint Server 2016, SharePoint Online и OneDrive для бизнеса;

- в) использование 51 встроенного типа конфиденциальных данных (кредитные карты, номера паспортов, номера социального страхования и т. д.);

- г) использование DLP Queries из семейства веб-сайтов обнаружения электронных данных, чтобы находить конфиденциальный контент, связанный с распространенными нормативными требованиями отрасли, в Центре обнаружения электронных данных SharePoint, определять документы, которые нарушают эти требования, и экспортировать отчет;

д) включение DLP Policies в семействе веб-сайтов Центра политик соответствия, чтобы сообщать пользователям и администраторам о документах с конфиденциальными данными, сохраненными в SharePoint, и автоматически защищать документы от утечек.

В числе изменений появились, например, такие как:

– функция скачивания в Excel доступна пользователям для создания сводных таблиц Access;

– расширение функций элемента управления «Связанный элемент» Access;

– в Access доступно каскадное поле со списком;

– URL-адреса на основе ресурсов сохраняют ссылки при переименовании и перемещении документов в SharePoint;

– новые командлеты Microsoft PowerShell: *New-SPCentralAdministration, Remove-SPCentralAdministration*;

– операция *psconfig.exe -cmd adminvs*;

– улучшено шифрование подключения;

– быстрое создание семейств веб-сайтов за счет использования шаблонов, которые работают на одном уровне с SQL Server, что сокращает число полных обходов между SharePoint и серверами SQL;

– возможность использования командлетов Microsoft PowerShell *SPSiteMaster* для быстрого создания сайтов и семейств веб-сайтов;

– сняты ограничения на использование:

а) символов *&*, *~*, *{* и *}* в именах файлов;

б) идентификаторов GUID, имена, которые начинались с точки;

в) имен, длина которых превышала 128 символов;

– просмотр изображений и видеороликов в библиотеках документов SharePoint;

– поддержка отправки и скачивания файлов размером более 2047 МБ;

– настройка максимального размера файла для каждого веб-приложения;

– *MinRole* – это новая функция SharePoint Server 2016, которая позволяет администратору фермы SharePoint определить роль каждого сервера в топологии фермы;

- возможность вертикального (добавление для сервера процессоров или памяти) и горизонтального (добавление дополнительных серверов в ферму) масштабирования системы;
- улучшенные возможности навигации с мобильных устройств, в том числе за счет сенсорного управления;
- новые элементы управления для работы с OneDrive для бизнеса;
- новые возможности при работе в системе управления проектами, например, несколько временных шкал в одном представлении, запрашивание нужных ресурсов, общее хранилище Project Server и SharePoint, масштабирование в облаке и др.;
- возможность бизнес-аналитики данных за счет использования SQL Server 2016 и надстроек Power Pivot и Power View;
 - а) обновление данных по расписанию;
 - б) использование книг в качестве источников данных;
 - в) использование панели управления Power Pivot;
 - г) создание отчетов Power View;
 - д) использование подписки Power View;
 - е) оповещение отчетов;
- усовершенствован общий доступ:
 - а) создание и совместное использование папок;
 - б) совместное использование подсказок;
 - в) просмотр сведений о том, кому доступна папка, во время ее просмотра;
 - г) участники могут обмениваться файлами;
 - д) улучшенное сообщение с приглашением;
 - е) отправка сообщения электронной почты одним щелчком мыши для утверждения или отклонения запроса на доступ;
 - ж) кэш элементов, недавно предоставленных для совместного доступа;
- поддержка индексирования до 500 миллионов элементов для каждого службой поиска SharePoint;
- доступно три варианта обновления семейств веб-сайтов;
- новое представление «Папки сайта», с помощью которого можно получать доступ к библиотекам документов на отслеживаемых пользователем сайтах;

– закрепление страницы сайтов – закрепленный сайт отображается в начале списка сайтов, на которые пользователь подписан;

– поддержка отправки сообщений на SMTP-серверы, использующие шифрование подключения STARTTLS;

– поддержка SMTP-серверов, использующих порты TCP, отличные от порта по умолчанию (25);

– протокол интерфейса открытой платформы веб-приложений (WOPI) для создания, переименования файлов и предоставления доступа к ним из iFrame WOPI на странице браузера;

Новые специальные возможности: для работы с библиотеками документов SharePoint Server 2016 доступны следующие функции:

– поддержка файлов формата Open Document (ODF) для использования в шаблонах библиотек документов (пользователи смогут редактировать файл в формате ODF в любой программе);

– ориентиры страницы упрощают навигацию, а для всех основных ссылок навигации улучшен замещающий текст;

– предусмотрены сочетания клавиш для следующих операций с документами:

ALT + N — «Создать»,

ALT + E — «Изменить»,

ALT + U — «Отправить»,

ALT + M — «Управление»,

ALT + S — «Общий доступ»,

ALT + Y — «Синхронизация»;

– улучшения фокуса, например, сохранение фокуса на предыдущих элементах и перехват фокуса;

– объявления о ходе отправки;

– объявления с именами и типами файлов при просмотре списков папок и файлов;

– улучшения для чтения выносок;

– обновления справочной документации.

Кроме того, предусмотрено множество изменений для установки и администрирования системы.

В SharePoint 2016 существуют следующие группы шаблонов сайтов:

– Collaboration;

– Enterprise;

– Publishing.

Имеются шаблоны для создания следующих сайтов:

- «Сайт команды» (для совместной работы, обмена документами и синхронизации);
- «Блог»;
- «Сайт проекта» (для управления и совместной работы над определенном проектом);
- «Сайт сообщества» (для обсуждения различных вопросов членами сообщества);
- «Центр документов» (для управления общими документами);
- «Центр записей» (для управления записями компаний);
- «Центр бизнес-аналитики» (с необходимыми функциями);
- «Центр поиска предприятия» (для поиска людей, видео, форумов и др.);
- «Центр базового поиска» (для общего поиска);
- «Репозиторий процессов Visio» (для хранения бизнес-процессов в формате MS Visio);
- «Публикационный сайт с рабочим процессом» (для публикации и включения рабочих процессов утверждения);
- Enterprise Wiki (для сбора и хранения информации из группового сообщества).

Компонентами SharePoint являются приложения, используемые, например, для хранения учетных документов, отслеживания контактов с клиентами и др. Для создания приложений используются следующие шаблоны:

- «Библиотека документов»;
- «Библиотека форм» (для создания бизнес-форм с использованием совместимого редактора, такого как InfoPath);
- «Библиотека страниц Wiki»;
- «Библиотека изображений»;
- «Ссылки» (для создания приложения, содержащего ссылки HTML);
- «Объявления»;
- «Контакты» (для хранения контактов);
- «Календарь»;
- «Дискуссионный совет» (для обсуждения темы в форуме);
- «Продвигаемые ссылки» (для хранения ссылок на определенные действия с использованием макета в виде плитки, при перемещении курсора мыши по плитке появляется

информация о действии, которое произойдет при нажатии клавиши мыши в этом месте);

- «Задачи» (для хранения задач);
- «Отслеживание проблем»;
- «Пользовательский список» (для создания пустого приложения на основе списка);
- «Пользовательский список в представлении Datasheet» (в виде таблицы данных);
- «Внешний список» (с подключением к внешним данным);
- «Обзор» (для проведения опросов);
- «Библиотека активов» (для хранения ресурсов сайта, таких как изображения, аудио, видео);
- «Библиотека подключения к данным»;
- «Библиотека отчетов»;
- Access App (для создания приложения, использующего веб-приложение Office Access);
- Import Spreadsheet (для импорта электронной таблицы и ввода в нее данных).

Повторно используемыми компонентами, отображающими контент на веб-страницах в SharePoint 2016, являются так называемые веб-части. Это фундаментальный компонент создания страниц SharePoint. Существует несколько веб-частей. Их состав зависит от выпуска, версии, лицензии SharePoint и от активированных функций. Кроме того, могут быть сторонние веб-части, подключаемые к среде SharePoint. Например, веб-части PerformancePoint доступны только с лицензией Enterprise и при активации функции PerformancePoint Services. Веб-части веб-приложения Project доступны только при установке Project Server.

Общие категории веб-частей:

- «Приложения» (каждый экземпляр приложения, добавленный на сайт, имеет связанную веб-часть – веб-части приложения позволяют добавлять данные в приложение);
- «Блог» (предоставляют веб-части для сайта блога);
- «Сообщество» (содержит веб-части для функций сообщества – это членство, присоединение к сообществу, информация о сообществе, инструмент для администраторов сообщества);
- «Накопитель содержимого» (веб-части используются для сворачивания агрегатного контента, результатов поиска,

предоставления резюме проекта, отображения временных графиков и соответствующих документов со всего сайта);

- «Наборы документов»;

- «Фильтры» (веб-части подключаются к другим веб-частям для обеспечения механизма фильтрации, который может быть задействован пользователем);

- «Формы» (веб-части позволяют вставлять HTML или формы InfoPath на странице);

- «Медиа и контент» (веб-части отображают изображения, видео, страницы, приложения Silverlight);

- «PerformancePoint» (предназначены для служб PerformancePoint);

- «Веб-приложение проекта» (веб-части для Project Server включают функциональные возможности для отображения информации о проекте, а именно проблемы, задачи, расписание, статус);

- «Поиск» (веб-части – окно поиска, результаты, уточнение результатов);

- «Контент, поддерживаемый поиском» (веб-части для отображения элементов, соответствующих определенному тегу, страницам на основе поискового запроса, недавно измененным элементам);

- «Социальное сотрудничество» (веб-части для контактных данных пользователя, общей доски объявлений, облака тегов, пользовательских задач).

Веб-страница отличается от текстового документа специальной разметкой. В SharePoint используются следующие типы веб-страниц:

- «Страница Wiki» (это страница контактов, можно вставлять изображения и веб-части);

- «Страница веб-части» (включает зоны веб-частей, куда можно добавлять веб-части, которые можно перетаскивать между зонами и подключать друг к другу, имеется несколько макетов зон веб-частей);

- «Страница публикации» (используется для разделения между публикацией контента и макетом содержимого на странице, позволяет создать стандартный макет страницы и позволить нескольким пользователям вводить контент).

SharePoint – это сложная система, требующая при инсталляции и настройке наличие ряда предустановленных системных программ. Например, для установки SharePoint 2016 указан следующий состав программ:

- SharePoint Server 2016;
- языковые пакеты для SharePoint Server 2016;
- Windows Server 2012 R2 или Windows Server 2016;
- Office 365 корпоративный;
- Microsoft SQL Server 2014 с пакетом обновления 1 (SP1);
- Microsoft .NET Framework 4.6;
- Microsoft WCF Data Services 5.6;
- клиент Майкрософт для защиты и контроля информации (MSIPC);
- Microsoft SQL Server 2012 Native Client с пакетом обновления 1 (устанавливается с пакетом дополнительных компонентов Microsoft SQL Server 2012);
- Microsoft ODBC Driver 11 for SQL Server;
- среда выполнения Microsoft Sync Framework 1.0 с пакетом обновления 1 (x64);
- Windows Server AppFabric 1.1;
- накопительный пакет обновления 7 для AppFabric 1.1 для Windows Server;
- распространяемый пакет Visual C++ для Visual Studio 2012 или распространяемый пакет Visual C++ для Visual Studio 2015;
- Microsoft Silverlight 3;
- управляемый API-интерфейс веб-служб Exchange, версия 1.2;
- Microsoft Identity Extensions.

Для установки необходимых компонентов системы SharePoint 2016 используется программа *prerequisiteinstaller.exe*. Порядок установки:

- роль сервера приложений, роль веб-сервера (IIS);
- Microsoft SQL Server 2012 Native Client с пакетом обновления 1 (SP1);
- Microsoft ODBC Driver 11 for SQL Server;
- среда выполнения Microsoft Sync Framework 1.0 с пакетом обновления 1 (SP1) (x64);
- Windows Server AppFabric 1.1;
- Microsoft Identity Extensions;
- клиент Microsoft для защиты и контроля информации 2.1;

- Microsoft WCF Data Services 5.6;
- Microsoft .NET Framework 4.6;
- накопительный пакет обновления 7 для Microsoft AppFabric 1.1 для Windows Server (KB 3092423);
- распространяемый пакет Visual C++ для Visual Studio 2012 (распространяемый пакет Visual C++ для Visual Studio 2015).

SharePoint 2019

2018 год – выпуск версии SharePoint 2019 на базе сервера SharePoint Server 2019. Усовершенствованы установка и настройка системы. Некоторые новые и улучшенные функции в SharePoint Server 2019:

- службы Access 2013 теперь поддерживают отправку электронной почты посредством действия *SendEmail* для отправки сообщений электронной почты из приложений;
- дополнительные ссылки на домашнюю страницу центра администрирования;
- возможность создания пользователями сайтов для общения без необходимости обращения к ИТ-специалистам, чтобы делиться новостями, демонстрировать статьи или передавать сообщения для других людей;
- быстрое, за несколько секунд, создание сайтов пользователями в следующих шаблонах сайтов:
 - а) личный сайт OneDrive [SPSPERS#10];
 - б) сайт группы (только современный) [STS#3];
 - в) сайт для общения [SITEPAGEPUBLISHING#0];
- быстрое создание сайтов используется при создании сайтов в следующих точках входа:
 - а) автоматическая подготовка личного сайта OneDrive;
 - б) кнопка «Создать сайт» на домашней странице SharePoint;
 - в) командлет PowerShell `New-SPSite` параметром-переключателем `CreateFromSiteMaster#`;
- увеличенный размер хранимых файлов в библиотеках документов до 15 Гб (в SharePoint Server 2016 до 10 Гб);
- современные возможности для списков и библиотек на сайтах групп, обеспечивающие взаимодействие, аналогичное SharePoint Online;

- возможность упрощенного общего доступа с пользовательским интерфейсом (можно легко делиться ссылками на содержимое с другими сотрудниками в организации, получать предупреждения, если предоставлен общий доступ большим группам или к большому количеству элементов);
- возможность добавления современных страниц сайтов и веб-частей на сайты групп;
- обновленный интерфейс поиска, например, возможность увидеть результаты до того, как пользователи закончат вводить текст в поле поиска, по мере ввода результаты обновляются;
- сайты групп используют новое и оперативное взаимодействие с пользователем для совместной работы участников группы:
 - а) обновленная домашняя страница упрощает обнаружение большинства распространенных задач совместной работы и помещает новости группы спереди по центру;
 - б) пользователи могут легко создавать современные сайты групп из домашней страницы SharePoint без необходимости обращения к ИТ-специалистам;
 - в) продолжение поддержки возможности создания классических сайтов групп;
- интеграция с облачными компонентами PowerApps, Power BI и Microsoft Flow (облако более удобное для пользователей);
- обновленные интерфейсы API IIS7+ для управления службами IIS (Internet Information Services);
- новый пользовательский интерфейс на домашней странице, который обеспечивает пользователям:
 - а) единый доступ ко всем сайтам: локальным и в Интернете (это позволяет пользователям легко перемещаться по внутренней сети, быстро отслеживать действия на своих сайтах, а также обеспечивает личное представление всем новостям группы);
 - б) точку запуска для самостоятельного создания новых сайтов;
 - в) переход на домашнюю страницу, щелкнув значок SharePoint в средстве запуска приложений SharePoint (заменен старый интерфейс sites.aspx);

– возможность самостоятельного создания пользователем на домашней странице новых сайтов в других веб-приложениях командой «Создать сайт», выбрав:

а) пункт «Это веб-приложение», создают сайт в том же веб-приложении;

б) пункт «Следующее веб-приложение» и выбрав веб-приложение из поля раскрывающегося списка, создают сайт в другом веб-приложении в локальной ферме (в конечном веб-приложении должна быть включена функция самостоятельного создания сайтов);

в) пункт «Следующее веб-приложение» и выбрав веб-удаленное приложение из поля раскрывающегося списка, (в локальной ферме, в которой размещается домашняя страница SharePoint, использовать функцию сопоставления с внешним ресурсом в альтернативных сопоставлениях доступа ААМ, чтобы предоставить URL-адрес веб-приложения, в котором будет сайт, затем в удаленной ферме используйте функцию сопоставления с внешним ресурсом в альтернативных сопоставлениях доступа ААМ, чтобы предоставить URL-адрес веб-приложения, в котором размещается домашняя страница, при этом должна быть включена функция самостоятельного создания сайтов в удаленной ферме), создают сайт в другом веб-приложении в удаленной ферме;

– поддержка нестандартных зон альтернативного сопоставления доступа ААМ с помощью функции самостоятельного создания сайтов на домашней странице SharePoint, внешний ресурс будет рассматриваться как внешнее веб-приложение;

– проверка подлинности для SMTP-серверов при отправке электронных сообщений, но продолжается поддержка анонимных подключений к SMTP-серверам, которые не требуют проверки подлинности – это упрощает для клиентов интеграцию SharePoint в среды с высоким уровнем безопасности, в которых требуется проверка подлинности для отправки электронных сообщений;

– синхронизация файлов с помощью клиента синхронизации OneDrive (NGSC) нового поколения вместо Groove.exe, чтобы синхронизировать файлы на личных сайтах и сайтах групп

SharePoint Server 2019 с их устройствами; клиент синхронизации OneDrive поддерживает дополнительные функции, такие как файлы по запросу, push-уведомления, защиту IRM, оставаясь при этом простым в использовании;

- использование символов # и % в именах файлов и папок;

- множество новых командлетов Microsoft PowerShell SharePoint Server 2019, например такие командлеты синхронизации профиля пользователя, как:

- а) Get-SPContentDatabase – возвращает базы данных контента, которые не были синхронизированы с профилем пользователя за последние n дней;

- б) Clear-SPContentDatabaseSyncData – удаляет сведения о синхронизации профиля пользователя из баз данных контента в ферме, которые не были синхронизированы за последние n дней;

- в) Update-SPProfileSync – обновляет параметры синхронизации профиля пользователя, чтобы задать основное расписание синхронизации, очистить расписание для определения новых пользователей, а также выявить веб-приложения, которые следует исключить из синхронизации;

- г) Get-SPContentDatabaseOrphanedData – находить потерянные объекты в базе данных контента;

- добавлено новое правило анализатора работоспособности для средства выбора людей;

- добавлено новое правило анализатора работоспособности для проверки подлинности SMTP;

- обновленная функция службы распределенного кэша AppFabric Velocity – использование фоновой сборки «мусора» по умолчанию;

- поддержка длины части URL-адреса, содержащей имена сайта и дочерних сайтов, библиотеки документов, папок и самого файла (кроме имени сервера и номера порта), до 400 символов (в предыдущей версии до 260 символов);

- строка состояния гибридного SharePoint добавлена в верхнюю часть центра администрирования, предоставляющая прямой доступ для запуска мастера гибридной конфигурации SharePoint;

– доступ в SharePoint Server 2019 к интерфейсу «OneDrive по умолчанию» для SharePoint Server 2013 для перенаправления в Office 365;

– добавление и обновление ссылок в центре администрирования для запуска мастера гибридной конфигурации SharePoint;

– улучшение восстановления из корзины – восстанавливать можно не только свои удаленные элементы, но и элементы, удаленные другими пользователями;

– обновленный дизайн общего доступа к уведомлениям электронной почты;

– улучшение навигации по набору и средства запуска приложений – интерфейс близок к внешнему виду Office 365, чтобы у гибридных клиентов SharePoint не возникало трудностей при взаимодействии со средой при перемещении между SharePoint Server 2019 и SharePoint Online;

– возможности конфиденциальности телеметрии, обеспечивающие прогнозирование будущих отчетов телеметрии, которые позволят клиентам связать SharePoint Server и телеметрию клиента синхронизации OneDrive с их гибридной областью клиента; предоставленный адрес электронной почты не отправляется за пределы фермы SharePoint, даже в корпорацию Майкрософт;

– в службах Visio внедрено несколько улучшений специальных возможностей для высококонтрастных отображений, навигации с помощью клавиатуры и экранного диктора, пользователи смогут получать доступ к разным областям с помощью следующих сочетаний клавиш:

а) *ALT+R* – переместить фокус на область примечаний;

б) *ALT+S* – переместить фокус на область данных фигуры;

в) *ALT+C* – переместить фокус на полотно;

– удаление устаревших, далее не развиваемых корпорацией Microsoft, функций (часть пока сохранена, например, поддержка служб Access 2013 сохранится, но они являются устаревшими для выпуска SharePoint Server 2019, клиентам рекомендуется ознакомиться с продуктами Microsoft PowerApps и Microsoft Flow в качестве возможной замены служб Access 2013); к устаревшим также относят:

а) службы Access 2010;

- б) службы Access 2013;
 - в) объединенный канал новостей (вместо него новости группы, сайты для общения, Yammer или группы);
 - г) настраиваемая справка (вместо нее новая система справки SharePoint обрабатывается в облаке и будет наполняться обновленным содержимым, синхронизированным с Office 365);
 - г) клиент синхронизации Groove — это клиент для синхронизации документов между личными устройствами и сайтами групп SharePoint Server 2010, 2013 и 2016 (вместо него поддержка клиента синхронизации нового поколения OneDrive);
 - д) службы InfoPath;
 - е) веб-служба списков;
 - ж) служба машинного перевода;
 - з) поддержка функции вариантов;
 - и) службы PerformancePoint Services (вместо них Microsoft PowerBI);
 - к) клиент SharePoint Designer;
 - л) почтовый ящик сайта (вместо него общие почтовые ящики);
 - м) диспетчер сайтов (вместо него функции в новом средстве перемещения файлов);
- удалены следующие функции:
- а) изолированные решения, основанные на коде;
 - б) дайджест-проверки подлинности в службах IIS (вместо них Kerberos, NTLM или SAML);
 - в) автоматический режим входящей электронной почты (вместо него использование расширенного режима, который позволяет вручную управлять службой SMTP IIS и папкой сброса);
 - г) мультитенантность;
 - д) коллекция и обновление PowerPivot (вместо него Power BI);
 - е) визуализация на основе Silverlight в службе Visio (предлагается отрисовывать схемы Visio с помощью технологии на основе формата PNG);
 - ж) сценарии бизнес-аналитики.

