**API**

**Описание API для работы сервером StoreHouse5 через HTTP**

Сервер SH5WAPI2 предоставляет возможность получать и передавать информацию на сервер StoreHouse5 посредством файлов формата JSON ([www.json.org](http://www.json.org/)) через протокол HTTP.

Сервер StoreHouse5 общается с внешним миром исключительно с помощью процедур, идентифицируемых по имени. Процедура в качестве параметров принимает датасеты (таблицы) и в ответ заполняет их запрошенной информацией. Дататсеты SH5 могут быть однострочными и многострочными.

**Всего доступно две операции:**

• Запросить список датасетов процедуры: /api/sh5struct

• Выполнить процедуру: /api/sh5exec

По указанным путям надо выполнить POST-запрос с передачей в Content JSON-файла следующего вида:

{

 "UserName": "Admin",

 "Password": "12345",

 "procName": "XDivisions"

}

• UserName (string)- имя пользователя StoreHouse5

• Password (string)- пароль пользователя StoreHouse5

• procName (string)- название процедуры StoreHouse5

Эти три параметра являются обязательными в любом запросе к серверу. В результате выполнения запроса будет прислан ответный JSON-файл. Пример ответа:

{

 "errorCode": 1,

 "errMessage": "Процедура XDivisions не найдена.",

 "Version": "0.1.4"

}

• errorCode (number)- код ошибки

• errMessage (string)- текст ошибки

• Version (string)- версия сервера SH5WAPI2

Эти три значения приcутствуют обязательно в любом ответе от сервера. В случае успешного выполнения запроса errorCode будет равен 0, и будут присутствовать и другие элементы:

• actionName (String) - название процедуры StoreHouse5

• actionType (String) - «Structure» либо «Execute» - запрошенная операция

• shTable (массив) - список объектов «структура таблицы» либо «таблица с данными»

Значения errorCode: 0 либо 1.

Файлы JSON должны быть в кодировке UTF-8.

**Запрос списка датасетов процедуры**

Запрос посылается по пути /api/sh5struct и содержит только три обязательных параметра, перечисленных выше. Ответ содержит массив элементов «структура таблицы».

Объект «структура таблицы» содержит следующие элементы:

• head (String) - идентификатор таблицы

• SingleRow (Bool) - однострочный датасет (true) или многострочный (false)

• fields (массив) - список объектов «поле таблицы»

Объект «поле таблицы» содержит следующие элементы: • path (String) - оригинальное имя поля

• type (String) - тип поля

• size (String) - размер поля

• alt (String) - необязательный, альтернативное имя поля

Пример успешного ответа:

{

 "errorCode": 0, "errMessage": "OK", "Version": "0.1", "actionName": "Divisions",

 "actionType": "Structure",

 "shTable":

 [

 {

 "head": "103", "SingleRow": false,

 "fields":

 [

 { "path": "1", "type": "tUint32", "size": 4, "alt": "Rid" },

 { "path": "4", "type": "tGuid", "size": 16, "alt": "Guid" },

 { "path": "3", "type": "tStrZ", "size": 255, "alt": "Name" },

 { "path": "7\\$Qush", "type": "tStrP", "size": 0 },

 { "path": "7\\$PDocNum", "type": "tStrP", "size": 0 },

 { "path": "7\\$IDocNum", "type": "tStrP", "size": 0 },

 { "path": "7\\$GDocNum", "type": "tStrP", "size": 0 },

 { "path": "7\\Chef", "type": "tStrP", "size": 0 }

 ]

 },

 {

 "head": "103#1", "SingleRow": true,

 "fields":

 [

 { "path": "239", "type": "tUint32", "size": 4, "alt": "MaxCount" },

 { "path": "240", "type": "tUint32", "size": 4 }

 ]

 }

 ]

}

**Выполнение процедуры**

Запрос посылается по пути /api/sh5exec. Передача данных в процедуру осуществляется через параметр «Input». Либо, для совместимости со старой версией, через параметр «dsParams». Второй метод устарел и в будущем может быть исключён. Параметр «Input» (массив) содержит список объектов «таблица с данными». Параметр «dsParams» (объект) содержит объекты «поле с данными», при этом допускаются только альтернативные названия полей .

Ответ содержит массив элементов «таблица с данными».

Типы «tShortDate» и «tLongDate» должны передаваться в формате «yyyy-mm-dd». И если в ответе есть такие типы данных, то они будут именно в этом формате.

