Документация от разработчика по intranet. ( sharepoint server 2013 )  
  
Оглавление для разработчика :  
 a) Общее:  
 0. Введение  
 1. Общее построение страниц в sharepoint.

1.1 Понятия: извлечение, публикация в sharepoint.  
 2. Описание содержимого masterpage.  
 2.1 Добавление js/css/wsp/ файлов и html содержимого.  
 3. Создание и описание внешней страницы, как добавлять веб части и произвольный html   
 4. Wsp файлы - описание, как внедрять в sharepoint, разработка  
 5. Технология CSR для списков

6. Роль поиска и протокола REST в портале intranet

6.1 Панель администратора sharepoint, навигация по ней, описание служб  
 6.2 Свойства пользователей и их мапинг и использование в поиске  
 7. Инструменты разработчика  
 7.1 IDE

7.2 специальная программа по поиску

7.3 специальная программа по логам

7.4 специальная программа по иерархии портала. Для списков часто используется

7.5 chrome dev tool, дебаг js, console.log, поиск нужного контента, смотреть rest запросы  
 7.7 sharepoint designer – когда и как используется.

7.8 Advanced REST client – специальная программа для REST запросов

8. powershell – его роль в портале intranet  
 9. Известные пути в sharepoint: layout, stile library.

10. Контроль версий – git.  
 11. Блоги, полезные ресурсы

12. Библиотеки для sharepoint, которые были использованы

13. Порядок загрузки скриптов в sp.  
 14. asp.net программирование.

15. правила доступа в sp, группы

16. тестовый пользователь.

17. sharepoint hosted app.

18. кеш, ctrl f5

19. принтеры, офисная техника, другой портал

20. совместимость броузеров, основной броузер ie

b) Детальное, постраничное описание портала intranet:

1.1. Как построены страницы на intranet.  
 2. Папки CustomCss, CustomjsLibs на server-sp для intranet  
 2.1 Контроль версий для этих папок - git.

3. Страницы - детальное описание, файлы js, css:

3.1 Главная   
 3.2 Сотрудники

3.3 Документы

3.4 Форум  
3.5 Техподдержка для пользователей  
3.6 Техподдержка для сотрудников IT  
3.7 Личный кабинет.

4. мастер страницы портала intranet (их расположение):

контакты

1. Введение
2. Общее построение страниц в sharepoint

Основа страницы – это файл с раcширением master. Это скелет страниц sharepoint. Название этого файла – masterpage. Классический портал состоит из коллекций сайтов, которые можно создавать из панели администратора. При создании предлагается выбор шаблонов – для документов, форума и тд. Для каждой коллекции сайтов создается свой masterpage. Просмотр, редактирование и обновление этого файла происходит через sharepoint designer. Чтобы увидеть file.master нужно зайти в sharepoint designer, открыть нужную коллекцию сайтов, далее “все файлы” => “\_catalogs” => “masterpage” => и выбрать нужный файл.  
  
С другой стороны когда коллекция сайтов создана и если на нее зайти через броузер, то обладая правами администратора, через правый верхний тогл, можно создавать новую страницу или изменить содержимое текущей страницы.   
В моем абстрактном понимании – страница sharepoint = masterpage + внешняя страница.  
При нажатии на изменение текущей страницы, попадаешь на режим “editmode page”,   
в котором ты волен добавлять любые веб части на странице.

1.1 При нажатии на изменение текущей страницы, автоматически страница попадет в состояние “извлеченной”. Понятие “извлеченный файл” – это удобный способ  
редактировать файлы в фоновом режиме, чтобы другие пользователи портала не видели сайд эффекты режима редактирования, а пользовались старой версией файла/страницы.  
И чтобы выкатить стабильную новую версию для пользователей нужно нажать опубликовать. Такое версионирование файлов используется так же во всех файлах, которые можно открыть через sharepoint designer: masterpage и в технологии CSR, которая будет описана позже.

