

Структура проекта

Описание директорий

assets - контент, ресурсы используемые в проекте (иконки, изображения);

docs - документация проекта, предоставленная заказчиком;

js - исходный код Javascript - язык управления элементами на сайте;

libs - исходный код Javascript используемых библиотек;

php - исходный код PHP;

styles - исходный код CSS - описание внешнего вида веб-страницы;

templates - HTML разметка, которая используется для структурирования и отображения веб-страницы и её контента;

tools - дополнительные конфигурационные файлы, утилиты и прочее.

Структура директории **js**

Файл **api.js**

Модуль, формирования обращений к API

```
request(url, requestData)
```

Шаблон запроса для get и post методов по api

Параметры:

url : string - Url для обращения

requestData : array_like, optional -

Данные для отправки по post

get(url)

Get-запрос по api

Параметры:

url : string - Url для обращения

post(url, data)

Post-запрос по api

Параметры:

url : string - Url для обращения

data : array_like - Данные для отправки по
post

cache(type, page, url)

Запрос на получение данных из кэша

Параметры:

type: string - Тип текущего
отображения(tree, table)

page: integer, optional - номер "страницы"
в кэше для подгрузки в таблицу

url: string, optional - Url, который
является ключом в кэше, значение - ответ по
данному url

getAll(page)

Получить значение из кэша по странице

Параметры:

page: integer, string - Страница данных,
необходимая для подгрузки. Может принимать
значение 'all' для получения всех данных
из кэша.

postEnterprise()

Формирование данных о предприятии и отправка их по post методу

postDivision()

Формирование данных о подразделении и отправка их по post методу

Файл **change_view.js**

Модуль смены вида отображения данных

removeChilds(element)

Используется для очистки блока с таблицей/деревом. Также подходит для удаления любых дочерних элементов

Параметры:

element: DOM-object - Элемент с детьми, которых необходимо удалить

changeView()

Переключение табличного и древовидного представлений

freezeButton()

Отключение вызова функции смена вида при нажатии на кнопку переключения (заморозка)

Файл **contextmenu.js**

Модуль формирование контекстного меню для работы с таблицей. Удаление/Добавление столбцов, сохранение текущего отображения

add_delete_column()

Формирование контекстного меню и его функционала

deleteColumn(title)

Удаление столбца из таблицы

Параметры:

title: string - Имя заголовка

addColumn(title)

Добавление столбца в таблицу

Параметры:

title: string - Имя заголовка

saveView()

Сохранение текущего отображения в БД

Файл **createNew.js**

Модуль создания нового объекта

createNewObject()

Создание нового подразделения/предприятия

createSelectList(selectlist)

Создание выпадающего списка

Параметры:

selectlist: array_like – Массив данных для
заполнения выпадающего списка

postNew()

*Вызов необходимых функций для отправки post запроса
с данными об объекте*

check_null(start, end)

Проверка на пустоту заполняемых значений

Параметры:

start: integer – Начальное значение
идентификатора поля заполнения данных
end: integer – Конечное значение
идентификатор поля заполнения данных

Файл **dynamic.js**

*Модуль, отвечающий за динамическую подгрузку данных в
табличном отображении*

checkLastElement()

*Выполнение проверок и подгрузка данных, если
пользователь долистал до последнего элемента*

Файл **edit_form.js**

Модуль режима редактирования объекта

edit()

Переключение на вид редактирования, вызов функций для отправки отредактированных элементов

selectList(name)

Формирование выпадающих списков для формы редактирования

Параметры:

name: string - Идентификатор поля, для которого необходимо формирование выпадающего списка

getFieldName(obj)

Поиск поля, где написано имя для выпадающего списка

Параметры:

obj: array_like - Объект, в котором необходимо найти имя

editView()

Генерация полей режима редактирования

findID(name)

Поиск ID объекта

Параметры:

name: DOM-object - Объект, в котором необходимо найти id

removeID()