Для эффективного использования SharePoint важно, насколько правильно и оптимально выбрана информационная архитектура системы, то есть, как организована информация на сайте и представлена пользователям. Информационная архитектура включает в себя каркас и карту сайтов, поиск и навигацию, управляемые теги метаданных и типы контента. Хорошая информационная архитектура системы должна соответствовать следующему целеполаганию:

- управляемость: может ли ИТ-подразделение эффективно реализовать и контролировать информацию;
- соответствие нормативным требованиям;
- соответствие требованиям конфиденциальности и безопасности;
- повышение эффективности бизнеса.

Поэтому к информационной архитектуре при ее построении рекомендуется предъявлять следующие требования:

- использование метаданных для упрощения поиска и сравнения данных или контента;
- оперативное управление версиями документов и записями, чтобы определить, какая версия документа является достоверной;
- правильное каталогизирование и хранение информации, чтобы лица, принимающие решения, могли найти и использовать нужные данные;
- разработка способов навигации и представления информации так, чтобы пользователи могли найти важные сайты и сведения;
- интеграция информационной архитектуры со стратегией поиска информационной среды, чтобы пользователи могли найти нужные сведения;
- определение стратегии публикации: распределение задач создания и использования публикации на нескольких сайтах, чтобы управлять макетом сайта и отображением контента.

Или необходимо найти ответы на следующие вопросы:

- как будут предоставляться данные;
- как будет осуществляться навигация пользователей сайта;
- как будет настроен и оптимизирован поиск;
- как можно организовать контент так, чтобы поиск возвращал полезные результаты;
- какие типы контента будут динамически доступны на сайтах;

- как контент будет отмечаться тегами и как будут управляться метаданные;
- применяются ли к контенту на сайтах уникальные требования безопасности;
- каков достоверный источник терминов;
- как будут предоставляться сведения для конкретных аудиторий;
- существуют ли версии сайтов для определенных языков или продуктов;
- кто будет создавать контент для сайта и какой метод будет использоваться для его публикации.

Средствами управления информацией в SharePoint являются:

- рабочие процессы и утверждения для центров документов и страниц сайта, где бы ни хранилась официальная документация;
- журнал версий и система управления версиями для хранения журнала и основного документа;
- организатор контента для библиотек;
- политики сайта для управления жизненным циклом семейства веб-сайтов;
- права на доступ к данным (IRM) и возможности аудита для защиты и проверки важных корпоративных активов и всех сайтов, содержащих конфиденциальные данные;
- правила и политики, необходимые для следующих типов элементов:

- а) страниц;
- б) списков;
- в) документов;
- г) записей;
- д) полнофункциональных активов;
- е) блогов и вики-ресурсов;
- ж) каналов новостей;
- з) анонимных комментариев;
- и) терминов и наборов терминов;
- к) внешних данных.

На рисунке 6 представлена восьмиуровневая пирамида информационной архитектуры системы:

- ферма;
- серверы;
- зоны;

- веб-приложения;
- управляемые пути;
- базы данных;
- коллекция сайтов;
- сайты.

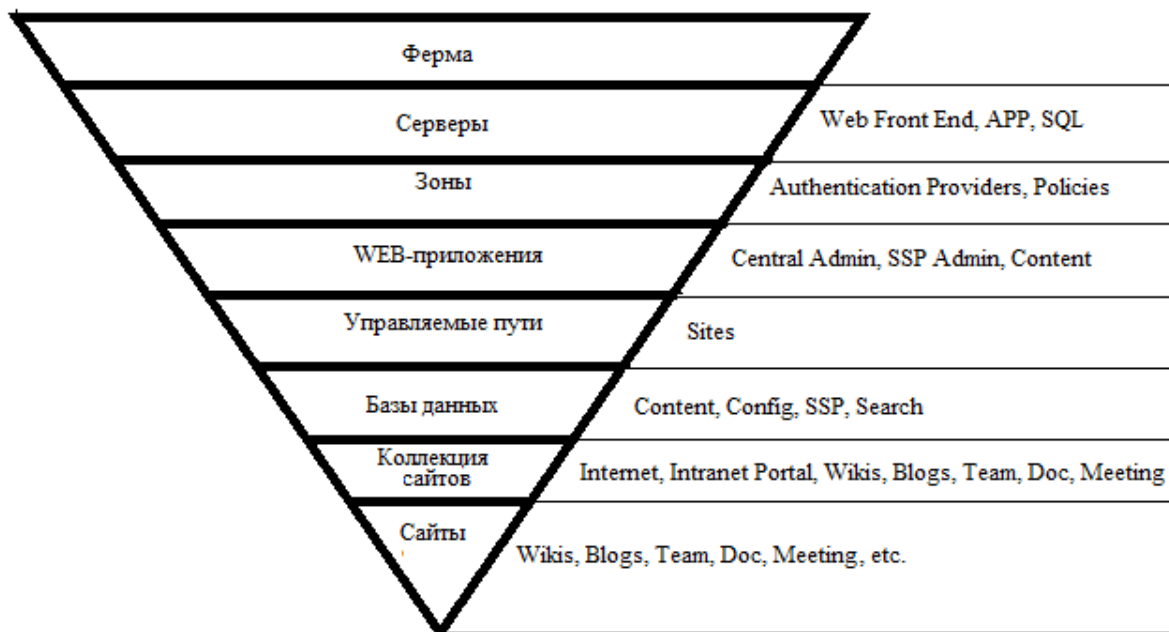


Рисунок 6 – Информационная архитектура SharePoint¹⁰

Для определения пула пользователей, заинтересованных лиц, участников решения для управления документами решаются следующие вопросы:

- кто создает документы в организации;
- какие типы документов создают пользователи;
- какая роль у пользователя документа;
- кто просматривает документы;
- кто редактирует документы;
- кто использует документы;
- кто утверждает публикацию документов;
- кто разрабатывает веб-сайты для размещения документов;
- кто устанавливает правила и политики для управления документами;
- кто управляет записями в организации;

¹⁰ URL: <https://www.simpleportals.com/SharePoint/SharePoint-Information-Architecture.aspx>

– кто развертывает и поддерживает серверы, на которых хранятся документы.

Определяются типы документов и авторы каждого типа (причем указываются роли), например, «Аналитическая заметка по капиталу», назначение «Предоставляет клиентам рекомендации по капиталу и вспомогательные сведения», автор «Финансовый аналитик», пользователи документа «Клиенты», «Члены группы», расположение документа «Клиентский компьютер», расположение копии документа «Веб-сервер». Для библиотек документов может быть использовано несколько сайтов или семейств веб-сайтов. Возможно перемещение контента между сайтами с помощью ручных и рабочих процессов или настраиваемых решений.

Ниже приведен пример цикла создания контента консультантами, работающими удаленно:

– партнер оценивает идеи относительно сотрудничества и формирует предложения;

– после заключения договора менеджер проекта собирает группу консультантов и создает с учетом особенностей сотрудничества рабочий сайт, на котором сохраняются результаты исследования и выполняется проект;

– по завершении проекта конечные документы публикуются на защищенном интернет-сайте, где пользователи получают к ним доступ;

– на основе проекта группа консультантов составляет документы с рекомендациями и конкретные примеры;

– менеджеры базы знаний собирают, упорядочивают и архивируют рекомендации и другие документы;

– конечные результаты, договоры и другие документы хранятся в качестве корпоративных записей;

– партнеры оценивают возможности и создают новые предложения, используя контент, который обслуживают менеджеры базы знаний.

Каждый тип контента может определяться следующими параметрами:

– свойства, связываемые с элементами этого типа;

– метаданные, связываемые с элементами этого типа;

– рабочие процессы, которые можно запускать из элементов этого типа;

- политики управления информацией, связываемые с элементами этого типа;
- шаблоны документов;
- пользовательские функции.

Тип контента можно связать со списком или библиотекой.

SharePoint Server использует на основе ядра СУБД SQL Server следующие типы реляционных баз данных (БД):

- «конфигурация» – это БД, содержащая данные о настройках веб-сайтов, веб-приложений, о шаблонах сайтов, о заблокированных типах файлов и др.;
- «статья» – это БД, содержащие весь контент сайтов;
- «приложение-служба» – это БД, содержащие данные, используемые приложениями-службами.

Для системы с SharePoint Server 2019 используют SQL Server 2016 или SQL Server 2017 RTM, а также Windows Server 2016 или Windows Server 2019.

На рисунке 7 приведена функциональная структура Microsoft SharePoint 2019:

- Business Intelligence (бизнес-аналитика) – Make informed decision through better insights into business data (принятие обоснованных решений с помощью лучшего понимания бизнес-данных: серверные таблицы Excel и визуализация данных, центр отчетов, веб-части BI, ключевые показатели эффективности / панели мониторинга);

- Collaboration (совместная работа, сотрудничество) – Make collaboration easy and intuitive (сделать сотрудничество простым и интуитивно понятным: документы, задачи, календари, блоги, Wiki, интеграция с электронной почтой, управление проектами управление проектами Lite, интеграция с Outlook, offline документы/списки);

- Portal (портал) – Deliver in-context and role-based access to information, processes and people (предоставление контекстного и ролевого доступа к информации, процессам и людям: шаблон корпоративного портала, каталог сайтов, личные сайты, социальные сети, контроль конфиденциальности);

- Search (поиск) – Instantly connect people with the right information and people (мгновенно предоставлять людям необходимую информацию и соединять с нужными людьми:

масштабируемость предприятия, контекстная релевантность, поиск необходимых людей и бизнес-данных);

– Content Management (управление контентом) – Manage the full enterprise content life-cycle (управление полным жизненным циклом корпоративного контента: интегрированное управление документами, записями и веб-контентом с помощью политик и рабочих процессов);

– Business Forms (формы рабочих процессов, веб-формы) – Capture critical business data and automate process (фиксация критически важных бизнес-данных и автоматизация процесса: rich и веб-формы на основе фронтальной доступности, LOB действия, подключаемый SSO);

– Windows SharePoint Platform (платформа SharePoint: рабочие области, управление, безопасность, хранение, топологии, модель сайта).



Рисунок 7 – SharePoint 2019¹¹

¹¹ URL: <https://trendogate.com/status/974114455029239812>

На рисунке 8 представлена домашняя страница SharePoint 2019.

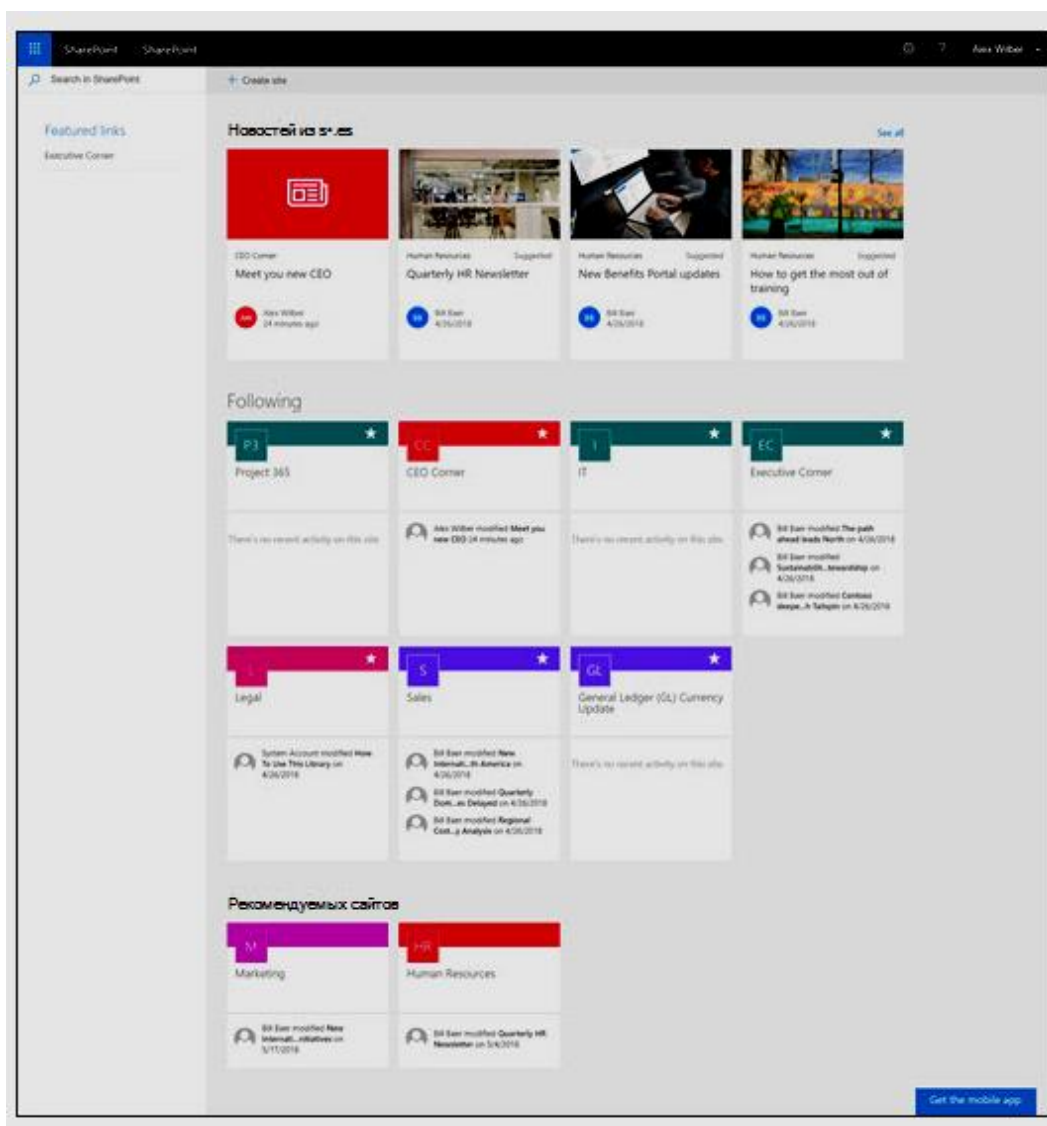


Рисунок 8 – Домашняя страница для поиска новостей, сайтов и порталов в SharePoint 2019¹²

Итак, Microsoft SharePoint Server 2019 — интегрированный пакет корпоративных приложений, призванный обеспечить увеличение производительности труда и помочь организовать совместную работу сотрудников, а также предназначенный для решения таких бизнес-задач, как контроль информационных

¹² URL: <https://support.office.com/ru-ru/article/%d1%80%d0%b0%d0%b7%d0%bb%d0%b8%d1%87%d0%b8%d1%8f-%d0%bc%d0%b5%d0%b6%d0%b4%d1%83-sharepoint-server-2016-%d0%b8-2019-ba84c8a3-3ce2-4252-926e-c67654ceb4a3?ui=ru-RU&rs=ru-RU&ad=RU>

потоков, принятие взвешенных решений и управление рабочими процессами.

Вопросы для самопроверки по содержанию раздела «Краткие сведения о системе MS SharePoint»

1. Перечислите функциональный состав платформы MS SharePoint 2010.
2. С каким пакетом программ присутствовала тесная интеграция в MS SharePoint 2010?
3. Приведите пример интеграции сервера MS SharePoint 2013 с другими серверами в системных комплексах.
4. С какими облачными компонентами может интегрироваться MS SharePoint 2019?
5. Назначение нового командлета Get-SPContentDatabaseOrphanedData в MS SharePoint 2019?
6. Основные бизнес-задачи, решаемые в MS SharePoint 2019?

MICROSOFT POWERSHELL

Общие сведения

Microsoft PowerShell – это оболочка командной строки и язык сценариев, предоставляющий администратору полный доступ к применимым интерфейсам прикладного программирования (API). Администраторы могут напрямую взаимодействовать с SharePoint Server для управления веб-приложениями, семействами сайтов, сайтами, списками и многое другое. Кроме того, администратор может создать сценарий.

PowerShell – это оболочка командной строки с поддержкой задач и язык скриптов на основе платформы .NET.

По умолчанию Powershell находится по следующему пути:

```
<%SystemRoot%>\System32\WindowsPowerShell\v1.0\  
PowerShell.exe.
```

PowerShell может использоваться авторизованными администраторами для выполнения различных задач в SharePoint локально и SharePoint Online.

Корпорация Майкрософт предоставляет командлеты, используемые для этих двух систем, а группа «Шаблоны и методы (PnP)» (the Patterns and Practices (PnP) group) создала дополнительный набор модулей, предоставляющих множество дополнительных возможностей.

Для обеспечения работы в SharePoint Online должна быть назначена роль глобального администратора на сайте SharePoint Online, на котором выполняется командлет PowerShell.

На сайте TechNet размещено множество сведений о продуктах Microsoft.

Центр сценариев TechNet (The TechNet Scripting Center) включает в себя множество ресурсов, которые помогут изучить основы PowerShell. Он также содержит репозитории сценариев с образцами сценариев, которые обычно используются с различными продуктами Microsoft.

В таблице 1 показаны основные учебные ресурсы, где раскрыты возможности PowerShell.

Таблица 1 – Учебные ресурсы

Страница TechNet Scripting Center	Описание
Windows Powershell Documentation on TechNet	В этом разделе библиотеки TechNet содержатся веб-копии разделов PowerShell Get-help. В разделе также есть веб-копии документа PowerShell Getting started, Powershell.exe – справка и пример PowerShell.
Scripting With Windows Powershell	Домашняя страница для учебных ресурсов сценариев PowerShell.
Windows Powershell Owner's Manual	Веб-руководство по началу работы с PowerShell.
Windows Powershell Quick Reference	Загружаемая копия краткого справочного документа, установленного с помощью PowerShell.

Перед использованием PowerShell для SharePoint Server полезно ознакомиться со следующими понятиями и командлетами:

- Get-Command;
- Get-Member;
- Get-Help;
- Aliasing;
- Piping and the Pipeline in Windows Powershell;
- Cmdlet Parameter Sets;
- Foreach-Object;
- Where-Object.