2. Описание содержимого masterpage.  
 При первом осмотре содержимого masterpage становится не понятно, что за что отвечает и как этот файл редактировать. В некоторых случаях рядом с файлом с расширением “master” присутствует html файл, тогда sharepoint не даст напрямую редактировать masterpage, а  
 нужно редактировать html файл, который потом конвертируется в masterpage.  
  
(Если редактируешь прямо masterpage, то добавлять без <!----> )

Добавление js:   
 <!--SPM:<SharePoint:ScriptLink language="javascript" name="CustomjsLibs/jquery-2.1.4.min.js" Defer="false" runat="server" Localizable="false"/>-->

Добавление css:

<!--SPM:<SharePoint:CssRegistration Name="/\_layouts/15/CustomCss/css/bootstrap-scope.min.css" runat="server"/>-->

Добавление html содержимого:  
Нажимаем ctrl f и ищем id “s4-bodyContainer” далее, во внутренний div можно вставлять html текст

1. Создание и описание внешней страницы, как добавлять веб части и произвольный html

Чтобы создать страницу, нужно нажать тогл => добавление страницы. Так были созданы страницы: Документы(она потом делает редирект на коллекцию сайтов документы), Курсы валют,  
Техподдержка для клиентов.

Чтобы добавить html содержимое, нужно добавить веб часть «Редактор Сценариев» (client editor web part - cewp) и в него добавлять верстку.   
Так были сверстаны: главная страница, страница документов, страница техподдержки, модальное окно для страницы техподдержки ИТ.

1. Wsp файлы - описание, как внедрять в sharepoint

Через wsp файлы, внедряется серверный код на sharepoint.  
На intranet есть серверный код (wsp) для:  
  
1. Погода и Курсы валют на главной. path: C:\sharepoint.apps\VisualWebPartCurrencyWeather

2. Курсы валют на отдельной странице path: C:\sharepoint.apps\VisualWebPartCurrencyTable  
3. Определение ip и имя компьютера на странице техподдрежки path: C:\sharepoint.apps\VisualWebPartDefineIP

1. В личном кабинет, для задач, (нет в продакшене) path: C:\sharepoint.apps\UserControlWidgetTasks

Решения создаются в visual studio, wsp file лежит в папке Debug разрабатываемого решения,   
Создается wsp файл при компиляции проекта. Случается, что при развертывании visual web part – не активируется feature решения и как результат visual web part не появляется в панели. Нужно активировать через powershell feature этого wsp файла.   
  
Разработка ведется на сервере devsp с локальный тестовый порталом <http://devsp/>  
В visual studio 15 community в свойствах проекта wsp решение, можно проставлять sharepoint url, куда будет ставиться решение в режиме дебага при нажатии build/f5. Url путь не влияет на wsp файл, этот wsp файл можно ставить на intranet.

Пример пути папки Debug:  
C:\sharepoint.apps\UserControlWidgetTasks\WidgetTasksUserControl\bin\Debug  
  
Посмотреть список всех wsp файлов в панели администратора:  
Панель администрирования => Параметры системы => Управление решениями для фермы.  
Если кликнуть на решение, то можно его отозвать, а затем удалить. Добавляется wsp file через powershell.

1. Технология CSR для списков

Client Side Rendering – внедряется js код для изменения визуализации списков sharepoint.   
Андреем Маркеевым был разработан плагин к chrome для CSR, названный cisar, у него так же есть блог, к котором есть очень хорошие статьи посвященной этой теме. Этот плагин используется для intranet. p.s Встроенная ide в chrome dev tool.

В intranet, CSR используется в техподдержки для ИТ: добавление кнопки принять заявку, отправить заявку.  
Так же используется на другом sharepoint портале для складского учета.  
clienttemplate.js – это файл от ms, для CSR. Беглый просмотр этого файла поможет понять

примерное api CSR. Найти этот файл можно в папке layout или на devsp в папке C:\sharepoint.apps\Core Libs (там он скопирован из папки layout)  
Официального api по CSR нет.

1. Роль поиска и протокола REST в портале intranet

Поиск в sharepoint очень мощный и обладает хорошим api как на клиенте, так и через c#.

