Документация по реализации запросов со стороны клиента (браузера) к серверной части приложения и обратно.

Используемые языки: Javascript (Фреймворк angularJS), html (5), css

Используемые технологии: ajax, restfull

Обмен данными между клиентом и сервером происходит в формате JSON

В 1 главе описана структура запросов.

В 2 главе описана структура ответов.

В 3 главе описан разработанный нами сервис для реализации запросов и общения с серверной частью.

Конкретное апи описано в других документах здесь только общие правила их формирования.

# Структура запросов

Описание запросов:

1. Методы - (GET, POST, PUT, DELETE, OPTIONS, ~~HEAD, PATCH, TRACE, LINK, UNLINK, CONNECT~~)
2. Запрос - url адрес ресурса
3. Предопределенные параметры запроса в адресной строке
4. Предопределенны сегменты запроса в адресной строке (их порядок и наличие опционально)
   1. token хеш-строка (кука) для идентификации пользователя (смотри примечание)
   2. parent
   3. pid
   4. self
   5. sid
   6. child
   7. cid
5. Данные запроса

Это простая переменная, объект либо список (массив) объектов в формате json

int | float | string - простая переменная

{“Id” int, “Name” string, “Price” float, “Flag” bool, …} - объект

[{“Id” int, “Name” string, “Price” float, “Flag” bool }] - список объектов

Объекты в свою очередь помимо простых значений могут включать другие объекты либо списки объектов.

{} – Обязательные параметры

[] – Не обязательные параметры

Примеры и синтаксис запроса:

http://www.domain.ru/api /v1/modname/ctrlname/profile/23/orders/34

Примечание:

Возможные варианты передачи токена и порядок их поиска.

1. в URI – только если это указано в описании метода
2. в параметре URI с именем access-token
3. в Header с именем X-Access-Token и значением {token}
4. в сессионной куке

Если токен передан одновременно несколькими способами, то используется только тот который был найден первым, остальные токены игнорируются

# Структура ответов

Описание ответов:

Ответ на все ajax запросы приходит в следующей структуре:

{“errorCode” int, “errorMessage” string[, …]}

errorCode - код ошибки (0 = ошибок нет, 0 != код ошибки)

errorMessage - сообщение ошибки

[…] - Данные любой структуры в формате json (опционально, может не быть).

# Angularjs cервис (на стороне клиента - браузера)

Скрипт: /services.js

## Общение с сервером

Запросы сервиса используют архитектуру REST и формат передачи данных в формате json

Список запросов реализующих общение с серверной частью:

### Server.QUERY(url, method[, data])

Абстрактный запрос на одновременную передачу и получения данных с сервера.

url - относительный или абсолютный адрес ресурса

method - метод запроса: GET, POST, PUT, DELETE, HEAD, OPTIONS, PATCH, TRACE, LINK, UNLINK, CONNECT

data - данные отправляемые на сервер (любой объект, переманная (javascript))

параметр необязателен

Возвращает объект promise

### Server.GET(url)

Абстрактный запрос на получения данных с сервера.

url - относительный или абсолютный адрес ресурса

Возвращает объект promise

### Server.POST(url[, data])

Абстрактный запрос на передачу данных на сервер.

url - относительный или абсолютный адрес ресурса

data - данные отправляемые на сервер (любой объект, переманная (javascript))

параметр необязателен

Возвращает объект promise

### Server.PUT(url[, data])

Абстрактный запрос на передачу данных на сервер.

url - относительный или абсолютный адрес ресурса

data - данные отправляемые на сервер (любой объект, переманная (javascript))

параметр необязателен

Возвращает объект promise

### Server.DELETE(url)

Абстрактный запрос на передачу данных на сервер.

url - относительный или абсолютный адрес ресурса

Возвращает объект promise

## Асинхронный обмен и сохранение данных между контроллерами AngularJS

Централизованное сохранение данных в сервисе на стороне клиента и асинхронный обмен ими между контроллерами AngularJS. В переделах приложения. До ее перезагрузки.

### Server.SetData(index, value)

Сохранение данных в сервисе

index - индекс под которым сохраняются передаваемые в хранилище данные.

value - сохраняемые данные (любой объект, переменная (javascript))

Ничего не возвращает

### Server.GetData(index)

Получение данных из сервиса

index - индекс под которым сохраняются передаваемые в хранилище данные.

Возвращает сохраненные данные по указанному индексу

### Server.RemData(index)

Удаление данных из сервиса

index - индекс под которым сохраняются передаваемые в хранилище данные.

Ничего не возвращает

## Синхронный обмен данными между контроллерами AngularJS

Создание канала и подписка на каналы в сервисе для синхронного обмена данными в случае их изменения.

### Server.SubscribeEvent(chanel[, param …])

Событие канала.

chanel - имя канала (произвольная строка)

param - произвольный набор параметров через запятую. Параметры передаются всем функциям, которые подписаны на указанный канал (методом 1.3.2).

### Server.Subscribe(chanel, function)

Подписка на канал

chanel - имя канала (произвольная строка)

function - функция которая будет вызвана при возникновении события в канале. Ей будут переданы данные посылаемые в этот канал (методом 1.3.1).