Командная консоль SharePoint Online (The SharePoint Online Management Shell) – это модуль Windows PowerShell, который можно использовать для управления пользователями, сайтами и семействами сайтов SharePoint Online. Операции командной строки в Windows PowerShell состоят из серии команд. Каждая команда использует командлет и набор параметров. Например, командлет New-SPOSite cmdlet используется для команды,

создающей новое семейство сайтов SharePoint Online с указанным заголовком, URL-адресом, владельцем, квотой хранения и шаблоном (это параметры).

Команды SharePoint Online PowerShell могут быть очень эффективны для пакетных операций – например, создание дюжины новых семейств сайтов или добавление сотен новых пользователей SharePoint Online на сайт.

Имеются отличия между командлетами SharePoint Online PowerShell и командлетами Office 365 PowerShell.

Главное различие – это существительные (the nouns), которые они используют. Каждый командлет SharePoint Online будет начинаться с SPO в качестве существительного. И наоборот, Office 365 будет начинаться с MSO в качестве существительного.

Набор команд командной консоли SharePoint Online отличается от набора команд PowerShell Office 365 для профессионалов и малого бизнеса. Команды командной консоли SharePoint Online предназначены для управления пользователями и семействами сайтов SharePoint Online, а команды Office 365 – для управления задачами уровня Office 365 (для всех служб Office 365), такими как пользователи, лицензии, сведения об организации и службы. Они управляют пользователями и группами. Команды SharePoint Online управляют пользователями и группами только для SharePoint Online.

Для управления пользователями или сайтами можно использовать два способа:

- с помощью SharePoint Online Admin Center;
- с помощью SharePoint Online PowerShell.

Первый способ целесообразно использовать новичкам, а второй способ более опытным администраторам, особенно если требуется управлять большим количеством пользователей, сайтов или семейств сайтов.

Чтобы узнать установленную на компьютере версию PowerShell, можно запустить консоль Windows PowerShell любым способом и выполнить следующую команду:

```
$PSversionTable
```

Результат: в строке PSVersion будет указана версия PowerShell.

Также, чтобы узнать версию PowerShell, можно использовать дополнительные команды:

```
Get-Host|Select-Object Version  
$host.version
```

Если операционная система использует PowerShell 5 или новее, можно установить SharePoint Online Management Shell, выполнив следующую команду в административном режиме:

```
Install-Module -Name Microsoft.Online.SharePoint.PowerShell  
Accessing PowerShell for SharePoint Server
```

После установки SharePoint Server применимые командлеты PowerShell доступны в командной консоли SharePoint 2016. Большинство возможностей SharePoint Server можно управлять в командной консоли SharePoint:

- создавать новые семейства веб-сайтов;
- создавать веб-приложения;
- создавать учетные записи пользователей;
- создавать приложения-службы;
- создавать прокси и многое другое.

Команды, вводимые в командной консоли SharePoint, возвращают объекты SharePoint, основанные на платформе Microsoft .NET Framework. Эти объекты можно применять в качестве входных данных для последующих команд или хранить в локальных переменных для последующего использования.

В командной консоли SharePoint нет необходимости регистрировать оснастку, содержащую командлеты. Регистрация Microsoft.SharePoint.PowerShell.dll модуля для командлетов SharePoint Server, является автоматической, в результате добавления

```
Add-PSSnapin Microsoft.SharePoint.PowerShell line в the  
SharePoint.ps1 file, т.е. в файл, который находится в  
%CommonProgramFiles%\Microsoft Shared\Web Server  
Extensions<версия > \Config\PowerShell\Registration,  
где версия "15" соответствует SharePoint Server 2013, а  
версия "16" соответствует SharePoint Server 2016. Чтобы
```

использовать консоль PowerShell, требуется регистрировать этот шаг вручную.

С помощью командной консоли SharePoint или консоли PowerShell можно также загрузить дополнительные оснастки.

Microsoft PowerShell можно использовать для выполнения одной административной задачи, но имеется возможность создания сценария для автоматизации ряда задач. Сценарий – это текстовый файл, содержащий одну или несколько команд Microsoft PowerShell. Сценарии Microsoft PowerShell имеют расширение имени файла .ps1.

Для выполнения сценариев минимально необходимой политикой выполнения для SharePoint Server является RemoteSigned, хотя политика PowerShell по умолчанию ограничена. Если политика остается ограниченной, командная консоль SharePoint изменит политику PowerShell на RemoteSigned. Это означает, что для запуска командной консоли SharePoint с повышенными правами администратора необходимо выбрать «Запуск от имени администратора». Это изменение будет применяться ко всем сеансам PowerShell.

PnP PowerShell overview

SharePoint Patterns and Practices (PnP) содержит библиотеку команд PowerShell (PnP PowerShell), которая позволяет выполнять сложные действия по подготовке и управлению артефактами для SharePoint. Команды используют CSOM и могут работать как с SharePoint Online, так и с SharePoint локально, на предприятии.

Существует три способа установки командлетов. По возможности рекомендуется устанавливать их из галереи PowerShell. Кроме того, можно загрузить файлы установки или запустить сценарий PowerShell для загрузки модуля PowerShellGet и последующей установки командлетов.

Предлагаются следующие актуальные в настоящее время варианты (рисунок 9).

PowerShell Gallery

If you main is Windows 10, or if you have PowerShellGet installed, can run the following commands to install the Powershell cmdlets:

SHAREPOINT VERSION	COMMAND TO INSTALL
SharePoint Online	Install - Module SharePointPnPPowerShellOnline
SharePoint2019	Install - Module SharePointPnPPowerShell2019
SharePoint2016	Install - Module SharePointPnPPowerShell2016
SharePoint2013	Install - Module SharePointPnPPowerShell2013

Рисунок 9 – Варианты загрузки модуля PowerShellGet

Каждый месяц будет доступен новый выпуск командлетов PnP PowerShell.

Если ранее были установлены командлеты с помощью файла установки, то можно просто загрузить последнюю версию и запустите установку. Это обновит существующую установку.

Если были установлены командлеты с помощью PowerShellGet with Install-Module из галереи PowerShell, то можно использовать следующую команду для установки последней обновленной версии:

`Update-Module SharePointPnPPowerShell*`

Это автоматически загрузит модуль после запуска PowerShell 3.0.

Установленные версии PnP-PowerShell можно проверить с помощью следующей команды:

`Get-Module SharePointPnPPowerShell* -ListAvailable | Select-Object Name,Version | Sort-Object Version - Descending`

Актуальные командлеты

В таблице 2 представлен актуальный перечень командлетов. В столбце «Платформа» таблицы значение «Все» означает, что данный командлет может быть использован в четырех

платформах: SharePoint Server 2013, SharePoint Server 2016, SharePoint Server 2019, SharePoint Online.

Таблица 2 – Список командлетов

Командлет (Cmdlet)	Описание	Платформа
Apps		
Add-PnPApp	Добавление / загрузка доступного приложения в каталог приложений	SharePoint Server 2019, SharePoint Online
Get-PnPApp	Возвращает доступные приложения из каталога приложений	SharePoint Server 2019, SharePoint Online
Install-PnPApp	Устанавливает доступное приложение из каталога приложений	SharePoint Server 2019, SharePoint Online
Publish-PnPApp	Публикует/развертывает / доверяет доступному приложению в каталоге приложений	SharePoint Server 2019, SharePoint Online
Remove-PnPApp	Удаляет приложение из каталога приложений	SharePoint Server 2019, SharePoint Online
Uninstall-PnPApp	Удаляет доступную надстройку из сайта	Все
Unpublish-PnPApp	Не публикует / убирает доступную надстройку из каталога приложений	SharePoint Server 2019, SharePoint Online
Update-PnPApp	Обновляет доступное приложение из каталога приложений	SharePoint Server 2019, SharePoint Online

Продолжение таблицы 2

Get-PnPAppInstance	Возвращает экземпляр надстройки SharePoint	Все
Uninstall-PnPAppInstance	Удаляет экземпляр надстройки с сайта	Все
Import-PnPAppPackage	Импортирует экземпляр надстройки SharePoint на сайт	Все
Grant-PnPTenantServicePrincipalPermission	Явно предоставляет указанное разрешение в главную службу «клиент SharePoint Online»	SharePoint Online
Base Cmdlets		
Get-PnPAppAuthAccessToken	Возвращает маркер доступа	Все
Get-PnPAppAuthenticationRealm	Возвращает область проверки подлинности	Все
Get-PnPAzureADManifestKeyCredentials	Возвращает фрагмент Манифеста JSON для приложения Azure	Все

Продолжение таблицы 2

Get-PnPAzureCertificate	Возвращает PEM значения и параметры манифеста существующего сертификата (.pfx) при использовании CSOM via только для приложения ADAL. Ключевые учетные данные содержат разделы манифеста приложения ADAL. Сертификат содержит в формате PEM сертификат закрытого ключа.	Все
New-PnPAzureCertificate	<p>Генерирует новый 2048-битный самозаверяющий сертификат и параметры манифеста пользователя при использовании CSOM via приложения ADAL.</p> <p>Ключевые учетные данные содержат разделы манифеста приложения ADAL. Сертификат содержит в формате PEM сертификат закрытого ключа.</p>	Все
Get-PnPConnection	Возвращает текущее соединение	Все
Get-PnPContext	Возвращает текущий контекст	Все

Продолжение таблицы 2

Set-PnPContext	Устанавливает контекст клиента	Все
Get-PnPException	Возвращает последнее возникшее исключение	Все
Get-PnPHealthScore	Получает оценку состояния	Все
Connect-PnPOnline	Подключает к сайту SharePoint	Все
Disconnect-PnPOnline	Отключает контекст	Все
Get-PnPProperty	Возвращает ранее не загруженное свойство объекта клиента	Все
Invoke-PnPQuery	Выполняет текущие действия в очереди	Все
Add-PnPStoredCredential	Добавляет учетные данные в Windows Диспетчер Учетных Данных	Все
Get-PnPStoredCredential	Получает учетные данные	Все
Remove-PnPStoredCredential	Удаляет учетные данные	Все
Set-PnPTraceLog	Включает / выключает log журнал	Все
Branding		
Add-PnPCustomAction	Добавляет пользовательское действие	Все
Get-PnPCustomAction	Возвращает пользовательское действие	Все
Remove-PnPCustomAction	Удаляет пользовательское действие	Все

Продолжение таблицы 2

Get-PnPHomePage	Возвращает домашнюю страницу	Все
Set-PnPHomePage	Устанавливает домашнюю страницу текущего веб-сайта	Все
Add-PnPJavaScriptBlock	Добавляет ссылку на фрагмент/блок JavaScript к веб-сайту или семейству сайтов	Все
Add-PnPJavaScriptLink	Добавляет ссылку на файл JavaScript к сайту или семейству сайтов	Все
Get-PnPJavaScriptLink	Возвращает все или определенные пользовательские действия с типом расположения ScriptLink	Все
Remove-PnPJavaScriptLink	Удаляет ссылку или блок JavaScript из веб-сайта или семейства сайтов	Все
Get-PnPMasterPage	Возвращает URL-адреса Мастера Страниц по умолчанию и настраиваемой Главной страницы	Все
Set-PnPMasterPage	Устанавливает главную страницу	Все

Продолжение таблицы 2

Set-PnPMinimalDownloadStrategy	Активирует или деактивирует минимальную стратегию загрузки	Все
Add-PnPNavigationNode	Добавляет пункт в элемент навигации	Все
Get-PnPNavigationNode	Возвращает все или определенный узел навигации	Все
Remove-PnPNavigationNode	Удаляет элемент из любого меню навигации, быстрого запуска или верхнего	Все
Disable-PnPResponsiveUI	Отключает пользовательский интерфейс PnP добавления ответа	Все
Enable-PnPResponsiveUI	Подключает пользовательский интерфейс PnP добавления ответа	Все
Get-PnPTheme	Возвращает текущую тему / композицию просмотра текущего web	Все
Set-PnPTheme	Задает тему текущего сайта	Все
Set-PnPWebTheme	Задает тему текущего веб-сайта	SharePoint Online
Client-Side Pages		
Get-PnPAvailableClientSideComponents	Возвращает доступную клиентскую сторону компоненты на определенной странице	SharePoint Online, SharePoint 2019

Продолжение таблицы 2

Move-PnPClientSideComponent	Перемещает компоненту на стороне клиента в различные раздел / столбец	SharePoint Online, SharePoint 2019
Add-PnPClientSidePage	Добавляет страницу на стороне клиента	SharePoint Online, SharePoint 2019
ConvertTo-PnPClientSidePage	Преобразует классическую страницу (wiki или часть web страницы) в страницу на стороне клиента	SharePoint Online
Get-PnPClientSidePage	Возвращает страницу на стороне клиента	SharePoint Online, SharePoint 2019
Remove-PnPClientSidePage	Удаляет страницу на стороне клиента	SharePoint Online, SharePoint 2019
Set-PnPClientSidePage	Задаёт параметры клиентской страницы	SharePoint Online, SharePoint 2019
Add-PnPClientSidePageSection	Добавляет новый раздел на стороне клиента страницы	SharePoint Online, SharePoint 2019
Add-PnPClientSideText	Добавляет текстовый элемент на клиентскую страницу	SharePoint Online, SharePoint 2019

Продолжение таблицы 2

Add-PnPClientSidewebPart	Добавляет клиентскую веб-часть на клиентскую страницу	SharePoint Online
Content Types		
Add-PnPContentType	Добавляет новый тип контента	Все
Get-PnPContentType	Возвращает тип контента	Все
Remove-PnPContentType	Удаляет тип контента с web	Все
Remove-PnPContentTypeFromList	Удаляет тип контента из списка	Все
Get-PnPContentTypePublishingHubUrl	Возвращает URL-адрес типа контента Издательского Центра	Все
Add-PnPContentTypeToList	Добавляет новый тип контента в список	Все
Set-PnPDefaultContentTypeToList	Устанавливает тип контента по умолчанию для списка	Все
Remove-PnPFieldFromContentType	Удаляет поле из типа контента сайта	Все
Add-PnPFieldToContentType	Добавляет существующее поле в тип контента сайта	Все
Diagnostic utilities		
Measure-PnPList	Возвращает статистику по объектам списка	Все

Продолжение таблицы 2

Measure-PnPResponseTime	Получает статистику по времени отклика для заданной конечной точки, отправив пробный запрос	Все
Measure-PnPWeb	Возвращает статистику по веб-объекту	Все
Document Sets		
Remove-PnPContentTypeFromDocumentSet	Удаляет тип контента из пакета документов	Все
Add-PnPContentTypeToDocumentSet	Добавляет тип контента в пакет документов	Все
Add-PnPDocumentSet	Создает новый пакет документов в библиотеке	Все
Set-PnPDocumentSetField	Устанавливает столбец сайта из списка доступных типов контента для набора документов	Все
Get-PnPDocumentSetTemplate	Извлекает шаблон из пакета документов	Все
Event Receivers		
Add-PnPEventReceiver	Добавляет новый удаленный приемник событий	Все
Get-PnPEventReceiver	Возвращает зарегистрированные приемники событий	Все
Remove-PnPEventReceiver	Удаляет приемник событий	Все

Продолжение таблицы 2

Features		
New-PnPExtensibilityHandlerObject	Создает объект обработчика расширяемости для использования в Командлет Get-SPOProvisioningTemplate	Все
Disable-PnPFeature	Отключает функцию	Все
Enable-PnPFeature	Включает функцию	Все
Get-PnPFeature	Возвращает все активированные или определенную активированную функцию	Все
Fields		
Add-PnPField	Добавляет поле	Все
Get-PnPField	Возвращает поле из списка или сайта	Все
Remove-PnPField	Удаляет поле из списка или сайта	Все
Set-PnPField	Изменяет одно или несколько свойств поля в определенном списке или для всего web	Все
Add-PnPFieldFromXml	Добавляет поле в список или в виде столбца сайта на основе определения поля CAMEL / XML	Все
Add-PnPTaxonomyField	Добавить поле таксономии	Все
Set-PnPView	Изменить свойства вида	Все

Продолжение таблицы 2

Files and Folders		
Add-PnPFile	Загружает файл в web	Все
Copy-PnPFile	Копирует файл или папку в другое местоположение	Все
Find-PnPFile	Находит файл в виртуальной файловой системе web	Все
Get-PnPFile	Загружает файл	Все
Move-PnPFile	Перемещает файл в другое место	Все
Remove-PnPFile	Удаляет файл	Все
Rename-PnPFile	Переименовывает файл в его текущем местоположении	Все
Set-PnPFileCheckedIn	Проверяет в файле	Все
Set-PnPFileCheckedOut	Проверяет файл	Все
Reset-PnPFileVersion	Сброс файла к предыдущей версии	Все
Add-PnPFolder	Создает папку в родительской папке	Все
Get-PnPFolder	Возвращает объект Папка	Все
Move-PnPFolder	Перемещает папку в другое место текущего web	Все
Remove-PnPFolder	Удаляет папку в родительской папке	Все
Rename-PnPFolder	Переименовывает папку	Все

Продолжение таблицы 2

Resolve-PnPFolder	Возвращает папку с данного сайта по относительному пути и создаст ее, если она не существует	Все
Get-PnPFolderItem	Список содержимого в папке	Все
Copy-PnPItemProxy	Командлет Proxy для использования Copy-Item между провайдером SharePoint и провайдером файловых систем	Все
Move-PnPItemProxy	Командлет Proxy для использования Move-Item между провайдером SharePoint и провайдером файловых систем	Все
Information Management		
Get-PnPLabel	Возвращает метку / тег указанного списка или библиотеки (если доступно)	SharePoint Online
Set-PnPLabel	Устанавливает метку/тег в указанном списке или библиотеке	SharePoint Online
Get-PnPListInformationRightsManagement	Получить список ролей (управление)	Все

Продолжение таблицы 2

Set-PnPListInformationRightsManagement	Установить список ролей (управление)	Все
Get-PnPSiteClosure	Получить статус закрытия сайта, к которому применяется политика сайта	Все
Set-PnPSiteClosure	Открывает или закрывает сайт, к которому применяется политика сайта	Все
Set-PnPSitePolicy	Устанавливает политику сайта	Все
Get-PnPSitePolicy	Извлекает все или определенную политику сайта	Все
Lists		
Clear-PnPDefaultColumnValues	Очистить значения столбцов по умолчанию для библиотеки документов	Все
Get-PnPDefaultColumnValues	Возвращает значения столбцов по умолчанию для всех папок в библиотеке документов	Все
Set-PnPDefaultColumnValues	Устанавливает значение по умолчанию столбца библиотеки документов	Все
Get-PnPList	Возвращает список объектов	Все

Продолжение таблицы 2

New-PnPList	Создает новый список	Все
Remove-PnPList	Удаляет список	Все
Set-PnPList	Обновляет список настроек	Все
Add-PnPListItem	Добавляет элемент в список	Все
Get-PnPListItem	Извлекает элементы списка	Все
Remove-PnPListItem	Удаляет элемент из списка	Все
Set-PnPListItem	Обновляет элемент из списка	Все
Set-PnPListItemPermission	Устанавливает разрешения для элементов списка	Все
Move-PnPListItemToRecycleBin	Переместить элемент из списка в корзину Bin	Все
Set-PnPListPermission	Устанавливает список разрешений	Все
Get-PnPProvisioningTemplateFromGallery	Извлекает или ищет инициализацию шаблонов из Галереи шаблонов PnP	Все
Request-PnPReIndexList	Помечает список для полной индексации во время следующего инкрементного обхода	Все
Add-PnPView	Добавляет представление в список	Все

Продолжение таблицы 2

Get-PnPView	Возвращает одно или все представления из списка	Все
Remove-PnPView	Удаляет представление из списка	Все
Microsoft Graph		
Connect-PnPMicrosoftGraph	Подключение к Microsoft Graph	Все
Add-PnPSiteClassification	Добавляет одно или несколько значений классификаций сайтов в список возможных значений. Требуется подключение к системе Microsoft Graph.	SharePoint Online
Disable-PnPSiteClassification	Отключает классификации сайтов для арендатора. Требуется подключение к системе Microsoft Graph	SharePoint Online
Enable-PnPSiteClassification	Включает классификации сайтов для арендатора. Требуется подключение к системе Microsoft Graph	SharePoint Online