На клиенте, с помощью поиска с помощью протокола REST можно получать нужные данные. Все динамические данные связанные с пользователями на портале intranet извлекаются через javascript sharepoint search rest api. (календарь, страница сотрудников). P.s Так же протокол REST используется для crud operation в списках.   
[Знакомство со службой REST для SharePoint 2013](https://msdn.microsoft.com/ru-ru/library/office/fp142380.aspx)

[Работа со списками и элементами списков в службе REST](https://msdn.microsoft.com/ru-ru/library/office/dn292552.aspx)

Пример rest запроса ниже.

6.1 Панель администратора sharepoint, навигация по ней, описание служб

Служба поиска и служба учетных записей пользователей – это две самые частые службы с которыми придется иметь дело. Первая все индексирует для поиска и необходима разработчику, чтобы в схеме поиска создавать новые свойства, которые можно мапить с чем угодно, а вторая синхронизирует данные пользователей из ad в sharepoint. P.s использовалось для маппинга свой пользователей из AD.  
  
Конкретный пример мапинга: панель администратора sharepoint => управление приложениями => управление приложениями-службами=> приложение службы поиска=>схема поиcка=> в фильтре написать “Department” и можно увидеть это найденное свойство, которое сопоставлено с «Люди:Department, ows\_Department»

Эти управляемые свойства в схеме поиска используется в js коде, в REST search api.

пример:  
$.ajax({

url: "/\_api/search/query?querytext='\*'&trimduplicates=false&enablequeryrules=false&rowlimit=600&bypassresulttypes=true&selectproperties='Title%2cJobTitle%2cDepartment%2cBirthday%2 …  
  
управляемые свойства: Title,JobTitle,Departmen,Birthday

все их можно найти в схеме поиска

Не понятные по названию свойства:

RefinableString01 - матчится с postal code, главенство сотрудников в департаменте

RefinableString02 - матчится с LastName

7. Инструменты разработчика  
 7.1 IDE

Visual studio 2015, visual studio code на server-sp  
visual studio code очень удобен для js,ts, быстр, интегрирована консоль, подключаемые модули, alt shift f – форматирование, ,contrl shift b – компиляция ts,

7.2 специальная программа по поиску

Называется: searchquerytoolv2.5

Создана, чтобы через «консольное» приложение с интерфейсом делать rest запросы поиска и смотреть результат. Очень удобна, так как rest запросы поиска очень часто используются.

7.3 специальная программа по логам sp  
 Называется: ulsviewer

Sharepoint постоянно пишет в лог, и если что-то пошло не так, можно попытаться найти по логам причину.  
Так же если броузер, показал, что что-то пошло не так и показал id ошибки, то удобно с помощью скрипта powershell который принимает id ошибки, получить описание ошибки. Название скрипта:  
C:\powershell\ launchWsp.ps1

7.4 специальная программа по иерархии портала.

Называется: SharePoint 2013 Client Browser v1.9.

Использовалась, чтобы получать internal name столбцов списка, или

7.5 chrome dev tool, дебаг js, console.log, поиск нужного контента.

Открывается через нажатие f12.

вкладка Network:

Смотреть что и сколько времени загружается, текущие запросы (rest)

вкладка Sources:  
 Смотреть все, что подключено к текущей странице , js,css,html.

Тут можно делать дебаг js файлов: открыть любой js файл и поставить точку останова и перезагрузить страницу и по f10 построчно итерировать код.   
вкладка Console:  
 вывод js функции console.log, и если что-то идет не так, хорошая вероятность, что броузер тут оповестит о случившемся. Если по логу понятно, что где-то рядом со строчкой console.log в коде ошибка, то можно прямо нажать на console.log, тебя перекинет на файл, в котором написан console.log и поставить точку останова и шанс быстро понять, что пошло не так

Вкладка elements:

Быстрый способ работы с версткой: смена css, написание новых классов, просмотр ховер эффетов.  
Выбираем нужный элемент на сайте, у которого хотим поработать с верстой, нажимаем правую кнопку – посмотреть элемент => попадаем на этот элемент, колонка справа: все о css текущего элемента. Тут можно динамически менять свойства классов css,   
 кнопка :hov для просмотра hover эффектов,

кнопка .cls для написания нового правила к элементу, которое можно потом скопировать в css файл

так же удобно найти нужную часть верстки через ctrl f  
chrome dev tool – лучший друг web разработчика.

