Оглавление

Вызовы JSON RPC

Для вызова по JSON RPC используется стандарт 2.0, описание см тут
https://www.jsonrpc.org/specification

Для вызова надо в URL сервера использовать **/rpc**

Например:
167.10.20.122/rpc
erp-sng/galaktika.ru/rpc

Поскольку, vip является объектно-ориентированным языком, то в method надо указать имя класса и имя метода через точку.

Например:
“method”: “callAtm.delaysec”
Тут callAtm имя класса, delaysec -имя метода

Авторизация

Для вызовов через /rpc используется BASIC авторизация. Реализован пул подключений для одноименных пользователей. Размер пула задается при старте webserver через параметр -b<num>

Например:
webserver -b25

По умолчанию значение равно нулю, что означает неограниченный размер пула подключений.
При получении запроса по /rpc веб сервер пытается найти свободное подключение, и если оно найдено, то использует его. Если нет ни одного свободного - создает новое. Если размер пла не ограничен - в этом случае всегда создается новое подключение. Если

ограничен - при достижении максимального размера пула запросы будут выстраиваться в очередь.
Например, если размер пула 25, а пришло 40 запросов одновременно, то 15 встанут в очередь.

Время жизни неиспользуемого подключения задается так же, как и для сессии UI, через параметр -x<num>, в секндах, и по умолчании равно 7200 сек или 2 часа

Например:
webserver -b25 -x600

Пример:

См приложенный пример PostExample,java

Результат работы:

```
{
  "id": 3,
  "jsonrpc": "2.0",
  "method": "callAtm.delaysec",
  "params": {
    "d": "10"
  }
}
.....
{"jsonrpc": "2.0", "result":1, "id":3}
```

Вызовы чистый REST

Для вызова через чистый REST надо использовать **/rest**

Например:
167.10.20.122/rest
erp-sng/galaktika.ru/rest

По сути, вызов мало чем отличается от вызова JSON RPC, вызов перенаправляется на метод **HandleRequest** класса **vRestHandler**. Присылаемый запрос идет единственным параметром в виде строки.

По сути мы имеем единый обработчик для всех входящих REST запросов.

Изменить обработчик можно через параметры
-j - имя класса
-h - имя метода

Например:
webserver -jvMyRESTHandler -hHandleMyREST

Вызовы через стандартный API

Для вызова через стандартный API надо использовать **/api**

Авторизация

В этом случае используется authentication token (bearer) и вызовы с одним токен всегда используют одно и то же подключение. В этом режиме запрещено повторное подключение под одним и тем же пользователем.

Использование

При попытке отправки асинхронных запросов с одинаковым токеном второй и более запросов получит ошибку “connection is busy”. Все запросы через одно подключение должны выполняться последовательно, в отличии от /rpc, где допускается асинхронное выполнение запросов под одним и тем же юзером.
Очевидно, что под разными подключениями асинхронные запросы допускаются.

Пример:

См приложенный пример TestREST.vih и TestREST.vip