Продолжение таблицы 2

Get-PnPSiteClassification	Возвращает определенные классификации сайтов для арендаторов. Требуется подключение к системе Microsoft Graph	SharePoint Online
Remove-PnPSiteClassification	Удаляет один или несколько значений классификации существующих сайтов из списка возможных значений. Требуется подключение к системе Microsoft Graph	SharePoint Online
Update-PnPSiteClassification	Обновляет классификации сайтов для арендатора. Требуется подключение к системе Microsoft Graph	SharePoint Online
Get-PnPUnifiedGroup	Получает одну группу Office 365 (она же объединенная группа) или список групп Office 365	SharePoint Online
New-PnPUnifiedGroup	Создает новую группу Office 365 (она же объединенная группа)	SharePoint Online

Продолжение таблицы 2

Remove-PnPUnifiedGroup	Удаляет группу Office 365 (она же объединенная группа)	SharePoint Online
Set-PnPUnifiedGroup	Устанавливает группу Office 365 (она же объединенная группа)	SharePoint Online
Get-PnPUnifiedGroupMembers	Возвращает членов определенной группы Office 365 (она же объединенная группа)	SharePoint Online
Get-PnPUnifiedGroupOwners	Получает владельцев определенной группы Office 365 (она же объединенная группа)	SharePoint Online
Provisioning		
Export-PnPClientSidePage	Экспортирует клиентскую страницу в PnP шаблон инициализации	SharePoint Online
Add-PnPDataRowsToProvisioningTemplate	Добавляет потоки данных во внутренний список PnP шаблона инициализации	Все
Remove-PnPFileFromProvisioningTemplate	Удаляет файл из PnP шаблона инициализации	Все
Add-PnPFileToProvisioningTemplate	Добавляет файл в PnP шаблон инициализации	Все
Convert-PnPFolderToProvisioningTemplate	Создает упакованный файл pnp существующего шаблона xml, и включает все файлы в текущую папку	Все

Продолжение таблицы 2

Add-PnPListFoldersToProvisioningTemplate	Добавляет папки в список в PnP шаблона инициализации	Все
Add-PnPProvisioningTemplate	Добавляет объект шаблона инициализации PnP к шаблону клиента	SharePoint Online
Apply-PnPProvisioningTemplate	Применяет шаблон сайта к веб-сайту	Все
Convert-PnPProvisioningTemplate	Преобразует шаблон инициализации в другую версию схемы	Все
Get-PnPProvisioningTemplate	Создает шаблон инициализации сайта из web	Все
New-PnPProvisioningTemplate	Создает объект нового шаблона инициализации	Все
Read-PnPProvisioningTemplate	Загружает / читает Php-файл из файловой системы	Все
Save-PnPProvisioningTemplate	Сохраняет шаблон сайта PnP в файловую систему	Все
New-PnPProvisioningTemplateFromFolder	Создает шаблон инициализации из полученной папки, включая только файлы, которые присутствуют в этой папке	Все
Set-PnPProvisioningTemplateMetadata	Устанавливает метаданные шаблона инициализации	Все

Продолжение таблицы 2

Add-PnPTenantSequence	Добавляет объект последовательности арендаторов в шаблон арендатора	SharePoint Online
Get-PnPTenantSequence	Возвращает одну или несколько инициализаций последовательности объектов из шаблона арендатора	Все
New-PnPTenantSequence	Создает новый объект последовательности арендаторов	SharePoint Online
New-PnPTenantSequenceCommunicationSite	Создает объект узла связи	SharePoint Online
Add-PnPTenantSequenceSite	Добавить объект сайта существующей последовательности арендаторов к шаблону арендатора	SharePoint Online
Get-PnPTenantSequenceSite	Возвращает один или несколько сайтов из шаблона арендатора	Все
Add-PnPTenantSequenceSubSite	Добавляет объект вложенного сайта последовательности арендаторов в объект сайта последовательности арендаторов	SharePoint Online
New-PnPTenantSequenceTeamNoGroupSite	Создает новый сайт группы арендаторов без объекта группы Office 365 в памяти	SharePoint Online

Продолжение таблицы 2

New-PnPTenantSequenceTeamNoGroupSubSite	Создает дочерний сайт сайта группы арендаторов без объекта группы Office 365	SharePoint Online
New-PnPTenantSequenceTeamSite	Создает объект сайта группы	SharePoint Online
Apply-PnPTenantTemplate	Применяет шаблон арендатора к текущему арендатору	SharePoint Online
New-PnPTenantTemplate	Создает новый объект шаблона арендатора	SharePoint Online
Read-PnPTenantTemplate	Загружает / считывает шаблон арендатора PnP из файловой системы и возвращает в память экземпляр этого шаблона	SharePoint Online
Save-PnPTenantTemplate	Сохраняет иерархию инициализации PnP в файловой системе	SharePoint Online
Test-PnPTenantTemplate	Проверяет шаблон арендатора на недопустимую ссылку	Все
Publishing		
Set-PnPAvailablePageLayouts	Устанавливает доступные макеты страниц для текущего сайта	Все
Set-PnPDefaultPageLayout	Устанавливает определенный макет страницы в качестве макета страницы по умолчанию для сайта публикации	Все

Продолжение таблицы 2

Add-PnPHtmlPublishingPageLayout	Добавляет макет страницы публикации на основе HTML	Все
Add-PnPMasterPage	Добавляет Мастера страниц	Все
Add-PnPPublishingImageRendition	Добавляет представление изображения, если имя представления изображения уже не существует. Это предотвращает создание двух изображений представления с одинаковым именем	Все
Get-PnPPublishingImageRendition	Возвращает все представления изображения или если указывается конкретная идентичность одного	Все
Remove-PnPPublishingImageRendition	Удаляет существующее представление изображения	Все
Add-PnPPublishingPage	Добавляет страницу публикации	Все
Add-PnPPublishingPageLayout	Добавляет макет страницы публикации	Все
Add-PnPWikiPage	Добавляет страницу Wiki	Все
Remove-PnPWikiPage	Удаляет страницу Wiki	Все
Get-PnPWikiPageContent	Возвращает содержимое / источник страницы wiki	Все

Продолжение таблицы 2

Set-PnPWikiPageContent	Устанавливает контент страницы wiki	Все
Records Management		
Get-PnPInPlaceRecordsManagement	Возвращает место записи, если функция управления включена	Все
Set-PnPInPlaceRecordsManagement	Активирует или деактивирует в месте записи функцию управления	Все
Disable-PnPInPlaceRecordsManagementForSite	Отключает управление местом записей на сайте	Все
Enable-PnPInPlaceRecordsManagementForSite	Включает управление местом записей на сайте	Все
Clear-PnPListItemAsRecord	Очищает список от элемента как записи	SharePoint Online
Set-PnPListItemAsRecord	Устанавливает элемент списка как запись	SharePoint Online
Test-PnPListItemIsRecord	Проверяет, является ли элемент списка записью	SharePoint Online
Get-PnPListRecordDeclaration	Возвращает объявление вручную настроек записи для списка	SharePoint Online

Продолжение таблицы 2

Set-PnPListRecordDeclaration	Параметр объявления записи в списке поддерживает 4 значения: AlwaysAllowManualDeclaration; NeverAllowManualDeclaration; UseSiteCollectionDefaults	SharePoint Online
Search		
Get-PnPSearchConfiguration	Возвращает конфигурацию поиска	Все
Remove-PnPSearchConfiguration	Удаляет конфигурацию поиска	Все
Set-PnPSearchConfiguration	Устанавливает конфигурацию поиска	Все
Get-PnPSearchCrawlLog	Возвращает записи из log журнала поиска SharePoint	SharePoint Online
Submit-PnPSearchQuery	Выполняет произвольный поисковый запрос без индекса поиска SharePoint	Все
Get-PnPSiteSearchQueryResults	Выполняет поисковый запрос для извлечения индексированного семейства сайтов	Все
SharePoint Recycle Bin		
Clear-PnPRecycleBinItem	Безвозвратно удаляет все или определенный элемент из корзины Bin	Все

Продолжение таблицы 2

Get-PnPRecycleBinItem	Возвращает элемент в корзину Bin из контекста	Все
Move-PnPRecycleBinItem	Перемещает все элементы или определенный элемент текущего сайта в поле корзины Bin на первом этапе и сбор и утилизация на втором этапе	SharePoint Online
Restore-PnPRecycleBinItem	Восстанавливает выбранный элемент корзины на свое первоначальное местоположение	Все
Get-PnPTenantRecycleBinItem	Возвращает все коллекции современных и классических сайтов в области арендатора корзины	SharePoint Online
SharePoint WebHooks		
Add-PnPWebhookSubscription	Добавляет новую подписку Webhook	SharePoint Online
Remove-PnPWebhookSubscription	Удаляет подписку Webhook из ресурса	SharePoint Online
Set-PnPWebhookSubscription	Обновляет подписку Webhook	SharePoint Online
Get-PnPWebhookSubscriptions	Возвращает все подписки Webhook в ресурс	SharePoint Online
Sites		
Set-PnPAppSideLoading	Включает функцию AppSideLoading на сайте	Все

Продолжение таблицы 2

Get-PnP Auditing	Возвращает параметры аудита сайта	Все
Set-PnP Auditing	Устанавливает параметры аудита сайта	Все
Test-PnP Office365GroupAliasIsUsed	Проверяет используется ли данный псевдоним	SharePoint Online
Add-PnP RoleDefinition	Добавляет определение роли (уровень разрешений) к семейству сайтов в текущем контексте	Все
Get-PnP RoleDefinition	Извлекает определения ролей сайта	Все
Remove-PnP RoleDefinition	Удаляет определения ролей сайта	Все
Get-PnP Site	Возвращает текущее семейство сайтов из контекста	Все
Set-PnP Site	Устанавливает свойства семейства сайтов	SharePoint Online
Add-PnP SiteCollectionAdmin	Добавляет одного или нескольких пользователей в качестве администраторов семейства сайтов в текущем контексте	Все
Get-PnP SiteCollectionAdmin	Возвращает текущих администраторов семейства сайтов в текущем контексте	Все

Продолжение таблицы 2

Remove-PnPSiteCollectionAdmin	Удаляет одного или нескольких пользователей в качестве администраторов семейства сайтов в текущем контексте	Все
Install-PnPSolution	Устанавливает изолированное решение на семейство сайтов. Внимание! Этот метод может удалить свою составленную галерею взглядов из-за метода, используемого для активации решения. Рекомендуется использовать этот командлет, если только вы не против этого	Все
Uninstall-PnPSolution	Удаляет изолированное решение из семейства сайтов	Все
Taxonomy		
Get-PnPSiteCollectionTermStore	Возвращает хранилище терминов семейства сайтов	Все
Export-PnPTaxonomy	Экспортирует таксономию на любой вывод или в файл	Все

Продолжение таблицы 2

Import-PnPTaxonomy	Импортирует таксономию из любого массива строк или файла	Все
Set-PnPTaxonomyFieldValue	Устанавливает значение термина таксономии в поле listitem	Все
Get-PnPTaxonomyItem	Возвращает элемент таксономии	Все
Remove-PnPTaxonomyItem	Удаляет элемент таксономии	Все
Get-PnPTaxonomySession	Возвращает сеанс таксономии	Все
Get-PnPTerm	Возвращает термин таксономии	Все
New-PnPTerm	Создает термин таксономии	Все
Get-PnPTermGroup	Возвращает группу терминов таксономии	Все
New-PnPTermGroup	Создает группу терминов таксономии	Все
Remove-PnPTermGroup	Удаляет группу терминов таксономии и все наборы терминов, содержащиеся в ней	Все
Import-PnPTermGroupFromXml	Импортирует группу терминов таксономии из любого входа либо из XML-файла	Все
Export-PnPTermGroupToXml	Экспортирует группу терминов таксономии из любого выхода либо в XML-файла	Все
Get-PnPTermSet	Возвращает набор терминов таксономии	Все

Продолжение таблицы 2

Import-PnPTermSet	Импортирует набор терминов таксономии из файла в стандартном формате	Все
New-PnPTermSet	Создает набор терминов таксономии	Все
Tenant Administration		
Get-PnPAccessToken	Возвращает текущий OAuth Маркер доступа	Все
Get-PnPHideDefaultThemes	Возвращает значение по умолчанию OOTB темы, т.е. видимы для пользователей или нет	SharePoint Online
Set-PnPHideDefaultThemes	Устанавливает значение по умолчанию OOTB темы, т.е. видимы для пользователей или нет	SharePoint Online
Get-PnPHubSite	Получить все или определенный hubsite	SharePoint Online
Register-PnPHubSite	Регистрирует сайт в качестве концентратора	SharePoint Online
Set-PnPHubSite	Устанавливает свойства концентратора	SharePoint Online
Unregister-PnPHubSite	Отменяет регистрацию сайта как hubsite	SharePoint Online

Продолжение таблицы 2

Add-PnPHubSiteAssociation	Подключает сайт к концентратору	SharePoint Online
Remove-PnPHubSiteAssociation	Отключает сайт от концентратора	SharePoint Online
Grant-PnPHubSiteRights	Предоставление дополнительных разрешений к разрешениям в месте связывания сайтов с концентратором сайтов для одного или нескольких определенных пользователей	SharePoint Online
Add-PnPOffice365GroupToSite	Добавляет классический сайт группы путем создания группы на Office 365 и подключения сайта с новыми созданными группами	SharePoint Online
Disable-PnPPowerShellTelemetry	Отключает PnP PowerShell отслеживания телеметрии	Все
Enable-PnPPowerShellTelemetry	Разрешает PnP PowerShell отслеживание телеметрии	Все
Get-PnPPowerShellTelemetryEnabled	Возвращает «истину», если в PnP PowerShell включено отслеживание телеметрии	Все

Продолжение таблицы 2

New-PnPSite	Создает новое семейство сайтов	SharePoint Online
Add-PnPSiteCollectionAppCatalog	Добавляет семейство сайтов в рамках каталога приложений к сайту	SharePoint Online
Remove-PnPSiteCollectionAppCatalog	Удаляет семейство сайтов в рамках каталога приложений от сайта	SharePoint Online
Add-PnPSiteDesign	Создает новый дизайн сайта для текущего арендатора	SharePoint Online
Get-PnPSiteDesign	Возвращает проекты сайтов, которые были зарегистрированы на сайте текущего арендатора	SharePoint Online
Invoke-PnPSiteDesign	Применить дизайн сайта к существующему сайту. *Требуются права администрирования арендатора	SharePoint Online
Remove-PnPSiteDesign	Удаляет проект сайта	SharePoint Online
Set-PnPSiteDesign	Изменяет проект сайта текущего арендатора	SharePoint Online
Get-PnPSiteDesignRights	Возвращает участников с правами на проектирование по конкретному проекту сайта	SharePoint Online

Продолжение таблицы 2

Grant-PnPSiteDesignRights	Предоставляет указанные права участников на использование проекта сайта	SharePoint Online
Revoke-PnPSiteDesignRights	Отменяет права указанных участников на использование проекта сайта	SharePoint Online
Get-PnPSiteDesignRun	Возвращает список проектов сайта, применяемого в указанном семействе сайтов. Если WebUrl параметр не указан, то показывается список проектов, применяемых в текущем сайте. Возвращаемый выход включает в себя идентификатор ID запланированного задания, web и сайт идентификаторы и идентификатор проекта сайта, версию и название	SharePoint Online
Get-PnPSiteDesignRunStatus	Возвращает список всех действий скрипта, выполняемого для указанного проекта сайта и применяемого к сайту	SharePoint Online

Продолжение таблицы 2

<p>Add-PnPSiteDesignTask</p>	<p>Эта команда используется для применения опубликованного проекта сайта как целевого в указанном семействе сайтов. Он планирует деятельность, позволяющую применение более большого места для скрипта (Invoke-PnPSiteDesign ограничивается 30 действиями и вспомогательными действиями). Эта команда предназначена для замены Invoke-PnPSiteDesign и используется когда нужно применить большое количество действий или несколько сценариев сайта</p>	<p>SharePoint Online</p>
------------------------------	--	--------------------------

Продолжение таблицы 2

Get-PnPSiteDesignTask	Используется для получения запланированного скрипта проекта сайта. Он принимает идентификатор запланированного проекта сайта и URL-адрес сайта, на котором находится проект сайта и планируется к применению	SharePoint Online
Remove-PnPSiteDesignTask	Удаляет задачу проектирования сайта. Если исполнение связанного скрипта сайта уже начал свое исполнение, он не будет завершен	SharePoint Online
Add-PnPSiteScript Get-PnPSiteScript	Создает новый скрипт сайта текущего арендатора Возвращает сценарии сайта, которые были зарегистрированы на сайте текущего арендатора	SharePoint Online SharePoint Online
Remove-PnPSiteScript	Удаляет сценарий сайта	SharePoint Online
Set-PnPSiteScript	Устанавливает существующий сценарий сайта текущего арендатора	SharePoint Online

Продолжение таблицы 2

Get-PnPStorageEntity	Возвращает объекты хранения / свойства фермы для любого каталога приложений арендатора или для текущего сайта, если он имеет рамки каталога приложений	SharePoint Online
Remove-PnPStorageEntity	Удаляет объекты хранения / свойства фермы от любого арендатора в рамках каталога приложений или текущего семейства сайтов	SharePoint Online
Set-PnPStorageEntity	Устанавливает объекты хранения / свойства фермы от любого арендатора в рамках каталога приложений или текущего семейства сайтов	SharePoint Online
Get-PnPTenant	Возвращает свойства семейства сайтов уровня организации	SharePoint Online
Set-PnPTenant	Устанавливает свойства семейства сайтов уровня организации	SharePoint Online
Get-PnPTenantAppCatalogUrl	Возвращает url-адрес каталога приложений с областью действия арендатора	SharePoint Online

Продолжение таблицы 2

Get-PnPTenantCdnEnabled	Возвращает, если Office 365 CDN (сеть доставки контента) был включен	SharePoint Online
Set-PnPTenantCdnEnabled	Включает или отключает государственный или частный Офис 365 CDN	SharePoint Online
Add-PnPTenantCdnOrigin	Добавляет новый источник к государственной или частной CDN	SharePoint Online
Get-PnPTenantCdnOrigin	Возвращает текущие зарегистрированные источники государственной или частной CDN	SharePoint Online
Remove-PnPTenantCdnOrigin	Удаляет источник государственной или частной CDN	SharePoint Online
Get-PnPTenantCdnPolicies	Возвращает политики CDN для указанный CDN (государственной или частной)	SharePoint Online
Set-PnPTenantCdnPolicy	Устанавливает политики CDN для указанный CDN (государственной или частной)	SharePoint Online
Clear-PnPTenantRecycleBinItem	Безвозвратно удаляет семейство сайтов из корзины арендатора	Все
Restore-PnPTenantRecycleBinItem	Восстановить семейство сайтов из корзины арендатора	SharePoint Online

Продолжение таблицы 2

Disable-PnPTenantServicePrincipal	Отключает службу текущего арендатора «Клиент SharePoint Online»	SharePoint Online
Enable-PnPTenantServicePrincipal	Включает службу текущего арендатора «Клиент SharePoint Online»	SharePoint Online
Get-PnPTenantServicePrincipal	Возвращает службу текущего арендатора «Клиент SharePoint Online»	SharePoint Online
Revoke-PnPTenantServicePrincipalPermission	Отменяет разрешение, которое ранее было предоставлено службой «Клиент SharePoint Online»	SharePoint Online
Get-PnPTenantServicePrincipalPermissionGrants	Возвращает семейство разрешений, ранее предоставленных службой «Клиент SharePoint Online»	SharePoint Online
Approve-PnPTenantServicePrincipalPermissionRequest	Утверждает разрешение на запрос службы текущего арендатора «Клиент SharePoint Online»	SharePoint Online
Deny-PnPTenantServicePrincipalPermissionRequest	Отклоняет разрешение на запрос службы текущего арендатора «Клиент SharePoint Online»	SharePoint Online