Удалить ID из заголовка формы просмотра/редактирования

Файл **excel.js**

Модуль экспорта табличного представления в *.xlsx*

createMockBody()

Формирование тела экспортной таблицы

toExcel()

Функция *onclick* для кнопки, обработка данных и условий экспорта

Файл **functions_for_buttons.js**

Модуль, содержащий *callback* функции для кнопок
основного окна

close_edit()

Закреть форму редактирования/просмотра

undo_edit()

Перейти в режим просмотра из режима редактирования
без изменений

reload_cache()

Обновление кэша

delete_object()

Запрос на удаление объекта

accept_filters()

Принять выбранные фильтры и сортировки

filter_open()

Открыть блок фильтров

filter_close()

Скрыть блок фильтров

drop_filters()

Сбросить фильтры

Файл **search.js**

Модуль, отвечающий за поиск и отображение искомых данных

search()

Функция поиска, валидация ввода, *onclick* для кнопки "Найти"

clear(input)

Очистка поля ввода и возврат к виду таблицы/дерева по умолчанию

Параметры:

input: DOM-object - Поле поиска

inputEvent(element)

Обработчик событий при вводе в строку поиска

Параметры:

element: DOM-object - Поле поиска

Файл **show_edit.js**

Модуль, содержащий методы отображения информации по существующему объекту

selectItemEnterprise(id)

Загрузить форму просмотра для предприятия с идентификатором *id*

Параметры:

id: integer - Идентификатор предприятия
для просмотра

selectItemDivision(id)

Загрузить форму просмотра для подразделения с идентификатором *id*

Параметры:

id: integer - Идентификатор подразделения
для просмотра

findName(id, obj)

Поиск имени для подставления по ID в выпадающих списках

Параметры:

id: integer - Идентификатор необходимого имени
obj: array_like - Массив объектов для поиска

getType(object)

Определение типа объекта (подразделение, предприятие)

Параметры:

object: DOM-object - Объект для определения

blockSelect(freeze)

Запрет на генерацию событий указателя

Параметры:

freeze: bool - Текущее состояние запрета

Файл **table.js**

Модуль формирования табличного представления данных

isAllow(item)

Проверка содержимого объекта данных на соответствие фильтрам

Параметры:

item: array_like - Объект данных, полученный по api

createHead()

Формировании заголовков таблицы

createBody(data, tbody, backlightPattern)

Формирование тела таблицы, либо добавление новых данных в существующее тело таблицы

Параметры:

data: array_like - Данные для добавления в таблицу
tbody: DOM-object, optional - Существующее тело таблицы
backlightPattern: string, optional - Искомая строка в подстроке

table(data, backlightPattern)

Формирование таблицы

Параметры:

data: array_like, optional - Данные для добавления в таблицу
backlightPattern: string, optional - Подсветка искомой подстроки

Файл **tree.js**

Модуль формирования древовидного представления данных

clearTreeData()

Очистка информации о наличии детей у объекта, очистка кэша с объектами данных, очистка хранилища сессии

getData(url)

Получить данные из кэша для дерева

Параметры:

url: string - Url для получения детей объектов

createElemWithAttr(item, attributes)

Создание DOM-объекта, с заданными атрибутами

Параметры:

item: string - Тип объекта, который создается
attributes: object - Свойства, присваиваемые к объекту

hasChilds(id)

Есть ли дочерние элементы у объекта

Параметры:

id: integer - Идентификатор родителя

openChilds(element, button)

Загрузка детей объекта, раскрытие/закрытие узла

Параметры:

element: DOM-object - Родитель
button: DOM-object - Кнопка раскрытия/закрытия детей

createLi(classItem, item, searchPattern)

Создание элементарного List Item элемента дерева и его необходимого заполнения

Параметры:

classItem: string - Класс узла дерева
(root, child)
item: object - Объект данных
searchPattern: string, optional - Пользовательская поисковая строка, используется для подсветки вхождений

createUl(classItem, arr, searchPattern)