7.7 sharepoint designer – когда и как используется.

Используется для работы с masterpage и рабочих процессов (сейчас это отправка писем)

1. powershell – его роль в портале intranet

Используется для развертывания wsp файлов. Так же для вывода ошибки по id. Активация feature wsp файлов, если это требуется.

Файл развертывания: C:\powershell\launchWsp.ps1.  
Этот файл нужно переписать, так как он не автоматизирован.   
Он только добавляет wsp файл на ферму.  
его нужно два раза запустить, для деплоя и активации.  
Удаляю вручную wsp файлы через панель администрирования.

1. Известные пути в sharepoint: layout, stile library.

Layout – папка на сервере sharepoint, туда рекомендуется класть пользовательские js и css файлы.  
путь: C:\Program Files\Common Files\microsoft shared\Web Server Extensions\15\TEMPLATE\LAYOUTS

stile library – аналог layout, но работа через sharepoint designer(есть версионирование и извлечение)  
в этих папках есть различия : во второй через sp designer, появляется версионирование, “понятие извлечения”…  
выбран путь, что код лежит в папке layout, а не в style library, так как в sp designer хуже работать ,чем в visual studio code.

1. Git.

В папке layout (C:\Program Files\Common Files\microsoft shared\Web Server Extensions\15\TEMPLATE\LAYOUTS), в которой есть пользовательские js и css, создан git репозиторий.  
Чтобы отслеживать изменения файлов в папках CustomCss, CustomjsLibs.  
Популярные команды: git status - показывает текущие изменения файлов, git add, git commit, git checkout, git clean, git reset, git log, git diff, git show  
  
есть два репозитория на github, которое относятся к Ривс:  
<https://github.com/spbsmile/rivs_sharepoint> - главный, репозиторий «продакшена», папок CustomjsLibs/CustomCss. Лежит на сервере: server-sp (C:\Program Files\Common Files\microsoft shared\Web Server Extensions\15\TEMPLATE\LAYOUTS\)  
<https://github.com/spbsmile/sharepoint.apps> - локальная разработка, тут лежат все wsp серверные- решения, Лежит на сервере: devsp (C:\sharepoint.apps)  
  
Работать с git можно через консоль git, по cd перейдя на путь проекта, либо через gui программу: git extension, она установлена на devsp, а на server-sp нет

1. Блоги, полезные ресурсы  
   Два хороших блога :  
   Блог Андрея Маркеева : http://omlin.blogspot.ru/

Блок Стаса : http://blog.gandjustas.ru/  
 Так же были куплены записи семинаров Стаса, очень полезные, их расположение:  
 [documents/intranet/Forms/AllItems.aspx](http://intranet/sites/documents/intranet/Forms/AllItems.aspx)

http://sharepoint.stackexchange.com/  
  
подписки на twitter:

<https://twitter.com/paul_tavares> - создатель библиотеки spswidget (для user input),

<https://twitter.com/sympmarc> - создатель библиотеки spservice (удобна для работы с группами (принадлежность человека к группе) и определение тек пользователя)

<https://twitter.com/OfficeDev> от ms  
  
<https://twitter.com/gandjustas> - Стас  
  
<https://twitter.com/amarkeev> - Маркеев  
  
<https://twitter.com/vgrem> - русский, из топа по ответам на stack overflow по sp  
  
<https://twitter.com/eliostruyf> - создатель sp templates

1. Библиотеки js, которые были использованы

по sharepoint:

Camljs.js – использовалась, для caml запросов к списку (в принтерах)

jquery.SPServices-2014.02.js – использовалась для определения принадлежности текущего

пользователя к определенной группе и получении истории изменений элементов списка

jquery.SPWidgets.min.js – использовалась для input контрола выбора нужного сотрудника.