Продолжение таблицы 2

Get-PnPTenantServicePrincipalPermissionsRequests	Возвращает семейство разрешений на запрос службы текущего арендатора «Клиент SharePoint Online»	SharePoint Online
Get-PnPTenantSite	Возвращает информацию о сайте арендатора	SharePoint Online
New-PnPTenantSite	Создает новое семейство сайтов для текущего арендатора	Все
Remove-PnPTenantSite	Удаляет семейство сайтов для текущего арендатора	SharePoint Online
Set-PnPTenantSite	Установить информацию о сайте арендатора	SharePoint Online
Add-PnPTenantTheme	Добавляет или обновляет тему для арендатора	SharePoint Online
Get-PnPTenantTheme	Возвращает тему для арендатора	SharePoint Online
Remove-PnPTenantTheme	Удаляет тему для арендатора	SharePoint Online
Get-PnPTimeZoneId	Возвращает идентификатор часового пояса	Все
Get-PnPWebTemplates	Возвращает доступные web шаблоны	SharePoint Online

Продолжение таблицы 2

User and group management		
Add-PnPAlert	Добавляет предупреждение для пользователя в список	SharePoint Server 2019, SharePoint Online
Get-PnPAlert	Возвращает зарегистрированное предупреждение для пользователя	SharePoint Server 2019, SharePoint Online
Remove-PnPAlert	Удаляет предупреждение для пользователя	SharePoint Server 2019, SharePoint Online
Get-PnPGroup	Возвращает определенную группу SharePoint или все группы SharePoint на сайте	Все
New-PnPGroup	Добавляет группу в список групп сайта и возвращает объект группы	Все
Remove-PnPGroup	Удаляет группу из web	Все
Set-PnPGroup	Устанавливает группу	Все
Get-PnPGroupMembers	Возвращает всех членов группы	Все
Get-PnPGroupPermissions	Возвращает разрешения для конкретной группы SharePoint	Все
Set-PnPGroupPermissions	Добавляет и/или удаляет разрешения для конкретной группы SharePoint	Все

Продолжение таблицы 2

Get-PnPUser	Возвращает пользователей сайта текущей web	Все
New-PnPUser	Добавляет пользователя к встроенному списку сведений о пользователях сайта и возвращает объект пользователя	Все
Remove-PnPUser	Удаляет определенного пользователя сайта из списка сведений о пользователях	Все
Remove-PnPUserFromGroup	Удаляет пользователя из группы	Все
Add-PnPUserToGroup	Добавляет пользователя в группу	Все
User Profiles		
New-PnPPersonalSite	Только для Office 365: создает личный / OneDrive для бизнес-сайта	SharePoint Online
New-PnPUPABulkImportJob	Отправить новый профиль пользователя в основной части задания импортом	SharePoint Online
Get-PnPUPABulkImportStatus	Возвращает импортированный статус профиля пользователя	SharePoint Online

Продолжение таблицы 2

Get-PnPUserProfileProperty	Необходимо подключиться к администратору сайта арендатора (https://admin.sharepoint.com) с PnPOnline подключением для того, чтобы использовать этот командлет	Все
Set-PnPUserProfileProperty	Только для Office 365: использует API арендатора для получения информации о сайте. Необходимо подключение к администратору веб-сайта арендатора (https://admin.sharepoint.com) с PnPOnline подключением для того, чтобы использовать этот командлет	SharePoint Online
Utilities		
Send-PnPMail	Отправляет сообщение электронной почты с помощью службы SMTP Office 365 или SharePoint, в зависимости от указанных параметров	Все

Продолжение таблицы 2

Web Parts		
Get-PnPClientSideComponent	Возвращает один или несколько клиентских компонентов со страницы	SharePoint Server 2019, SharePoint Online
Remove-PnPClientSideComponent	Удаляет клиентский компонент со страницы	SharePoint Server 2019, SharePoint Online
Set-PnPClientSideText	Установить свойства текста клиентского компонента	SharePoint Server 2019, SharePoint Online
Set-PnPClientSideWebPart	Установить свойства web-части клиентского компонента	SharePoint Server 2019, SharePoint Online
Get-PnPWebPart	Возвращает описание объекта web-части	Все
Remove-PnPWebPart	Удаляет web-часть со страницы	Все
Get-PnPWebPartProperty	Возвращает свойства web-части	Все
Set-PnPWebPartProperty	Устанавливает свойства web-части	Все
Add-PnPWebPartToWebPartPage	Добавляет web-часть на страницу web-части в определенную зону	Все
Add-PnPWebPartToWikiPage	Добавляет web-часть на wiki страницу в определенную строку или столбец	Все
Get-PnPWebPartXml	Возвращает web-часть XML web-части, зарегистрированной на сайте	Все

Продолжение таблицы 2

Webs		
Set-PnPIndexedProperties	Устанавливает значения пакета свойств, которые должны быть индексированы для поиска. Обратите внимание, что будут перезаписаны существующие флаги, т. е. только свойства, определяемые с помощью командлета будут индексироваться	Все
Add-PnPIndexedProperty	Устанавливает значение ключа пакета свойств, определенной индексации для поиска	Все
Remove-PnPIndexedProperty	Удаляет ключ из пакета свойств, который должен быть индексирован для поиска. Ключ и его значение остаются в пакете свойств, однако он больше не будет индексироваться	Все
Get-PnPIndexedPropertyKeys	Возвращает ключи значений пакета свойств, которые были отмечены для поисковой индексации	Все

Продолжение таблицы 2

Get-PnPPropertyBag	Возвращает значения пакета свойств	Все
Remove-PnPPropertyBagValue	Удаляет значение из пакета свойств	Все
Set-PnPPropertyBagValue	Устанавливает значение свойств пакета	Все
Request-PnPReIndexWeb	Отмечает web для полной индексации во время следующего инкрементного обхода	Все
Get-PnPRequestAccessEmails	Возвращает запрос доступа к электронной почте	SharePoint Online
Set-PnPRequestAccessEmails	Устанавливает запрос доступа к электронной почте web	SharePoint Online
Get-PnPSubWebs	Возвращает вложенные веб-страницы текущего веб-сайта	Все
Get-PnPWeb	Возвращает текущий веб-объект	Все
New-PnPWeb	Создает новую вложенную веб-страницу текущего веб-сайта	Все
Remove-PnPWeb	Удаляет вложенную веб-страницу текущего веб-сайта	Все
Set-PnPWeb	Устанавливает свойства web	Все

Окончание таблицы 2

Invoke-PnPWebAction	Выполняет операции в web, списках, элементах списка	Все
Set-PnPWebPermission	Устанавливает разрешения	Все
Workflows		
Add-PnPWorkflowDefinition	Добавляет определение рабочего процесса	Все
Get-PnPWorkflowDefinition	Возвращает определение рабочего процесса	Все
Remove-PnPWorkflowDefinition	Удаляет определение рабочего процесса	Все
Get-PnPWorkflowInstance	Возвращает экземпляр рабочего процесса	Все
Resume-PnPWorkflowInstance	Возобновить рабочий процесс	Все
Start-PnPWorkflowInstance	Запускает экземпляр рабочего процесса для элемента списка	Все
Stop-PnPWorkflowInstance	Останавливает экземпляр рабочего процесса для элемента списка	Все
Add-PnPWorkflowSubscription	Добавляет подписку на рабочий процесс в список	Все
Get-PnPWorkflowSubscription	Возвращает подписку на рабочий процесс в списке	Все
Remove-PnPWorkflowSubscription	Удаляет подписку на рабочий процесс	Все

Далее будет рассмотрен пример создания списков на сайте.

Создание списков на сайте SharePoint

С помощью создания скриптов можно создавать на сайте SharePoint любые списки.

Например, имеется SharePoint сайт, предназначенный для поддержки работы библиотеки.

Сначала необходимо создать на сайте списки для хранения данных:

– «Подписка на журналы и газеты» – список журналов, на которые у библиотеки имеются подписки;

– «Поступление журналов» – список для учета новых поступлений журналов.

Структура списка «Подписка на журналы и газеты» представлена в таблице 3.

Структура списка «Поступление журналов» представлена в таблице 4.

Таблица 3 – Структура списка «Подписка на журналы и газеты»

Индекс издания	Однострочный текст
Наименование издания	Однострочный текст Выбор из списка: – иностранные издания; – отечественные издания
Изменено	Дата и время
Создано	Дата и время
Кем создано	Пользователь или группа
Кем изменено	Пользователь или группа

Таблица 4 – Структура списка «Поступление журналов»

Журнал	Подстановка из списка «Подписка на журналы и газеты» столбец «Наименование издания»
Выпуск	Однострочный текст
Дата поступления	Дата и время
Название	Однострочный текст
АссертYear	Вычисляемый (=ГОД(СЕГОДНЯ()-ГОД([Дата поступления]))
Изменено	Дата и время
Создано	Дата и время
Кем создано	Пользователь или группа
Кем изменено	Пользователь или группа

Сотрудники библиотеки имеют права членов узла, то есть удаленно, обращаясь через браузер, могут редактировать данные, хранимые в этих списках. Прав для управления списками у них нет. Они не могут изменять структуру таблиц, скрипты, выполняющиеся на стороне сервера.

В соответствии с внутренним регламентом работы, например, каждую среду, на стендах библиотеки должны быть вывешены объявления, содержащие списки последних поступлений журналов.

Рассмотрим вариант автоматизации формирования списка новых поступлений с использованием веб-служб SharePoint из PowerShell. Список будем формировать в среде MS Word.

Шаг 1. Предварительные операции.

Создадим файл документа MS Word, задав ему, например, имя «Новые поступления.docx», содержащий заголовок и таблицу подобную таблице 3.

Это будет файл-шаблон для текста объявления. В дальнейшем в разрабатываемом скрипте также будет использована дата последнего изменения этого файла для формирования поискового запроса.

Шаг 2. Среда PowerShell и её настройка.

Права на выполнение неподписанных скриптов

Если PowerShell скрипты ещё не запускались, на вашей системе используется политика безопасности, определяемая по умолчанию при её установке, в которой выполнение скриптов запрещено.

Узнать текущие разрешения на выполнение в среде PowerShell в вашей системе можно выполнив команду (правильнее называть командлет):

```
Get-ExecutionPolicy
```

Если результат выполнения этого командлета имеет значение Restricted, это означает что запуск любых скриптов запрещен.

Для того что бы разрешить выполнение скриптов, тем более не подписанных цифровой подписью, нужно изменить настройку политики.

Для этого следует открыть среду PowerShell с правами администрирования и выполнить командлет Set-ExecutionPolicy, с параметрами, разрешающими исполнять не подписанные скрипты:

```
Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope CurrentUser
```

Более подробно о этих командлетах и правах на выполнение скриптов можно узнать по следующим адресам:

<https://www.osp.ru/winitpro/2011/12/13014396/>

<https://www.darkoperator.com/blog/2013/3/5/powershell-basics-execution-policy-part-1.html>

Лист скрипта приведен в приложении 1.

Собственно, скрипт

В переменную \$MSGFile, в виде простой текстовой строки, поместим полный путь к файлу «Новые поступления.docx», например:

```
$MSGFile = "C:\Users\Student13\Обслуживание  
библиотеки\Новые поступления.docx"
```

Проверим, существует ли файл-шаблон объявления:

```
Test-Path $MSGFile
```

Если файл существует, продолжаем работу, иначе или прекращаем выполнение скрипта, сформировав и отобразив сообщение о возникшей ошибке, или создаем требуемый файл в скрипте и продолжаем его работу штатно (в этом случае надо определить какая дата будет считаться датой, начиная с которой надо будет отбирать записи вместо даты последнего изменения списка).

Так как, в дальнейшем будет использована объектная модель MS Word, следует проверить наличие на компьютере этого продукта:

```
Test-Path HKLM:\SOFTWARE\Classes\Word.Application
```

Если файл существует, подготовим другие необходимые на этом этапе переменные.

Адрес веб-службы SharePoint:

```
$SiteURL = "http://www.imm.uran.ru/rus/Library"
```

```
$uri = $SiteURL + "/_vti_bin/Lists.asmx?wsdl"
```

Проверим, есть ли доступ к серверу SharePoint.

Если доступ к сайту отсутствует, то заканчиваем работу скрипта с соответствующим уведомлением. Иначе готовим данные для обращения к веб-сервису.

Здесь потребуются следующие переменные.

Имя списка, из которого будут отбираться данные:

```
$ListName = "Поступление журналов"
```

Диапазон дат для поискового запроса:

```
$PrewDate = Get-Date ((Get-ChildItem $MSGFile).LastWriteTime)
```

```
-format s
```

```
$CurrentDate = Get-Date -format s
```

Запрос:

```
$xmlDoc = new-object System.Xml.XmlDocument
```

```
$query = $xmlDoc.CreateElement("Query")
```

Запрос формируем на языке CAML, и здесь имеется одна известная трудность. Поля списков и библиотек SharePoint, как и ряд других сущностей, имеют несколько различных

идентификаторов, из которых для данной задачи важны отображаемое имя и внутреннее имя. Эти имена могут отличаться. Формируя SAML-запрос требуется указывать внутренние имена полей.

При создании поля через web-интерфейс задается отображаемое имя, которое автоматически преобразуется в имя внутреннее. Если при создании поле называют по-русски (отображаемое имя), то внутреннее имя поля будет сформировано специальным способом с преобразованием самих символов в текстовые значения их шестнадцатичных кодов, которым предшествуют символ подчеркивания и символ x, и последовательность завершает также символ подчеркивания, например, поле с отображаемым именем "Дата поступления" будет иметь внутреннее имя `_x0414_x0430_x0442_x0430_x00`, а поле "Выпуск" – `_x041f_x043e_x0441_x0442_x04`.

Вычислим внутреннее имя поля средствами PowerShell.

Для этого используем командлет Format-Hex. В результате выполнения такого командлета:

```
"Дата поступления" | Format-Hex -Encoding BigEndianUnicode
```

получаем набор нужных нам шестнадцатичных значений:

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
```

```
00000000 04 14 04 30 04 42 04 30 00 20 04 3F 04 3E 04 41  
...0.B.0. .?>.A
```

```
00000010 04 42 04 43 04 3F 04 3B 04 35 04 3D 04 38 04 4F  
.B.C.?.;5.=.8.O
```

Преобразуем результат в строку:

```
("Дата поступления" | Format-Hex -Encoding  
BigEndianUnicode).ToString()
```

И преобразуем строку в массив, разбив на элементы по символам пробела:

```
$internalFieldName = ("Дата поступления" | Format-Hex -  
Encoding BigEndianUnicode).ToString() -split(" ")
```

Из получившегося массива берем элементы с 3 по 11 и собираем в нужное нам имя:

```
$internalFieldName = `
"_x" + $internalFieldName[3] + $internalFieldName[4] + "_" + `
"_x" + $internalFieldName[5] + $internalFieldName[6] + "_" + `
"_x" + $internalFieldName[7] + $internalFieldName[8] + "_" + `
"_x" + $internalFieldName[9] + $internalFieldName[10] + "_" + `
"_x" + $internalFieldName[11]
```

В результате получаем внутреннее имя атрибута:

```
_x0414_x0430_x0442_x0430_x00
```

Замечание 1

Обычно можно найти другой способ получения внутренних имен. Он состоит в следующем. Открываем в веб-интерфейсе параметры списка, кликаем на имени столбца и переходим на страницу «Изменение столбца». Смотрим на адрес в адресной строке. В адресе выделяем значение параметра Field. В нем, заменяем символы %5F на символ подчеркивание _ и получаем внутреннее имя данного поля.

Замечание 2

Чтобы данная проблема не возникала, следует при создании задавать имена полей на латинице без пробелов. Внутренние имена в таком случае будут такие же, как и заданные отображаемые. После этого полям можно сразу дать функциональные русскоязычные имена удобные пользователям, при этом внутренние имена останутся прежними.

Действия по формированию внутренних имен можно произвести один раз вручную и использовать в скрипте далее полученные значения.

Прежде чем формировать требуемое объявление, проверим диапазон дат, если сегодня уже формировался список поступлений, снова его создавать не будем:

```
if((Get-Date $CurrentDate -format d) -eq (Get-Date $PrewDate -
format d)) {
```

Окончательно формируем CAML-запрос на отбор записей о новых поступлениях в библиотеку:

```
$QueryCAML = '<Where><Eq><FieldRef  
Name="_x0414__x0430__x0442__x0430__x00" /><Value  
IncludeTimeValue="FALSE" Type="DateTime">' + $CurrentDate +  
'</Value></Eq></Where><OrderBy><FieldRef  
Name="_x0414__x0430__x0442__x0430__x00"/></OrderBy>'  
else {$QueryCAML = '<Where><And><Leq><FieldRef  
Name="_x0414__x0430__x0442__x0430__x00" /><Value  
IncludeTimeValue="FALSE" Type="DateTime">' + $CurrentDate +  
'</Value></Leq><Gt><FieldRef  
Name="_x0414__x0430__x0442__x0430__x00" /><Value  
IncludeTimeValue="FALSE" Type="DateTime">' + $PrewDate +  
'</Value></Gt></And></Where><OrderBy><FieldRef  
Name="_x0414__x0430__x0442__x0430__x00"/></OrderBy>'  
$query.set_InnerXml("$QueryCAML")
```

Теперь можно окончательно подготовить остальные параметры запроса. Возвращаемые в ответе на запрос столбцы (поля) списка, по умолчанию – все:

```
$viewFields = $xmlDoc.CreateElement("ViewFields")
```

Некоторые другие параметры оставляем в значении по умолчанию:

```
$queryOptions = $xmlDoc.CreateElement("QueryOptions")
```

В ответе на запрос пусть будет возвращено не более 1000 записей:

```
$rowLimit = "1000"
```

При обращении к веб-службам SharePoint обязательно требуется предъявить пользовательский идентификатор и соответствующий ему пароль.

Будем считать, что клиентский компьютер находится вне доменной сети, и пользовательский идентификатор, и пароль текущего пользователя, под учетной записью, которого будет выполняться скрипт автоматизации, не могут быть использованы для авторизации при обращении к веб-службам SharePoint. В скрипте надо предусмотреть код, выполняющий формирование

учетных данных для доступа к службам и их хранение. Понятно, что по соображениям безопасности нельзя хранить пароль в открытом виде. Поэтому зашифруем его. Пользовательский идентификатор и пароль не требуется сохранять при каждом запуске скрипта автоматизации.

Это можно сделать один раз при настройке скрипта. В дальнейшем может лишь понадобится изменение значения пароля.

Пароль можно запросить непосредственно у пользователя командой:

```
$SecurePassword = Read-Host -Prompt "Enterpassword" -  
AsSecureString
```

Или использовать текстовую строку с паролем в открытом виде:

```
$PlainPassword = "P@ssw0rd"  
$SecurePassword = $PlainPassword | ConvertTo-SecureString -  
AsPlainText -Force
```

Или одной строкой:

```
$SecurePassword = "P@ssw0rd" | ConvertTo-SecureString -  
AsPlainText -Force
```

В результате получится объект типа
System.Security.SecureString.

Для сохранения этот объект следует преобразовать к текстовому виду с помощью команды ConvertFrom-SecureString.

Таким образом, получаем следующий конвейер:

```
$SecureStringAsPlainText = "P@ssw0rd" | ConvertTo-SecureString  
-AsPlainText -Force | ConvertFrom-SecureString
```

Результатом его выполнения (значением переменной \$SecureStringAsPlainText) будет текстовая строка со значением похожим на следующее:

```
01000000d08c9ddf0115d1118c7a00c04fc297eb010000008f6d37a8  
4c5b6d40904e0e691c3803d200000000020000000000106600000010  
0002000000068811105fd2ca1773c8063f39e97ab1ff35c932eff073a88c  
09feedb155e98a3000000000e80000000020000200000008a838c08e8b  
24c8626a205afe0379cbe7ddce1a0e80b7441cbdd1930961bc1c8200000
```

