

# Использование REST hook для авторизации пользователя по домену

## Описание

Допустим, возникла задача ограничить доступ пользователей к серверу, основываясь на домене, например, разрешить просмотр видео только с определенного домена. Эта задача может быть решена при помощи REST hooks.

WCS передает в REST-запросе типа 1 `connect` к бэкенд-серверу поле `origin`, содержащее доменное имя WCS-сервера, по которому к нему обратился пользователь, например

```
{
  "nodeId" : "5tWOFn5ZoMQs22KrEls2Ulhee57hQ09D",
  "appKey" : "defaultApp",
  "sessionId" : "/192.168.1.45:53438/abcdef0123456789",
  "useWsTunnel" : false,
  "useWsTunnelPacketization2" : false,
  "useBase64BinaryEncoding" : false,
  "mediaProviders" : [ "WebRTC", "MSE", "WSPlayer" ],
  "clientVersion" : "0.5.28",
  "clientOSVersion" : "5.0 (Windows)",
  "clientBrowserVersion" : "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0",
  "keepAlive" : false,
  "origin" : "https://test2.flashphoner.com:8888"
}
```

Таким образом, для авторизации пользователя по домену необходимо реализовать REST hook `/connect`.

## Пример реализации REST hook типа 1 для авторизации по домену

### Требования к реализации

1. REST hook типа 1 должен быть доступен на веб-сервере как `http://yourhost/rest-hooks/connect`
2. REST hook должен обрабатывать POST `application/json` HTTP запросы.
3. REST hook должен возвращать в теле ответа в точности то же, что он получил в запросе (зеркало), за исключением настройки `restClientConfig`.

4. WCS сервер должен быть настроен на работу с REST hook следующим образом:

```
ssh -p 2001 admin@localhost
>update app -l http://yourhost/rest-hooks defaultApp
```

Обращения к REST методу можно отслеживать в [логе сервера](#)

```
tail -f /usr/local/FlashphonerWebCallServer/logs/server_logs/flashphoner.log
```

## Код примера и его разбор

В первых строках скрипта определяется метод и декодируется тело запроса. Здесь же задаем домен для авторизации:

```
<?php

$api_method = array_pop(explode("/", $_SERVER['REQUEST_URI']));
$incoming_data = json_decode(file_get_contents('php://input'), true);
$domain = "yourdomain.com";
```

Начинается обработка метода `connect`. Здесь определяется переданное в запросе поле `origin` и заполняется поле `restClientConfig` для ответа:

```
switch($api_method) {
    case "connect":

        $origin = $incoming_data['origin'];

        //logs
        error_log("sessionId: " . $incoming_data['sessionId']);
        error_log("origin: " . $origin);

        $rest_client_config =
        json_decode(file_get_contents('rest_client_config.json'), true);
        $incoming_data['restClientConfig'] = $rest_client_config;
        ...
}
```

Проверка домена. Если домен не найден, вызывается функция `ubnormalResponse` для формирования отрицательного ответа 403

```
switch($api_method) {
    case "connect":
        ...
        $found = strpos($origin, $domain);

        if ($found !== false){
            error_log("User authorized by domain " . $domain);
        } else {
            error_log("User not authorized by domain: " . $domain . "

```

```
Connection failed with 403 status.");
    ubnormalResponse(403);
}
break;
...
}
```

Вывод ответа на запрос

```
header('Content-Type: application/json');
echo json_encode($incoming_data);
```

Функция `ubnormalResponse()` завершает скрипт:

```
function ubnormalResponse($code) {
    if ($code == 403) {
        header('HTTP/1.1 403 Forbidden', true, $code);
    } else {
        header(':', true, $code);
    }
    die();
}
?>
```



Пример скрипта целиком



Содержимое `restClientConfig` из файла `rest_client_config.json` для  
ответа на запрос `/connect`