Общие:

moment.js – время   
jquery.js  
jquery.ui.js – модальное окно (выдать заявку)

jquery.ui.datepicker-ru  
boostrap.js – модальные окна(погода)  
  
Многие лежат локально, путь : C:\sharepoint.apps\Core Libs

13. Порядок загрузки js скриптов в sp.

У sp он свой, есть специальная библиотека SOD, хорошо раскрыты эти темы в блоге Стаса: [sharepoint-script-on-demand](http://blog.gandjustas.ru/2013/07/30/sharepoint-script-on-demand/)  
не выполнять, пока не загрузится скрипт sp.js, и тд. Примеры кода: SP.SOD.executeFunc …, SP.SOD.loadMultiple…,

14. asp.net программирование.  
Очень хорошо описано c# asp net программирование в книге (две главы):

Нейгел К., Ивьен Б., Глинн Дж., Уотсон К. C# 4.0 и платформа .NET4 для профессионалов – 2011

Расположение: [http://intranet/sites/documents/intranet/202011.djvu](http://intranet/sites/documents/intranet/%D0%9D%D0%B5%D0%B9%D0%B3%D0%B5%D0%BB%20%D0%9A.%2C%20%D0%98%D0%B2%D1%8C%D0%B5%D0%BD%20%D0%91.%2C%20%D0%93%D0%BB%D0%B8%D0%BD%D0%BD%20%D0%94%D0%B6.%2C%20%D0%A3%D0%BE%D1%82%D1%81%D0%BE%D0%BD%20%D0%9A.%20%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0%20.NET4%20%D0%B4%D0%BB%D1%8F%20%D0%BF%D1%80%D0%BE%D1%84%D0%B5%D1%81%D1%81%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D0%BE%D0%B2%20-%202011.djvu)

Если не считать, js , при программировании на c# под sharepoint используются старые технологии asp.net, вне sp на смену давно пришла технология asp.net mvc.  
  
на русском stackoverflow есть неплохой топик в котором оценивается процесс разработки под sharepoint. Он полезен для понимания, с чем придется столкнуться и чего лучше сразу избегать. Прочтя его, был выбран упор максимально делать все, что можно на клиенте через javascipt. [Что выбрать ASP.NET MVC или SharePoint для Enterprise?](http://ru.stackoverflow.com/questions/453328/%D0%A7%D1%82%D0%BE-%D0%B2%D1%8B%D0%B1%D1%80%D0%B0%D1%82%D1%8C-asp-net-mvc-%D0%B8%D0%BB%D0%B8-sharepoint-%D0%B4%D0%BB%D1%8F-enterprise) Русский stackoverflow по sharepoint не большой, можно всего его перечитать, много чего интересного можно найти.

Перспективен sp framework от ms, релиз для on premise после нг 2016.

15. правила доступа в sp, группы.  
В основном все делается, через панель администратора sp, через группы sp. У каждого списка есть в настройках: разрешение на права доступа, изначально они наследуются от сайта, наследование можно прерывать и ставить уникальные права, как на список, так и на отдельный элемент списка.  
Хорошая статья из блога на эту тему: [sharepoint-permissions](http://omlin.blogspot.ru/2013/02/sharepoint-permissions.html)

16. тестовый пользователь.

Нужен, чтобы проверять поведение сайта, под «правами» обычного пользователя.

Логин: sp-test, пароль: 12345678  
на devsp, есть броузер мозила, в котором портал открывается с учетной записью тестового пользователя.  
так же можно запускать ie c любой учетной записью. ( правая кнопка мыши – открыть от имени другого пользователя )

17. sharepoint apps.  
  
 На портале этот вид решений не используется. требуется повторная идентификация из-за дополнительного домена (это можно решить через групповую политику ), программист Сергей ,вероятно, будет использовать решение sharepoint hosted app. Сначала ошибочно было принято многое делать через sharepoint apps, но после все перешло на cewp.