```
009d14b066035f211223528b3ca71299d652861879a2f555af43a17bcd7
383e95740000000335666eb285e12a592381c2a98ee749ba5e692c8e00
9bcd09ce67a909e25279cf5f07b5cb84c080cdedcac5ac22ea09db7f1e3e
ae238c8250b86fafc6ac21ee0
```

Зашифрованный пароль надо где-то сохранить. Например, его можно сохранить в текстовом файле, даже просто в тексте вашего скрипта. Однако, снова по соображениям безопасности не следует хранить пароль просто в текстовом файле. Немного более безопасно сохранить его в реестре, в ветке текущего пользователя HKCU:\, таким образом пароль будет доступен для расшифровки только под той же учетной записью, под которой он был сохранен.

Для сохранения возьмем следующий путь в реестре HKCU:\Software\SClibrary\Credentials.

Пользовательский идентификатор и пароль поместим соответственно в свойства UserName и Password.

Прежде всего, проверим существует ли уже такая ветвь реестра выполнив команду:

```
!(Test-Path "HKCU:\Software\SClibrary\Credentials")
```

Если указанного пути нет, следует создать соответствующую ветвь реестра исполнив команду:

```
New-Item -Path "HKCU:\Software\SClibrary" -Name "Credentials"
-Force
```

Присваиваем свойствам UserName и Password соответствующие значения:

```
New-ItemProperty -Path HKCU:\Software\SClibrary\Credentials -
PropertyType String -Name UserName -Value
"Домен\Пользовательский идентификатор" -Force
New-ItemProperty -Path HKCU:\Software\SClibrary\Credentials -
PropertyType String -Name Password -Value
$SecureStringAsPlainText -Force
```

Проверим сохранение свойств, прочитав их. Для этого выполним команды:

```
$UserName = (Get-ItemProperty -Path
HKCU:\Software\SClibrary\Credentials).UserName
```

```
$SecureStringAsPlainText = (Get-ItemProperty -Path  
HKCU:\Software\SClibrary\Credentials).Password
```

Прочитанное из реестра значение пароля подготовим для использования:

```
$securePassword = ConvertTo-SecureString  
$SecureStringAsPlainText
```

И окончательно создадим объект, который будет использован для аутентификации при обращении к веб-службам SharePoint:

```
$credential = New-Object -TypeName  
System.Management.Automation.PSCredential -ArgumentList  
"Домен\Пользовательский идентификатор", $securePassword
```

Замечание 3

Если понадобится изменить пароль, для этого следует применить командлет:

```
Set-ItemProperty -Path HKCU:\Software\SClibrary\Credentials -  
Name Password -Value $SecureStringAsPlainText -Force
```

Для удаления значений свойств разделов реестра (очистки) надо применить командлет:

```
Remove-ItemProperty -Path  
HKCU:\Software\SClibrary\Credentials -Name Password -  
Confirm:$false -ErrorAction SilentlyContinue
```

Замечание 4

Аналогично в реестре можно сохранить дату последнего формирования списка новых поступлений

Теперь можно обратиться к веб-службе SharePoint:

```
$service = New-WebServiceProxy -Uri $uri -Namespace SpWs -  
Credential $credential -ErrorAction Stop
```

Получаем данные из списка:

```
$List = $service.GetListItems($listName, "", $query, $viewFields,  
$rowLimit, $queryOptions, "")
```

Обратите внимание, имена полей списка в результате выборки имеют дополнительный префикс ows_.

Например:

Дата поступления - ows__x0414__x0430__x0442__x0430__x00

Выпуск - ows__x041f__x043e__x0441__x0442__x04

Проверяем сколько записей из списка сервер вернул в ответ на запрос:

```
$Test = $($List.data.row | Measure-Object).Count
```

Если результат запроса не пустой, разбираем его и формируем документ MS Word с текстом объявления. Для формирования документа используем объектную модель MS Word:

```
$Test -gt 0
```

Открываем MS Word:

```
$word = new-object -ComObject "Word.Application"
```

Устанавливаем параметр видимости в значение «истина» для того, что бы наблюдать работу скрипта:

```
$word.Visible = $TRUE
```

При автоматическом исполнении этот параметр следует установить значение «ложь».

Открываем уже имеющийся документ-шаблон и далее используем объектную модель MS Word для формирования объявления:

```
$docu = $word.Documents.Open("$MSGFile")
```

Если в таблице шаблона строк меньше, чем найденных записей, надо добавить строки. В документе у нас имеется таблица, в списке объектов она будет под номером 1 (проверьте это):

```
$LETable = $docu.Tables.Item(1)
```

Узнаем число столбцов в таблице. Далее это значение нам потребуется для перемещения по столбцам при помещении в столбцы соответствующих данных:

```
$LETableCols = $LETable.Columns.Count
```

Узнаем число строк в таблице:

```
$LETableRows = $LETable.Rows.Count
```

Проведем проверку и, если строк в таблице документа недостаточно для размещения всех элементов из списка новых поступлений, добавляем строки:

```
if($Test -gt $LETableRows)
{
```

Если найдено строк больше, чем строк в таблице, надо добавить строки в цикле следующим образом:

```
For ($i=1; $i -le $($Test - $LETableRows); $i++) {$RC =
$LETable.Rows.Add()}
}
```

В том случае, если строк таблицы достаточно, проходим по всем элементам результата обращения к сервису и формируем таблицу со списком новых поступлений объявления:

```
$r = 1
foreach($Item in $List.data.row)
{
    if(1 -eq $r)
    {
```

При первой итерации изменяем дату в заголовке документа:

```
$Header = "Дата поступления" + " " + (Get-Date
$(($Item.ows__x0414__x0430__x0442__x0430__x00) -format D) +
" `n"
```

Оформляем текст, явно указывая параметры шрифта:

```
$docu.Paragraphs.item(1).Range.Font.Size = 14
$docu.Paragraphs.item(1).Range.Font.Name = "Times
New Roman"
$docu.Paragraphs.item(1).Range.Text = $Header
$docu.Paragraphs.item(1).Alignment = 1
}
```

Заполняем ячейки таблицы:

```
$LETable.Cell($r,1).Range.Text =
$(($Item.ows_Journal.split('#'))[1]
$LETable.Cell($r,1).Range.Font.Size = 14
```

```
$LETable.Cell($r,1).Range.Font.Name = "Times New Roman"
```

```
$LETable.Cell($r,2).Range.Text =  
$Item.ows__x041f__x043e__x0441__x0442__x04  
$LETable.Cell($r,2).Range.Font.Size = 14  
$LETable.Cell($r,2).Range.Font.Name = "Times New Roman"
```

Наращиваем счетчик строк:

```
$r = $r + 1
```

```
}
```

Если результат запроса содержит меньше элементов, чем имеется строк в таблице документа, удаляем из таблицы лишние строки:

```
For ($i=$LETableRows; $i -Ge $r; $i--)  
{ $LETable.Rows.Item($i).Delete() }
```

Принудительно обновляем форматирование (оформление) таблицы:

```
$LETable.UpdateAutoFormat()
```

Текст документа готов. Сохраняем документ:

```
$docu.Save()
```

Закрываем документ:

```
$docu.Close()
```

Завершаем работу MS Word:

```
$word.Quit()
```

Теперь можно распечатать документ:

```
Start-Process -FilePath "$MSGFile" -Verb print
```

В этом командлете можно указать принтер, отличный от установленного по умолчанию, на котором следует распечатать документ. Разберитесь как это сделать.

Замечание 5

Известна ошибка, из-за которой работа MS Word, запущенного из PowerShell, может не быть корректно завершена. Для окончательного завершения MS Word выполним следующие действия:

```
$rc =  
[System.Runtime.InteropServices.Marshal]::ReleaseComObject($word)  
}
```

Скрипт создания сайтов и учетных записей студентов

Исходные данные для группы ЭМ-421606

ЭМ-421606 1234Ab
"Буторина Наталья"
"Гагарин Вадим"
"Глызин Максим"
"Ихсанова Вероника"
"Курдюков Алексей"
"Макагонов Никита"
"Матвеевнина Арина"
"Меньшиков Владислав"
"Наумова Мария"
"Никешин Григорий"
"Пантюшина Мария"
"Пастухов Илья"
"Патшин Максим"
"Расковалов Андрей"
"Распопова Дарья"
"Родина Мария"
"Росс Анастасия"
"Фролова Наталья"
"Чечулина Ксения"
"Чупракова Татьяна"
"Шерстенников Никита"
"Юрочкина Елена"

Исходные данные для группы ЭМ-421603

ЭМ-421603 421603Ab
"Балабаев"
"Гилев"
"Ихтиар"
"Калачев"
"Коваленко"
"Немирович-Данченко"
"Подкорытов"
"Покрышкин"
"Русаков"
"Рябов"
"Сорокин"
"Трубин"
"Чикуров"
"Чуваков"
"Шайбакова"
"Шухарт"

CreateUsers

```
$Students =  
@(  
[pscustomobject]@{Id=1; Name="Буторина Наталья";  
Group=421606;info="1234Ab"},  
[pscustomobject]@{Id=2; Name="Гагарин Вадим";  
Group=421606;info="1234Ab"},  
[pscustomobject]@{Id=3; Name="Глызин Максим";  
Group=421606;info="1234Ab"},  
[pscustomobject]@{Id=4; Name="Ихсанова Вероника";  
Group=421606;info="1234Ab"},  
[pscustomobject]@{Id=5; Name="Курдюков Алексей";  
Group=421606;info="1234Ab"},  
[pscustomobject]@{Id=6; Name="Макагонов Никита";  
Group=421606;info="1234Ab"},  
[pscustomobject]@{Id=7; Name="Матвеевнина Арина";  
Group=421606;info="1234Ab"},  
[pscustomobject]@{Id=8; Name="Меньшиков  
Владислав";Group=421606;info="1234Ab"},
```

[pscustomobject]@{Id=9; Name="Наумова Мария";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=10;Name="Никешин Григорий";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=11;Name="Пантюшина Мария";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=12;Name="Пастухов Илья";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=13;Name="Патшин Максим";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=14;Name="Расковалов Андрей";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=15;Name="Распопова Дарья";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=16;Name="Родина Мария";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=17;Name="Росс Анастасия";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=18;Name="Фролова Наталья";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=19;Name="Чечулина Ксения";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=20;Name="Чупракова Татьяна";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=21;Name="Шерстенников
Никита";Group=421606;info="1234Ab"},
[pscustomobject]@{Id=22;Name="Юрочкина Елена";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=1; Name="Балабаев";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=2; Name="Гилев";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=3; Name="Ихтиар";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=4; Name="Калачёв";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=5; Name="Коваленко";
Group=421603;info="421603Ab"},

```

[pscustomobject]@{Id=6; Name="Немирович-Данченко";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=7; Name="Подкорытов";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=8; Name="Покрышкин";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=9; Name="Русаков";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=10;Name="Рябов";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=11;Name="Сорокин";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=12;Name="Трубин";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=13;Name="Чикуров";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=14;Name="Чуваков";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=15;Name="Шайбакова";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=16;Name="Шухарт";
Group=421603;info="421603Ab"}
)
foreach ($Student in $Students)
{
    $UserID = "S" + "$($Student.Group)".Substring(4,2) + "-" +
$Student.ID
    $PassWord = "$($Student.info)" + "$($Student.Id)"
    $SPSURL = "http://sp-vm-students.cloudapp.net/sites/" +
$UserID
Write-Host
"$($Student.Id)`t$($Student.Name)`t$UserID`t$PassWord`t$SPSURL
"

    $scn = [ADSI]"WinNT://SP-VM-Students"
    $user = $scn.Create("User","$UserID")
    $user.SetPassword("$PassWord")
    $user.setinfo()
    $user.FullName = "$($Student.Name)"
    $user.setinfo()

```

```
$user.description = "Group $($Student.Group) SharePoint Student
$($Student.Id)"
$user.SetInfo()

}
```

CreateSharePointSites

Add-PSSnapin Microsoft.SharePoint.Powershell

```
# Get-SPWebTemplate
```

```
$template = Get-SPWebTemplate "STS#0"
```

```
$Students =
```

```
@(
[pscustomobject]@{Id=1; Name="Буторина Наталья";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=2; Name="Гагарин Вадим";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=3; Name="Глызин Максим";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=4; Name="Ихсанова Вероника";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=5; Name="Курдюков Алексей";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=6; Name="Макагонов Никита";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=7; Name="Матвеевна Арина";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=8; Name="Меньшиков
Владислав";Group=421606;info="1234Ab"},
[pscustomobject]@{Id=9; Name="Наумова Мария";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=10;Name="Никешин Григорий";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=11;Name="Пантюшина Мария";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=12;Name="Пастухов Илья";
Group=421606;info="1234Ab"},
```

[pscustomobject]@{Id=13;Name="Патшин Максим";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=14;Name="Расковалов Андрей";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=15;Name="Распопова Дарья";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=16;Name="Родина Мария";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=17;Name="Росс Анастасия";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=18;Name="Фролова Наталья";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=19;Name="Чечулина Ксения";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=20;Name="Чупракова Татьяна";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=21;Name="Шерстенников
Никита";Group=421606;info="1234Ab"},
[pscustomobject]@{Id=22;Name="Юрочкина Елена";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=1; Name="Балабаев";
Group=421603;info="421603"},
[pscustomobject]@{Id=2; Name="Гилев";
Group=421603;info="421603"},
[pscustomobject]@{Id=3; Name="Ихтиар";
Group=421603;info="421603"},
[pscustomobject]@{Id=4; Name="Калачёв";
Group=421603;info="421603"},
[pscustomobject]@{Id=5; Name="Коваленко";
Group=421603;info="421603"},
[pscustomobject]@{Id=6; Name="Немирович-Данченко";
Group=421603;info="421603"},
[pscustomobject]@{Id=7; Name="Подкорытов";
Group=421603;info="421603"},
[pscustomobject]@{Id=8; Name="Покрышкин";
Group=421603;info="421603"},
[pscustomobject]@{Id=9; Name="Русаков";
Group=421603;info="421603"},

```

[pscustomobject]@{Id=10;Name="Рябов";
Group=421603;info="421603"},
[pscustomobject]@{Id=11;Name="Сорокин";
Group=421603;info="421603"},
[pscustomobject]@{Id=12;Name="Трубин";
Group=421603;info="421603"},
[pscustomobject]@{Id=13;Name="Чикуров";
Group=421603;info="421603"},
[pscustomobject]@{Id=14;Name="Чуваков";
Group=421603;info="421603"},
[pscustomobject]@{Id=15;Name="Шайбакова";
Group=421603;info="421603"},
[pscustomobject]@{Id=16;Name="Шухарт";
Group=421603;info="421603"}
)
foreach ($Student in $Students)
{
    $UserID = "S" + "$($Student.Group)".Substring(4,2) + "-" +
$Student.ID
    $PassWord = "$($Student.info)" + "$($Student.Id)"
    $SPSURL = "http://sp-vm-students.cloudapp.net/sites/" +
$UserID
Write-Host
"$($Student.Id)`t$($Student.Name)`t$UserID`t$PassWord`t$SPSURL
"

    New-SPSite -Url "http://sp-vm-students/sites/$UserID" -
OwnerAlias "sp-vm-students\$UserID" -Language 1049 -Template
$template
}

```

SharePoint Students List

1	Буторина Наталья	sp-vm- students\S06-1	1234Ab1	http://sp-vm- students.cloudapp.net/sites/S06-1
2	Гагарин Вадим	sp-vm- students\S06-2	1234Ab2	http://sp-vm- students.cloudapp.net/sites/S06-2
3	Глызин Максим	sp-vm- students\S06-3	1234Ab3	http://sp-vm- students.cloudapp.net/sites/S06-3
4	Ихсанова Вероника	sp-vm- students\S06-4	1234Ab4	http://sp-vm- students.cloudapp.net/sites/S06-4
5	Курдюков Алексей	sp-vm- students\S06-5	1234Ab5	http://sp-vm- students.cloudapp.net/sites/S06-5
6	Макагонов Никита	sp-vm- students\S06-6	1234Ab6	http://sp-vm- students.cloudapp.net/sites/S06-6
7	Матвевни- на Арина	sp-vm- students\S06-7	1234Ab7	http://sp-vm- students.cloudapp.net/sites/S06-7
8	Меньши- ков Владислав	sp-vm- students\S06-8	1234Ab8	http://sp-vm- students.cloudapp.net/sites/S06-8
9	Наумова Мария	sp-vm- students\S06-9	1234Ab9	http://sp-vm- students.cloudapp.net/sites/S06-9
10	Никешин Григорий	sp-vm- students\S06- 10	1234Ab10	http://sp-vm- students.cloudapp.net/sites/S06- 10
11	Пантюши- на Мария	sp-vm- students\S06- 11	1234Ab11	http://sp-vm- students.cloudapp.net/sites/S06- 11
12	Пастухов Илья	sp-vm- students\S06- 12	1234Ab12	http://sp-vm- students.cloudapp.net/sites/S06- 12
13	Патшин Максим	sp-vm- students\S06- 13	1234Ab13	http://sp-vm- students.cloudapp.net/sites/S06- 13
14	Расковалов Андрей	sp-vm- students\S06- 14	1234Ab14	http://sp-vm- students.cloudapp.net/sites/S06- 14
15	Распопова Дарья	sp-vm- students\S06- 15	1234Ab15	http://sp-vm- students.cloudapp.net/sites/S06- 15
16	Родина Мария	sp-vm- students\S06- 16	1234Ab16	http://sp-vm- students.cloudapp.net/sites/S06- 16
17	Росс Анастасия	sp-vm- students\S06-	1234Ab17	http://sp-vm- students.cloudapp.net/sites/S06-

		17		17
18	Фролова Наталья	sp-vm- students\S06- 18	1234Ab18	http://sp-vm- students.cloudapp.net/sites/S06- 18
19	Чечулина Ксения	sp-vm- students\S06- 19	1234Ab19	http://sp-vm- students.cloudapp.net/sites/S06- 19
20	Чупракова Татьяна	sp-vm- students\S06- 20	1234Ab20	http://sp-vm- students.cloudapp.net/sites/S06- 20
21	Шерстен- ников Никита	sp-vm- students\S06- 21	1234Ab21	http://sp-vm- students.cloudapp.net/sites/S06- 21
22	Юрочкина Елена	sp-vm- students\S06- 22	1234Ab22	http://sp-vm- students.cloudapp.net/sites/S06- 22
1	Балабаев	sp-vm- students\S03-1	421603Ab 1	http://sp-vm- students.cloudapp.net/sites/S03- -1
2	Гилев	sp-vm- students\S03-2	421603Ab 2	http://sp-vm- students.cloudapp.net/sites/S03- -2
3	Ихтиар	sp-vm- students\S03-3	421603Ab 3	http://sp-vm- students.cloudapp.net/sites/S03- -3
4	Калачёв	sp-vm- students\S03-4	421603Ab 4	http://sp-vm- students.cloudapp.net/sites/S03- -4
5	Коваленко	sp-vm- students\S03-5	421603Ab 5	http://sp-vm- students.cloudapp.net/sites/S03- -5
6	Немирович -Данченко	sp-vm- students\S03-6	421603Ab 6	http://sp-vm- students.cloudapp.net/sites/S03- -6
7	Подкоры- тов	sp-vm- students\S03-7	421603Ab 7	http://sp-vm- students.cloudapp.net/sites/S03- -7
8	Покрыш- кин	sp-vm- students\S03-8	421603Ab 8	http://sp-vm- students.cloudapp.net/sites/S03- -8
9	Русаков	sp-vm- students\S03-9	421603Ab 9	http://sp-vm- students.cloudapp.net/sites/S03- -9
10	Рябов	sp-vm-	421603Ab	http://sp-vm-

		students\S03-10	10	students.cloudapp.net/sites/S03-10
11	Сорокин	sp-vm-students\S03-11	421603Ab 11	http://sp-vm-students.cloudapp.net/sites/S03-11
12	Трубин	sp-vm-students\S03-12	421603Ab 12	http://sp-vm-students.cloudapp.net/sites/S03-12
13	Чикуров	sp-vm-students\S03-13	421603Ab 13	http://sp-vm-students.cloudapp.net/sites/S03-13
14	Чуваков	sp-vm-students\S03-14	421603Ab 14	http://sp-vm-students.cloudapp.net/sites/S03-14
15	Шайбакова	sp-vm-students\S03-15	421603Ab 15	http://sp-vm-students.cloudapp.net/sites/S03-15
16	Шухарт	sp-vm-students\S03-16	421603Ab 16	http://sp-vm-students.cloudapp.net/sites/S03-16

ModUsers

\$Students =