Объект «таблица с данными» содержит следующие элементы:

• head (String) - идентификатор таблицы

• original - массив оригинальных названий полей (string)

• fields - массив альтернативных названий полей (string)

• values - массив, содержащий массивы значений полей.

• status - необязательный, массив статусов записей (string)

Пример запроса с параметром «Input»:

{

 "UserName": "Admin", "Password": "", "procName": "GDoc0",

 "Input":

 [

 { "head": "111", "original": [ "1" ], "values": [ [ 3 ] ] }

 ]

}

Некоторые процедуры для модификации/удаления записи в таблице требуют передачи её статуса. В этом случае в объект «таблица с данными» надо добавлять массив «status», в котором перечислить нужные статусы для каждой записи. Возможные варианты: «Insert», «Modify», «Delete». По умолчанию все записи получают статус «Insert».

Пример запроса со статусами записей:

{

 "UserName": "Admin", "Password": "", "procName": "ModCountries",

 "Input":

 [

 {"head": "231",

 "original": [ "1", "2", "3" ],

 "values":

 [

 [ 1, null, 8 ],

 [ "643", "UTO", "NET" ],

 [ "Россия", "Утопия", "Нетландия" ]

 ],

 "status": [ "Modify", "Insert", "Delete" ]

 }

 ]

}

Объект «поле с данными» - это массив, содержащий значения поля для всех записей таблицы. Пример запроса с параметром «dsParams»:

{

 "UserName": "Admin","Password": "",

 "procName":"CntrSpecs",

 "dsParams":

 { "contractorRid":["2"] }

}

**Настройка сервера**

Параметры в командной строке

/Install инсталлировать как сервис

/Uninstall деинсталлировать сервис

/Desktop запустить как приложение, остановка через иконку в трее

/Silent Не показывать окна при инсталляции/деинсталляции сервиса

/Name:SH5WAPI2 имя сервиса/сервера

/Port:9797 TCP порт, на котором сервер ждёт запросов

Сервер в командной строке получает номер TCP-порта, на котором он ожидает подключения клиентов, и собственное имя. По умолчанию используется порт 9797 и имя SH5WAPI2. Порт - понятно зачем нужен, а вот имя используется для определения подкаталога, где сервер берёт остальные свои настройки и пишет лог-файлы, а в случае запуска сервисом - это ещё и имя сервиса. В указанном подкаталоге (=имени) от папки запуска сервера сервер читает текстовый конфигурационный файл CONFIG.INI следующего вида:

;уровень логирования 0-не вести лог; 1-вести основные логи;

;2-сохранять в файлах последние Headers и Content

LogLevel = 2

;для подключения к серверу StoreHouse5 использовать: 0-локальный протокол; 1-TCP;

sh5tcp = 0

;имя сервера StoreHouse5 для подключения через локальный протокол

sh5server = SDBSERV

;компьютер где запущен сервер StoreHouse5 для подключения через TCP

sh5host = 127.0.0.1

;порт, на котором работает сервер StoreHouse5 для подключения через TCP

sh5port = 7771

;кодовая страница базы данных StoreHouse5

sh5cp = 1251

;SSL файл сертификата, если требуется защищённое соединение

;SSL\_cert = cert1.pem

;SSL файл приватного ключа, если требуется защищённое соединение

;SSL\_priv = priv1.pem

;SSL пароль к приватному ключу, если он зашифрован

;SSL\_psw =

;файл центра сертификации, необязательный

;SSL\_CA =

Для работы с сервером SH5 используются следующие библиотеки:

borlndmm.dll

sdbcli.dll

domm.dll

dset.dll

cc3260.dll

Они должны лежать в папке запуска сервера. В подкаталог следующего уровня LOGS сервер пишет свои лог-файлы. В подкаталоге TEMPLATE от папки запуска сервера лежат необязательные JSON-файлы с альтернативными названиями полей датасетов процедур StoreHouse. Это на случай, если кому-то хочется обращаться к полям не по исходному имени, а по «более понятному». При наличии альтернативного имени возможность работать по исходному имени сохраняется - действуют оба. Файл с альтернативными именами полей создаётся из ответа на запрос структуры датасетов процедуры. К нужным полям надо просто добавить атрибут «alt». Название файла состоит из имени процедуры плюс «\_desc.json». Для проверки работы сервера можно использовать тестовую программу Swat.exe