18. кеш, ctrl f5

Те скрипты, которые добавлены в masterpage, сами автообновляются.  
Выглядит это так:  
<http://intranet/_layouts/15/core.js?rev=rjZicIy%2BXBea34HbdjHJhw%3D%3D>

rev= revision – идентификатор, который меняется

Скрипты, которые подключены в cewp, аналогично, только вручную ставятся идентификаторы.

Чтобы каждому пользователю не требовалось нажимать ctrl f5, крипты переименовываются с name.js на name\_1.js и тогда броузер, подхватывает свежую версию файла, при следующем попадании пользователя.

19. принтеры, офисная техника, другой портал  
  
<http://server-sp-it/sites/wiki>   
  
складской учет офисной техники, картриджей принтеров. По уникальным названиям картриджей идет общий учет их количества. Картриджы могут относится в принтерам на железноводской , уральской и тд. Ведется общая синхронизация количества картриджей по всем подразделениям и история выдачи картриджей. Историю отдельного картриджа можно посмотреть нажав на дату замены, а общую картину нажав на статистику.  
spservice удобна для получения истории изменений элементов списка.

20. совместимость броузеров, основной броузер ie  
  
 Разработка удобна в броузере chrome, но основной броузер – это ie10, 11. И не стоит забывать об этом. Верстка в хроме может сильно различаться с версткой ie. Так же , например, атрибут hidden html5 поддерживается с ie11+…

b) **Детальное, постраничное описание портала intranet**:  
 1.1. Как построены страницы на intranet.  
шапка сайта на всех страницах – это masterpage. (вплоть до картинки). Дальше на всех страницах  
встроена веб часть “редактор сценариев” в которую вставлена разметка.   
скрипты которые общие для всех коллекций сайтов прикреплены в masterpage.  
скрипты которые уникальны для каждой страницы добавляются в вебчасти “редактор сценариев”.  
  
Взаимодействие с серверным кодом на данный момент одностороннее. Т.е нужная информация считывается с сервера на страничку. (погода, курсы валют, Ip). Технически – это visual web part   
которая добавляется на страницу, ее aspx страница встраивается в страницу портала, серверный код записывает в div (с известным id и css style display none) нужные данные, а уже на клиенте по известному id js код считывает и записывает данные куда надо. Двухстороннее взаимодействие планируется в личном кабинете.

2. Папки CustomCss, CustomjsLibs на server-sp для intranet

В этих папках, лежит весь js/css код, который относится к порталу intranet. Он под контролем версий – git. ( Есть исключение в техподдержке для ИТ, некоторые файлы лежат в style library ).  
В папках, js и css файлы отсортированы по разделам портала. Есть общие файлы css/js, которые применяются ко всем коллекциям сайта.   
  
TypeScript – это надмножество над js, любой рабочий js код можно вставить в typescript файл и все будет работать. Если открыть visual studio code то если есть файлы ts и js c одинаковым имением, то js файл будет скрыт. (это пользовательская настройка visual studio code) Ts файл компилируется в js файл. Комбинация клавиш компиляции ts – Ctrl shift b. При изменениях ts надо не забывать компилировать в отличие от js, изменения которого броузер сразу подхватывает.

3. Страницы - детальное описание, файлы js, css, веб части:

js cкрипты, которые общие для всех разделов :

initClocks.ts – задает начальное время часам

initPermision.ts – определяет текущего пользователя и по вхождению к sp группе скрывает верхнюю панель администратора.

searchRightTopCorner – модальный верхний поиск

3.1 Главная

Веб части : VisualWebPartCurrencyWeather, “Редактор сценариев” ( увидеть их можно если нажать “Изменить” )

Js (их подключение в “Редакторе сценариев”):

<script type="text/javascript" src="/\_layouts/15/CustomjsLibs/1.devsp/news.js"></script>

<script type="text/javascript" src="/\_layouts/15/CustomjsLibs/1.devsp/weather\_valutes.js"></script>

<script type="text/javascript" src="/\_layouts/15/CustomjsLibs/1.devsp/coreCalendar\_4.js"></script>