```
@(
[pscustomobject]@{Id=1; Name="Буторина Наталья";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=2; Name="Гагарин Вадим";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=3; Name="Глызин Максим";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=4; Name="Ихсанова Вероника";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=5; Name="Курдюков Алексей";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=6; Name="Макагонов Никита";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=7; Name="Матвеевна Арина";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=8; Name="Меньшиков
Владислав";Group=421606;info="1234Ab"},
```

[pscustomobject]@{Id=9; Name="Наумова Мария";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=10;Name="Никешин Григорий";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=11;Name="Пантюшина Мария";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=12;Name="Пастухов Илья";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=13;Name="Патшин Максим";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=14;Name="Расковалов Андрей";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=15;Name="Распопова Дарья";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=16;Name="Родина Мария";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=17;Name="Росс Анастасия";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=18;Name="Фролова Наталья";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=19;Name="Чечулина Ксения";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=20;Name="Чупракова Татьяна";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=21;Name="Шерстенников
Никита";Group=421606;info="1234Ab"},
[pscustomobject]@{Id=22;Name="Юрочкина Елена";
Group=421606;info="1234Ab"},
[pscustomobject]@{Id=1; Name="Балабаев";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=2; Name="Гилев";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=3; Name="Ихтиар";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=4; Name="Калачёв";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=5; Name="Коваленко";
Group=421603;info="421603Ab"},

```
[pscustomobject]@{Id=6; Name="Немирович-Данченко";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=7; Name="Подкорытов";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=8; Name="Покрышкин";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=9; Name="Русаков";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=10;Name="Рябов";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=11;Name="Сорокин";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=12;Name="Трубин";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=13;Name="Чикуров";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=14;Name="Чуваков";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=15;Name="Шайбакова";
Group=421603;info="421603Ab"},
[pscustomobject]@{Id=16;Name="Шухарт";
Group=421603;info="421603Ab"}
)
```

```
foreach ($Student in $Students)
{
    $UserID = "S" + "$($Student.Group)".Substring(4,2) + "-" +
$Student.ID
    $PassWord = "$($Student.info)" + "$($Student.Id)"
    $SPSURL = "http://sp-vm-students.cloudapp.net/sites/" +
$UserID
Write-Host
"$($Student.Id)`t$($Student.Name)`t$UserID`t$PassWord`t$SPSURL
"
```

```
$cn = [ADSI]"WinNT://SP-VM-Students"
$User = $cn.psbases.Children.Find("$UserID")
```

```
$user.SetPassword("P@assWord15")
```

```

$user.setinfo()

# Password never expired
$flag = $user.UserFlags.Value -bor 0x10000
$user.Put("userflags",$flag)
$user.SetInfo()

$user.SetPassword("$PassWord")
$user.setinfo()
}

$cn.Children | where {$_.Class -eq "user"} | ft name, FullName,
description -auto

```

Полезные скрипты для SharePoint на PowerShell

Для тех, кто хочет почувствовать себя настоящим админом, в SharePoint имеется интеграция с PowerPoint.

Для изучения команд SharePoint на первом этапе можно пользоваться средой PowerShell ISE, это очень полезная оболочка для начинающих.

Эта среда позволяет в наглядном виде получать описания команд, а также при необходимости сохранять целые сценарии.

Запускаем PowerShell ISE и добавляем командлет SharePoint в консоли:

```
Add-PSSnapin Microsoft.Sharepoint.Powershell
```

Можно также встретить в скриптах более продвинутый вариант:

```

If ((Get-PSSnapIn -Name Microsoft.SharePoint.PowerShell -
ErrorAction SilentlyContinue) -eq $null )
{ Add-PSSnapIn -Name Microsoft.SharePoint.PowerShell }

```

Веб приложения и список сайтов

```

$webApp = Get-SPWebapplication 'http://london'
foreach($spSite in $webApp.Sites)
{

```

```
if ($spSite.HostHeaderIsSiteName)
{ Write-Host $spSite.Url 'is host-named' }
else
{ Write-Host $spSite.Url 'is path based' }
}
```

Результат приведен на рисунке 9.



```
http://london/apps/contoso is path based
http://dev.contoso.com is host-named
http://portal.contoso.com is host-named
http://team.contoso.com is host-named
```

Рисунок 9 – Результат скрипта

Коллекции сайтов могут быть двух типов, сайт на основе пути (*http://london/sites/demosite* по пути веб приложения создаются под сайты), и сайт на основе имени узла (*http://portal.contoso.com* любое красивое имя для сайта).

Подробнее на сайте *Microsoft*.¹³

Сайты на основе имени узла создаются только через Powershell, пример создания:

```
New-SPSite 'http://portal.contoso.com' -
HostHeaderWebApplication (Get-SPWebApplication 'http://london') -
Name 'Portal' -Description 'Portal site' -OwnerAlias
'contoso\administrator' -language 1033 -Template 'STS#0'
```

```
New-SPSite 'http://bi.contoso.com' -HostHeaderWebApplication
(Get-SPWebApplication 'http://london') -Name 'BI' -Description 'BI
site' -OwnerAlias 'contoso\administrator' -language 1033 -Template
'BICenterSite#0'
```

Доступные шаблоны можно посмотреть командой:

```
Get-SPWebTemplate
```

¹³ Архитектура и развертывание семейства сайтов с именем на основе узла в SharePoint Server <https://docs.microsoft.com/ru-ru/SharePoint/administration/host-named-site-collection-architecture-and-deployment?redirectedfrom=MSDN>

Список установленных компонентов

```
Get-SPFeature -Limit ALL | Where-Object {$_.Scope -eq "SITE"}
```

#Поиск по названию

```
Get-SPFeature -Limit ALL | Where-Object {$_.DisplayName -  
like "*Access*"}
```

Список доступных шаблонов сайтов

```
Get-SPWebTemplate | where {$_.Title -match "business"}
```

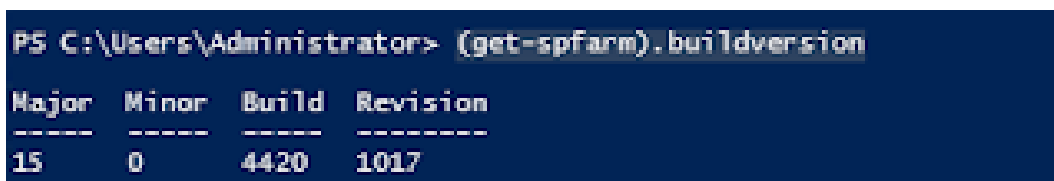
Список запущенных служб

```
Get-Service | Where-Object {$_.DisplayName -like "Sharep*"}
```

Текущая версия фермы

```
(get-spfarm).buildversion
```

Результат на рисунке 10.



```
PS C:\Users\Administrator> (get-spfarm).buildversion  
Major Minor Build Revision  
----
```

Major	Minor	Build	Revision
15	0	4420	1017

Рисунок 10 – Пример

Конвертация сайтов на основе пути сайта с именем

```
Backup-SPSite http://server_name/sites/site_name -  
Path C:\Backup\site_name.bak  
Remove-SPSite -Identity http://server_name/sites/site_name -  
Confirm:$False  
Restore-SPSite http://www.example.com -  
Path C:\Backup\site_name.bak -  
HostHeaderWebApplication http://server_name
```

Создание нового сайта

```
New-SPWeb -Url http://portal.contoso.com/TestSite -  
AddToQuickLaunch -AddToTopNav -Name TestSite -  
Template STS#0 -UseParentTopNav
```

Создание списка на сайте

```
$w= Get-SPWeb -Identity http://portal.contoso.com/TestSite  
$w.Lists.Add("Project Tasks", "" , "Tasks")
```

Экспорт сайтов, библиотек, списков

```
Export-SPWeb -identity "http://dev.contoso.com" –  
path "C:\Export_dev_contoso.cmp" -includeUserSecurity –Force
```

```
Export-SPWeb -identity "http://dev.contoso.com" -  
ItemUrl "/Contracts" -path "C:\Export_Contracts.cmp" -  
includeUserSecurity –Force
```

Импорт сайтов, библиотек, списков из стр файла (Content migration package)

```
Import-SPWeb -identity "http://dev.contoso.com" -  
path "c:\Export_contracts.cmp"
```

Выполнение Backup фермы, сайта

```
Backup-SPFarm -BackupMethod Full -  
Directory E:\FarmBackup\2016_09_01 –Force
```

```
Backup-SPSite http://london/sites/demosite -  
Path C:\Backup\site_name.bak
```

Восстановление удаленного сайта

```
Get-SPDeletedSite | select Path , siteid  
Restore-SpDeletedSite –Identity 4bff0c0e-4d56-4fec-8174-  
a60a16daeaac)
```

Включение расширенного логирования

```
Set-SPLogLevel -TraceSeverity VerboseEx -EventSeverity Verbose  
Merge-SPLogFile -Path "C:\FarmMergedLog.log" -Overwrite -  
StartTime (Get-Date).AddMinutes(-15)
```

```
#детальное описание ошибки по correlation ID  
Merge-SPlogfile -Path c:\log.log -Correlation ba05e237-0680-  
403a-b9f6-e49f96ac55d4
```

Отключение расширенного логирования

```
Clear-SPLogLevel
```

Список баз контента

```
Get-SPDatabase | ?{$_ .type -like "*business*" }
```

Подготовка базы для обновления

```
(Get-SPDatabase | ?{$_ .type -like "*business*" }).Provision()
```

Проверка базы данных контента на ошибки

```
Test-SPContentDatabase -name WSS_Content –  
webapplication http://london
```

Обновление конфигурации, исправление ошибок обновления

1. Запускать в командной строке под администратором.
2. Меняем путь на
C:\Program Files\Common Files\microsoft shared\Web Server
Extensions\15\BIN.
3. Для исправления ошибок патчей

PSConfig.exe -cmd upgrade -inplace b2b -force -cmd applicationcontent -install -cmd installfeatures.

Создание списка и элемента списка

```
$w = Get-SPWeb "http://portal.contoso.com/Testsite"  
$w.ListTemplates | Select name , type , type_client, Description  
$l = $w.Lists.Add("Project A Tasks" , "" , "Tasks")  
$l = $w.Lists.Add("Test Survey" , "" , "Survey")  
$list = $w.Lists["Project A Tasks"]  
$newItem = $list.AddItem()  
$newItem["Title"] = "Task 4"  
$newItem["Due Date"] = "6/20/2015 11:27 AM"  
$newItem["% Complete"] = 500  
$newItem.Update()
```

Подробно как изменять столбцы любого типа на сайте Microsoft.¹⁴

#список столбцов

```
$l.Fields | select Title , Type , StaticName
```

Создание библиотеки и загрузка файла

```
$DocLibName = "Files 2016"  
$lib = $w.Lists.Add($DocLibName, "", $w.ListTemplates["Docume  
nt Library"])
```

```
$file = Get-ChildItem "C:\Shared\day2.docx"  
$folder = $w.GetFolder($DocLibName )  
$files = $folder.Files
```

```
$timestamp = Get-Date -UFormat "%Y%m%d_%H%M%S"  
$fileName = [io.path]::GetFileNameWithoutExtension($file)  
$ext = [io.path]::GetExtension($file)  
#имя файла должно быть уникальным
```

¹⁴ SharePoint: A Complete Guide to Getting and Setting Fields using PowerShell
<https://social.technet.microsoft.com/wiki/contents/articles/20831.sharepoint-a-complete-guide-to-getting-and-setting-fields-using-powershell.aspx>

```
$f=$files.Add($DocLibName+ "/" + $fileName+ "_" + $timestamp+
$ext,$file.OpenRead(),$false)
#обновить значение поля
$f.Properties["TestColumn"] = "ABC"
$f.Update()
```

```
Update()
#Обновляет элемент в базе данных
#Обновляет поля "Modified" и "Modified by" .
#Создает новую версию
```

```
Systemupdate()
#Обновляет элемент в базе данных
#НЕ обновляет поля "Modified" и "Modified by" .
#НЕ создает новую версию
#Запускает системные обработчики
```

```
UpdateOverwriteVersion()
#Обновляет элемент но не создает новую версию.
#Обновляет поля "Modified" и "Modified by" .
```

Добавление столбца в список и создание представления

```
$w = Get-SPWeb "http://portal.contoso.com/Testsite"
$DocLibName = "Files 2016"
$lib = $w.Lists[$DocLibName]
$spFieldType = [Microsoft.SharePoint.SPFieldType]::Text
$lib.Fields.Add("TestColumn",$spFieldType,$false)
$lib.Update()
```

```
$v = $lib.Views["All Documents"]
$spField = $lib.Fields["TestColumn2"]
$v.ViewFields.Add($spField)
$v.Update()
```

Создание группы на сайте

```
$groupName = "Files 2016 Visitors"
#создаем группу на сайте
```

```
$group = $w.SiteGroups.Add($groupName, $w.EnsureUser("contoso\administrator"), $w.EnsureUser("contoso\administrator"), "Read Only Group")
```

Выдача уникального права на сайт, библиотеку или список

```
#прекратить наследование прав
$w.BreakRoleInheritance($true)
$lib.BreakRoleInheritance($true)
$list.BreakRoleInheritance($true)

#создаем назначение ролей, связываем группу и роль
$roleAssignment = new-
object Microsoft.SharePoint.SPRoleAssignment($group)
# $w.RoleDefinitions | select name, description,basepermissions
$roleDefinition = $w.RoleDefinitions["Read"]
#добавляем роль к назначению
$roleAssignment.RoleDefinitionBindings.Add($roleDefinition)
#Выдаем права например на сайт или на библиотеку
$w.RoleAssignments.Add($roleAssignment)
$lib.RoleAssignments.Add($roleAssignment)
$list.RoleAssignments.Add($roleAssignment)

$lib.Update()
$list.Update()
```

Пример готовой функции.¹⁵

Добавление пользователя в группу

```
$user=$w.EnsureUser("contoso\chen")
#Get group object
$group=$w.SiteGroups[$groupName]
#add user to group
$group.AddUser($user)
```

¹⁵ Add group to Library permissions using powershell
<https://sharepoint.stackexchange.com/questions/58293/add-group-to-library-permissions-using-powershell>

```
#Получить список пользователей из AD
Import-Module activedirectory
Get-ADUser -Filter {Enabled -eq "True"} | Select-
Object SamAccountName,Name,Surname,GivenName | Format-Table
```

Обновление поля списка

```
$w = Get-SPWeb https://sp
$l = $w.Lists["План по оцифровке"]
foreach($item in $l.Items) {
$item["Дата выполнения"] = $item["Дата выдачи"]
$item.Update() }
```

Получение элементов списка

```
$w = Get-SPWeb "http://portal.contoso.com/Testsite"
$l = $w.Lists["Project A Tasks"]
foreach($item in $l.Items) {"Title: {0} Due date: {1} , Created: {2}
" -f $item["Title"] , $item["Due Date"] , $item["Created By"] }
```

```
$l.Items | select Title,
@{Label="DueDate";Expression={$_[“DueDate”]}} ,
@{Label="PercentComplete";Expression={$_[“PercentComplete”]}} |
Export-Csv -path $path
```

```
#Экспортировать элементы списка в CSV
$path = "C:\Shared\Tasks_export.csv"
$list = $w.Lists["Project A Tasks"]
$list.Items | select Title , "Due Date" | Export-Csv -path $path
```

Получение элементов списка, используя CAML запрос

```
If ((Get-PSSnapIn -Name Microsoft.SharePoint.PowerShell -
ErrorAction SilentlyContinue) -eq $null )
{ Add-PSSnapIn -Name Microsoft.SharePoint.PowerShell }
```

```
$w = Get-SPWeb "http://portal.contoso.com/Testsite"
$l = $w.Lists["Project A Tasks"]
```

```

$query = New-Object Microsoft.SharePoint.SPQuery
$query.Query = "<Where><Gt><FieldRef
Name=""PercentComplete""></FieldRef><Value
Type=""number"">0.05</Value></Gt></Where>"
$query.ViewFields = "<FieldRef Name=""Title"" /><FieldRef
Name=""PercentComplete"" /><FieldRef Name=""Modified""
/><FieldRef Name=""DueDate"" /><FieldRef Name=""Editor""
/><FieldRef Name=""Priority"" />"
$query.ViewAttributes = "Scope=""Recursive""
$query.RowLimit = 0
$listItems = $l.GetItems( $query)
foreach($item in $listItems) {"Title: {0} Due date: {1} , Priority:
{2} ,PercentComplete:{3} , Editor: {4}" -
f $item["Title"] , $item["Due
Date"] , $item["Priority"] , $item["PercentComplete"] , $item["Edito
r"] }

```

Для подготовки CAML запроса можно использовать утилиту SP CAML Query Helper 2013

Загрузка списка из csv файла

```

$w = Get-SPWeb "http://portal.contoso.com/Testsite"
$list = $w.Lists["Project A Tasks"]
$tblData = Import-CSV "C:\Shared\Tasks.csv"
foreach ($row in $tblData)
{
    "Adding entry for "+$row."Title".ToString()
    $NewItem = $list.AddItem()
    $NewItem["Title"] = $row."Title".ToString()
    $NewItem["Due Date"] = $row."Due Date".ToString()
    $NewItem["% Complete"] = $row."PercentComplete".ToString()
    $NewItem.Update()
}

#Получить таблицу из базы
$SqlConnection = New-Object
System.Data.SqlClient.SqlConnection

```

```

$SqlConnection.ConnectionString =
"Server=$server;Database=$database;Integrated Security=True"
$SqlCommand = New-Object System.Data.SqlClient.SqlCommand
$SqlCommand.CommandText = " sp_helpdb"
$SqlCommand.Connection =
$SqlConnection$SqlCommand.CommandTimeout = 0
$SqlAdapter = New-Object System.Data.SqlClient.SqlDataAdapter
$SqlAdapter.SelectCommand = $SqlCommand
$DataSet = New-Object System.Data.DataSet
$SqlAdapter.Fill($DataSet)
$SqlConnection.Close()
$tblData = $DataSet.Tables[0]

```

Как сохранить элемент в таблицу.¹⁶

Удаление элементов списка

```

$w = Get-SPWeb "http://portal.contoso.com/Testsite"
$l = $w.Lists["Project A Tasks"]
$query = New-Object Microsoft.SharePoint.SPQuery
$query.Query = "<Where><IsNull><FieldRef
Name=""Title""></FieldRef></IsNull></Where>"
$query.ViewFields = "<FieldRef Name=""ID"" /><FieldRef
Name=""GUID"" /><FieldRef Name=""Title"" /><FieldRef
Name=""PercentComplete"" /><FieldRef Name=""Modified""
/><FieldRef Name=""DueDate"" /><FieldRef Name=""Editor""
/><FieldRef Name=""Priority"" />"
$query.ViewAttributes = "Scope=""Recursive""
$query.RowLimit = 0
$listItems = $l.GetItems( $query)

for ($x=$listItems.Count-1;$x -ge 0; $x--)
{
    Write-Host("DELETED: " + $listItems[$x].Item("ID"))
    $listItems[$x].Delete()
}

```

¹⁶ PowerShell Scripts for Export and Import SharePoint List to SQL table
<http://www.dotnetfunda.com/codes/show/6988/powershell-scripts-for-export-and-import-sharepoint-list-to-sql-table>

}

Реактивация RBS

```
$cdb = Get-SPContentDatabase –WebApplication  
http://sharepoint\  
$rbss = $cdb[0].RemoteBlobStorageSettings  
$rbss.Enable()
```

Управление оповещениями

Подробно указано на сайте SharePoint Diary¹⁷
Настройка собственных шаблонов оповещений указано на
сайте Zeb Sadiq¹⁸