Создание списка элементов дерева

Параметры:

classItem: string - Класс узла дерева
(root, child)
arr: array_like - Объект данных
searchPattern: string, optional -
Пользовательская поисковая строка,
используется для подсветки вхождений

createView(className, arr, searchPattern)

Создание корневых элементов для дерева

Параметры:

className: string - Класс узла дерева
(root, child)
arr: array_like, optional - Объект данных
searchPattern: string, optional -
Пользовательская поисковая строка

tree(data, backlightPattern)

Создание дерева

Параметры:

data: array_like, optional - Данные для
отображения
backlightPattern: string, optional -

Искомая строка

Файл **sortTable.js**

Модуль, отвечающий за сортировку таблицы

sort(header)

Сортировка таблицы по указанному столбцу

Параметры:

header: DOM-element - Элемент заголовка столбца таблицы

Структура директории **libs**

Файл **dragtable.js**

Модуль, отвечающий за функционал перетаскивания столбцов таблицы

Файл **jquery-ui.min.js, jquery.min.js**

Библиотека jquery

Файл **selectize.min.js**

Модуль для выпадающего списка с динамическим поиском

Файл **tableToExcel.js**

Модуль для формирования таблицы excel

Структура директории **php**

Файл **api.php**

Модуль, содержащий реализацию методов обращения к API заказчика

get(\$url)

Запрос данных по url

Параметры:

\$url: адрес сервера, ресурса

```
post( $url, $data = array() )
```

Отправка данных по url

Параметры:

\$url: string - Адрес сервера, ресурса

\$data: array - Набор передаваемых данных

Файл **auth.php**

Модуль, содержащий интеграцию с системой аутентификации заказчика

Файл **cache.php**

Модуль, реализующий помещение зашифрованных данных в кэш

```
cache( &$data )
```

Помещение data в локальное хранилище, осуществляя шифрование данных

Параметры:

\$data: array - Набор данных

Файл **get_cache.php**

Модуль, осуществляющий дешифровку и получение данных из кэша в зависимости от типа отображения (табличное или древовидное). Для табличного - из кэша происходит запрос по 25 элементов, для древовидного - запрос кэшированного ответа от сервера по API url ключу

Файл **hash.php**

Модуль, осуществляющий шифрование строковых данных

```
mc_encrypt( $encrypt, $key )
```

Шифрование данных

Параметры:

`$encrypt: array` - Набор исходных данных

`$key: string` - Ключ для шифрования

```
mc_decrypt( $decrypt, $key )
```

Дешифрование данных

Параметры:

`$decrypt: array` - Набор зашифрованных данных

`$key: string` - Ключ шифрования

Файл **reload_cache.php**

Модуль, осуществляющий запрос актуальных данных и их обновление в кэше

Файл **sort_cache.php**

Модуль, осуществляющий сортировку кэша

```
sort_high_low( $left, $right )
```

Быстрая сортировка кэша по возрастанию/убыванию

Параметры:

`$left: int` - левая "граница" словаря

`$right: int` - правая "граница" словаря

```
sort_null_not( $left, $right )
```

Быстрая сортировка кэша по пустые/непустые

Параметры:

`$left: int` - левая "граница" словаря

`$right: int` - правая "граница" словаря

Структура директории **templates**

Файл **index.php**

Отображение главной страницы, содержит определение всех компонент сайта.

Заголовок файла (тег head) - Содержит ссылки на основные CSS стили

Содержание веб-страницы (тег body)

Блок "container-fluid" - Блок шапки сайта

Блок "container" - Элементы панели управления

Блок "mySidebar" - Боковая панель от контента для размещения параметров фильтрации и сортировки

Блок "main" - Основной функционал сайта

Блок "container_information" - Заголовки функционала сайта

Блок "block_edit" - Форма для создания нового объекта

Структура директории **styles**

Файл **css.css**

Общие стили

Структура директории **tools**

Файл **README.md**

Руководство по проекту