Css:

<link rel="stylesheet" type="text/css" href="/\_layouts/15/CustomCss/1.devsp/currency.css"/>

<link rel="stylesheet" type="text/css" href="/\_layouts/15/CustomCss/1.devsp/calendar/calendar.css"/>

<link rel="stylesheet" type="text/css" href="/\_layouts/15/CustomCss/1.devsp/weather.css"/>

<link rel="stylesheet" type="text/css" href="/\_layouts/15/CustomCss/1.devsp/widgetsemployes.css"/>

<link rel="stylesheet" type="text/css" href="/\_layouts/15/CustomCss/1.devsp/news.css"/>

coreCalendar\_4.js – скрипт, которой выдает новых сотрудников, др сотрудников, и сама итерация календаря.  
  
weather\_valutes.js – считывает погоду и валюту с div, которой заполняется веб частью

news.js – rest запросом, считывает последние 10 новостей с коллекции сайтов “новости”

3.2 Сотрудники

Веб части: две веб части «редактор сценариев»  
в первой поиск, во второй div контейнер widgets  
скрипт widget\_control.js динамически генерирует страницу сотрудников.

понятие additionalEmployeе (в коде такое поле): Это те сотрудники «У которых есть второй департамент» Пример: Зимина Анна Алексеевна, у нее дополнительный департамент – «Руководство».

Технически:

добавлено дополнительное свойство (панель администратора: управление приложениями: управление приложениями службами: user profile service: управление свойствами пользователя: скролл вниз до конца: настраиваемые свойства: Отдел №2 – division, мапится через службу поиска, через схему поиска свойством: otdel2 (т.е в коде otdel2))

При динамическом создании главной страницы сотрудников идет цикл по словарю departmetsTree и у тех элементов словаря, у которых есть свойство additionalEmployeе, эти дополнительные сотрудники добавляются в div под виджет департамента, с display:none.  
Пример: виджет “Руководство” – правая кнопка мыши, показать элемент, <div id="additionalEmployee\_0" class="additionalEmployee">.  
При клике на виджет идет дополнительная проверка на наличие контента у <div id="additionalEmployee\_ и если он есть, то эти сотрудники добавляются в конце после

3.3 Документы

Состоит из страницы «Документы», которая перенаправляет на коллекцию веб сайта “Документы”  
Страница «Документы» - веб часть «редактор сценариев» в которой жестко прописаны

ссылки на библиотеки “Документы”

3.4 Форум

Коллекция сайта. Изменялся только css. Добавлены дефолтные скрипты js, которые есть у каждой мастер страницы портала.

3.5 Техподдержка для пользователей

Веб части: visualwebpartdefineip и «редактор сценариев»

Страница главной коллекции сайтов.

3.6 Техподдержка для сотрудников IT

Три списка и веб часть “редактор сценариев”

На ней включен ajax автообновления списков заявок. (нажать изменить страницу => найти список, => открыть свойства списка и там включить/отключить ajax). Этот ajax очень мешает обновлять   
cewp (веб часть «редактор сценариев»), изменения сбрасываются, поэтому если нужно внести изменения в cewp, то ajax лучше отключить.

На всех четырех коллекций сайтов (главная, документы, форум, сотрудники) есть группа SupportOwner, по которой скрипт initPermission.js (лежит в папке CustomJs) определяя текущего пользователя, делает различный редирект при нажатии на кнопку «техподдержка»   
  
3.7 Личный кабинет.

Wsp решение: C:\sharepoint.apps\UserControlWidgetTasks  
Планируется подключать dll, которая будет работать с внешней базой данных по задачам и эту dll подключить к решению visual web part, которое взаимодействует с интерфейсом личной страницы. Aspx страница visual web part , встраивается в личную страницу сотрудника.

Контакты

Забиякин Макар   
 email: [zabiyakin.makar@gmail.com](mailto:zabiyakin.makar@gmail.com)   
 тел: 8 981 8345963

Skype: mokar4eg  
github: https://github.com/spbsmile