```
$SPweb = Get-SPWeb https://sharepoint.com
```

```
foreach($alert in $SPweb.Alerts) {if ($alert.Title -eq 'Документы  
оповещения') {write-host $alert.Title}}
```

```
$SPweb.Alerts | Export-Csv C:\temp2>alerts.csv -encoding utf8
```

```
$SPList = $SPweb.lists["Документы"]  
$IDS = ""  
foreach($alert in $spweb.alerts)  
{  
    if($alert.ListID -eq $SPList.ID -and $alert.Title -eq  
'Документы')  
    {  
        $IDS += $alert.ID.toString() + "|"  
    }  
}
```

¹⁷ Create - Edit - Find - Delete SharePoint Alerts using PowerShell
<https://www.sharepointdiary.com/2011/11/managing-alerts-using-powershell.html>
#Read more: <https://www.sharepointdiary.com/2011/11/managing-alerts-using-powershell.html#ixzz61rFAxCHZ>

¹⁸ SharePoint 2010 custom alert template
<http://blog.zeksadiq.com/post/SharePoint-2010-custom-alert-template.aspx>

```

    write-host -nonewline "*"
  }
  write-host "deleting..."
  foreach($s in $IDS.Split("|"))
  {
    write-host -nonewline "*"
    write-host [GUID]$s
    # $spweb.alerts.delete([GUID]$s)
  }

```

Поиск списков с анонимным доступом

```

$site = Get-SPSite https://sp
foreach ($web in $site.AllWebs) {foreach($list in $web.lists) {if
($web.AnonymousState.toString() -eq "Enabled" -and
$list.AnonymousPermMask.toString() -ne "OpenWeb" -and
$list.AnonymousPermMask.toString() -ne "EmptyMask" -and
$list.Title -ne "TaxonomyHiddenList" ) {write-host $web.Url
,$list.Title , $list.AnonymousPermMask.toString()
,$web.AnonymousState.toString(),$web.AllowAnonymousAccess.tostr
ing() ,$web.AnonymousPermMask64.toString() }}}

```

Проверка SSID пользователя

```

$domain = "domain";
$user_name = "user1";
$application = "https://sharepoint.com";

$userID = $domain + "\" + $user_name;
$SPuser = Get-SPUser -Identity $userID -Web $application
$objUser = New-Object
System.Security.Principal.NTAccount($domain, $user_name)
$strSID =
$objUser.Translate([System.Security.Principal.SecurityIdentifier])

Write-Host '-----'

$SPuser.sid
$strSID.Value

```

Создание группы на основе группы AD

```
$SiteCollection = "http://site/sitecollection"
$ADGroupName = "domain\adgroup"
$SPGroupName = "My Test Group A1"
$SPGroupDescription = "Test Group A1's Description"
$SPGroupPermission = "Read"

#Start of script
$site = Get-SPWeb $SiteCollection

#Check if the group already exists
if ($site.SiteGroups[$SPGroupName] -eq $null)
{

    #Ensure Group/User is part of site collection users beforehand and
add them if needed
    $site.EnsureUser($ADGroupName)

    # Get the AD Group/User in a format that PowerShell can use
otherwise there will be a string error
    $ADGroupSPFriendly = $site | Get-SPUser $ADGroupName

    #Create the SharePoint Group – Group Name, Group Owner, Group
Member, Group Description. Can't add AD group yet...
    $NewSPGroup = $site.SiteGroups.Add($SPGroupName,
$site.CurrentUser, $site.CurrentUser, $SPGroupDescription)
    $site.AssociatedGroups.Add($site.SiteGroups[$SPGroupName]);
    $NewSPAccount = $site.SiteGroups[$SPGroupName]

    #Assign the Group permission
    $GroupAssignment = New-Object
Microsoft.SharePoint.SPRoleAssignment($NewSPAccount)
    $GroupRole = $site.RoleDefinitions[$SPGroupPermission]
    $GroupAssignment.RoleDefinitionBindings.Add($GroupRole)
    $site.RoleAssignments.Add($GroupAssignment)
```

#Add the AD Group/User to the group, can't be done during group creation when using Powershell otherwise errors so is done now.

```
Set-SPUser -Identity $ADGroupSPFriendly -Web $SiteCollection -  
Group $SPGroupName  
}  
$site.Dispose()  
}
```

Вопросы для самопроверки по содержимому раздела «Microsoft PowerShell»

1. Что такое PowerShell?
2. Отличия в существительных между командами SharePoint Online PowerShell и командами Office 365 PowerShell?
3. Расширение имени файла сценария PowerShell?
4. Команда для возвращения текущего соединения в MS SharePoint Online?

ПРИЛОЖЕНИЕ 1

СКРИПТ «ХРАНЕНИЕ ПАРОЛЕЙ В РЕЕСТРЕ ДЛЯ POWERSHELL. СОЗДАНИЕ СПИСКОВ»

```
# https://www.spjeff.com/2016/08/17/save-encrypted-passwords-to-registry-for-powershell/
# https://blogs.technet.microsoft.com/undocumentedfeatures/2015/10/01/storing-powershell-credentials-in-the-local-user-registry/
# https://social.technet.microsoft.com/wiki/contents/articles/4546.working-with-passwords-secure-strings-and-credentials-in-windows-powershell.aspx
#####
### Предварительные операции (не входят в скрипт автоматизации, выполняются по необходимости)
## Запрос политики выполнения скриптов в среде PowerShell
# Get-ExecutionPolicy
## Установка политики выполнения скриптов в среде PowerShell
# Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope CurrentUser
## Сохранение пользовательского идентификатора и пароля в реестре
$UserName = "userid"
$PassWord = "password" | ConvertTo-SecureString -AsPlainText -Force | ConvertFrom-SecureString
If (Test-Path "HKCU:\Software\SClibrary\Credentials")
{
# Если записи в реестре уже есть
    Set-ItemProperty -Path HKCU:\Software\SClibrary\Credentials -Name UserName -Value $UserName -Force
    Set-ItemProperty -Path HKCU:\Software\SClibrary\Credentials -Name Password -Value $PassWord -Force
} else {
# Если указанного пути нет, создаем
    New-Item -Path "HKCU:\Software\SClibrary" -Name "Credentials" -Force
# Присваиваем свойствам UserName и Password соответствующие значения
    New-ItemProperty -Path HKCU:\Software\SClibrary\Credentials -PropertyType String -Name UserName -Value $UserName -Force
    New-ItemProperty -Path HKCU:\Software\SClibrary\Credentials -PropertyType String -Name Password -Value $PassWord -Force
}
## Получаем внутренние имена полей списка
$internalFieldName = ("Дата поступления" | Format-Hex -Encoding BigEndianUnicode).ToString() -split(" ")
$internalFieldName =
"_x" + $internalFieldName[3] + $internalFieldName[4] + "_" + `
"_x" + $internalFieldName[5] + $internalFieldName[6] + "_" + `
```

```

"_x" + $internalFieldName[7] + $internalFieldName[8] + "_" + `
"_x" + $internalFieldName[9] + $internalFieldName[10] + "_" + `
"_x" + $internalFieldName[11]
$internalFieldName
$internalFieldName = ("Выпуск" | Format-Hex -Encoding
BigEndianUnicode).ToString() -split(" ")
$internalFieldName = `
"_x" + $internalFieldName[3] + $internalFieldName[4] + "_" + `
"_x" + $internalFieldName[5] + $internalFieldName[6] + "_" + `
"_x" + $internalFieldName[7] + $internalFieldName[8] + "_" + `
"_x" + $internalFieldName[9] + $internalFieldName[10] + "_" + `
"_x" + $internalFieldName[11]
$internalFieldName

```

```
#####
```

```

### Код для скрипта автоматизации
### Сохранение и/или изменение пароля выполняется отдельным скриптом
по необходимости
# Полный путь к файлу MS Word с текстом объявления (пример)
$MSGFile = "C:\Users\library\Новые поступления.docx"
# Проверка существует ли файл и установлен ли MS Word или нет
if(test-path HKLM:SOFTWARE\Classes\Word.Application) { if(Test-Path
$MSGFile)
{
# Файл существует
$SiteURL = "www.library.urfu.ru" # (пример)
$uri = "http://" + $SiteURL + "/rus/Library" + "/_vti_bin/Lists.asmx?wsdl"
if(Test-Connection -computer $SiteURL -quiet)
# Сайт доступен
# Имя списка, из которого будут отбираться данные:
>ListName = "Поступление журналов"
# Диапазон дат для поискового запроса:
$PrewDate = Get-Date ((Get-ChildItem $MSGFile).LastWriteTime) -
format s
$CurrentDate = Get-Date -format s
# Собственно запрос
$xmlDoc = new-object System.Xml.XmlDocument
$query = $xmlDoc.CreateElement("Query")
# CAML разный если дата предыдущего формирования объявления
сегодняшняя или в прошлом
if((Get-Date $CurrentDate -format d) -eq (Get-Date $PrewDate -format
d)) {$QueryCAML = '<Where><Eq><FieldRef
Name="_x0414_x0430_x0442_x0430_x00" /><Value

```

```

IncludeTimeValue="FALSE" Type="DateTime">' + $CurrentDate +
'</Value></Eq></Where><OrderBy><FieldRef
Name="_x0414_x0430_x0442_x0430_x00"/></OrderBy>'}
    else {$QueryCAML = '<Where><And><Leq><FieldRef
Name="_x0414_x0430_x0442_x0430_x00" /><Value
IncludeTimeValue="FALSE" Type="DateTime">' + $CurrentDate +
'</Value></Leq><Gt><FieldRef
Name="_x0414_x0430_x0442_x0430_x00" /><Value
IncludeTimeValue="FALSE" Type="DateTime">' + $PrewDate +
'</Value></Gt></And></Where><OrderBy><FieldRef
Name="_x0414_x0430_x0442_x0430_x00"/></OrderBy>'}
    $query.set_InnerXml("$QueryCAML")
# Возвращаемые в ответе на запрос столбцы (поля) списка, по умолчанию -
все
    $viewFields = $xmlDoc.CreateElement("ViewFields")
# Некоторые другие параметры оставляем в значении по умолчанию
    $queryOptions = $xmlDoc.CreateElement("QueryOptions")
# В ответе на запрос будет возвращено не более 1000 записей
    $rowLimit = "1000"
# Читаем пользовательский идентификатор и пароль из реестра
    If (Test-Path "HKCU:\Software\SClibrary\Credentials")
    {
# Если есть путь в реестре для хранения пароля
        $UserName = (Get-ItemProperty -Path
HKCU:\Software\SClibrary\Credentials).UserName
        $PassWord = (Get-ItemProperty -Path
HKCU:\Software\SClibrary\Credentials).Password
# Проверяем есть ли значения переменных
        If ($UserName -ne $NULL -and $PassWord -ne $NULL)
        {
            $PassWord = ConvertTo-SecureString $PassWord
# Почему так, а не сразу при чтении в конвейере?
# Если сразу, а значение пусто, будет ошибка и скрипт может упасть
            $credential = New-Object -TypeName
System.Management.Automation.PSCredential -ArgumentList $UserName,
$PassWord
# Тут помещать будем остальной код
            $service = New-WebServiceProxy -Uri $uri -Namespace SpWs -
Credential $credential -ErrorAction Stop
# Получаем данные из списка
            $List = $service.GetListItems($listName, "", $query, $viewFields,
$rowLimit, $queryOptions, "")
# Проверяем сколько записей из списка сервер вернул в ответ на запрос
            $Test = $($List.data.row | Measure-Object).Count

```

```

        if($Test -gt 0)
        {
# Есть результаты выборки. Открываем MS Word
            $word = new-object -ComObject "Word.Application"
# Устанавливаем параметр видимости в значение "истина" для того что бы
наблюдать работу скрипта
            $word.Visible = $TRUE
#-----#
# Перемещаем окно MS Word на передний план для отладки и наглядности
$H = (Get-Process winword).MainWindowHandle
Add-Type @"
    using System;
    using System.Runtime.InteropServices;
    public class Tricks {
        [DllImport("user32.dll")]
        [return: MarshalAs(UnmanagedType.Bool)]
        public static extern bool SetForegroundWindow(IntPtr hWnd);
    }
"@
[void] [Tricks]::SetForegroundWindow($H)
#-----#
# Открываем уже имеющийся документ - шаблон
            $docu = $word.Documents.Open("$MSGFile")
            $LETable = $docu.Tables.Item(1)
            $LETableCols = $LETable.Columns.Count
            $LETableRows = $LETable.Rows.Count
            if($Test -gt $LETableRows)
            {
# Если найдено строк больше, чем строк в таблице, надо добавить строки
                For ($i=1; $i -le $($Test - $LETableRows); $i++) {$SRC =
$LETable.Rows.Add()}
# Проходим по всем элементам результата обращения к сервису и
формируем таблицу со списком новых поступлений объявления.
                $r = 1 # Счетчик строк
                foreach($Item in $List.data.row)
                {
                    if(1 -eq $r)
                    {
# Заменяем заголовок
                        $Header = "Дата поступления" + " " + (Get-Date
$($Item.ows__x0414__x0430__x0442__x0430__x00) -format D) + "`n"
                        $docu.Paragraphs.item(1).Range.Font.Size = 14
                        $docu.Paragraphs.item(1).Range.Font.Name =
"Times New Roman"

```

```

        $docu.Paragraphs.item(1).Range.Text = $Header
        $docu.Paragraphs.item(1).Alignment = 1
    }
    $LETTable.Cell($r,1).Range.Text =
$(($Item.ows_Journal.split('#'))[1]
        $LETTable.Cell($r,1).Range.Font.Size = 14
        $LETTable.Cell($r,1).Range.Font.Name = "Times New
Roman"

        $LETTable.Cell($r,2).Range.Text =
$Item.ows__x041f__x043e__x0441__x0442__x04
        $LETTable.Cell($r,2).Range.Font.Size = 14
        $LETTable.Cell($r,2).Range.Font.Name = "Times New
Roman"

        $r = $r + 1
    } # Завершился проход по всем найденным записям
# Удаляем лишние строки
    For ($i=$LETTableRows; $i -Ge $r; $i--)
    { $LETTable.Rows.Item($i).Delete()
      $LETTable.UpdateAutoFormat()
#          $docu.Save() # сохраняем документ
#          $docu.Close() # закрываем документ
#          $word.Quit() # завершаем работу MS Word

# распечатываем документ (можно указать на каком конкретно принтере)
#          Start-Process -FilePath "$MSGFile" -Verb print
# Известна ошибка, из-за которой работа MS Word запущенного из
PowerShell может не быть корректно завершена. Для окончательного
завершения MS Word выпоним следующие действия
#          $rc =
[System.Runtime.InteropServices.Marshal]::ReleaseComObject($word)
    } else {"Записей, удовлетворяющих заданным условиям не
найдено"}
    } else {"Пользовательский идентификатор или пароль не
задан"}
    } else {"В реестре нет параметров для пользовательского
идентификатора и пароля"}
    } else {"Нет доступа сайту"}
    } else {"Нет файла - шаблона"}
    } else {"Не установлен MS Word"}

```

ПРИЛОЖЕНИЕ 2

ОТВЕТЫ НА ВОПРОСЫ САМОПРОВЕРКИ

К разделу «Краткие сведения о системе MS SharePoint»:

1. Функциональный состав платформы MS SharePoint 2010.
2. MS SharePoint 2010 тесно интегрирован с пакетом программ MS Office 2010.
3. Примером интеграции сервера MS SharePoint 2013 с другими серверами в системных комплексах является информационная система управления проектами с такими серверами как MS Project, MS SQL.
4. MS SharePoint 2019 интегрируется с такими облачными компонентами как Power Apps, Power BI, MS Flow.
5. Назначение нового командлета Get-SPContentDatabaseOrphanedData в MS SharePoint 2019 в MS SharePoint 2019 – это находить потерянные объекты в базе данных контента.
6. Основные бизнес-задачи, решаемые в MS SharePoint 2019: увеличение производительности труда, организация совместной работы сотрудников, контроль информационных процессов, принятие взвешенных решений, управление рабочими процессами.

К разделу «Microsoft PowerShell»:

1. PowerShell – это оболочка командной строки с поддержкой задач и язык скриптов на основе платформы .NET.
2. Главное различие – это существительные (the nouns), которые они используют. Каждый командлет SharePoint Online будет начинаться с SPO в качестве существительного. И наоборот, Office 365 будет начинаться с MSO в качестве существительного.
3. Расширение имени файла сценария MS PowerShell – это ps1.
4. Командлет MS SharePoint Online для возвращения текущего соединения – это Get-PnPConnection.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ноэл М., Спенс К. Microsoft SharePoint 2010. Полное руководство. – М. : Вильямс, 2012.
2. Сайт EazyBI. – URL: <http://www.eazybi.blogspot.com> (дата обращения: 2.09.2019).
3. Сравнение возможностей локальных версий SharePoint. – URL: <https://products.office.com/compare-all-microsoft-office-products?tab=2> (дата обращения: 28.06.2018).
4. Стаханов Е. Краткий обзор возможностей MS SharePoint (компания Terrelink). – URL: http://officemagazine.ru/по_category/101854 (дата обращения: 18.06.2018).
5. Архитектура и развертывание семейства сайтов с именем на основе узла в SharePoint Server. – URL: <https://docs.microsoft.com/ru-ru/SharePoint/administration/host-named-site-collection-architecture-and-deployment?redirectedfrom=MSDN> (дата обращения: 10.03.2019).
6. Add group to Library permissions using powershell. – URL: <https://sharepoint.stackexchange.com/questions/58293/add-group-to-library-permissions-using-powershell> (дата обращения: 24.03.2019).
7. Create – Edit – Find – Delete SharePoint Alerts using PowerShell. – URL: <https://www.sharepointdiary.com/2011/11/managing-alerts-using-powershell.html> (дата обращения: 16.04.2019).
8. Connect to SharePoint Online. – URL: <https://www.enjoysharepoint.com/connect-to-sharepoint-online-site-using-powershell/> (дата обращения: 9.07.2019).
9. Microsoft SharePoint Server 2013. – URL: <http://www.interface.ru/home.asp?artId=33974> (дата обращения: 8.09.2018).
10. Microsoft SharePoint Server 2016 Architectural Models. – [SharePoint2016ArchitecturalModels.pdf](#).
11. SharePoint: Complete Guide to Getting and Setting Fields using PowerShell. – URL: <https://social.technet.microsoft.com/wiki/contents/articles/20831.sharepoint-a-complete-guide-to-getting-and-setting-fields-using-powershell.aspx>. (дата обращения: 11.02.2019).

12. SharePoint 2010 custom alert template. – URL: <http://blog.zeksadiq.com/post/SharePoint-2010-custom-alert-template.aspx> (дата обращения: 19.12.2018).

13. SharePoint Server 2019. – URL: www.mssoft.ru (дата обращения: 9.03.2019).

14. SharePoint Server Cmdlets 2013, 2016, 2019. – URL: <https://docs.microsoft.com/en-us/powershell/sharepoint-ps/sharepoint/sharepoint-online-cmdlets> (дата обращения: 9.07.2019).

15. PowerShell Scripts for Export and Import SharePoint List to SQL table. – URL: <http://www.dotnetfunda.com/codes/show/6988/powershell-scripts-for-export-and-import-sharepoint-list-to-sql-table> (дата обращения: 13.04.2019).

Электронный текстовый ресурс

Ермаков Дмитрий Германович

Тебайкина Надежда Ивановна

ИСПОЛЬЗОВАНИЕ ВЕБ-СЛУЖБ MS SHAREPOINT

Методические указания к выполнению самостоятельных работ

Подготовлено к публикации
Компьютерный набор

Н.И. Тебайкина
Н.И. Тебайкина

Рекомендовано Методическим советом УрФУ

Разрешено к публикации _____

Электронный формат – pdf

Объем 3,82 уч.-изд. л.

620002, Екатеринбург, ул. Мира, 19
Информационный портал УрФУ
<http://www.study.urfu.ru>