|  |
| --- |
| C:\Users\Cloudium_PS\Documents\Cloudium\temp\Black Box logo\BB.png |
| InvisaPC REST Interface user manual ENG-005-020 |
|  |

**Document Number:** ENG-0005-023

**Author:** Marcus McInerney

**Status:** Draft

**Rev**:

**Date:**

|  |
| --- |
| This document provides instructions on how to launch a connection between an InvisaPC Receiver and InvisaPC Transmitter using the InvisaPC REST Interface |

Master Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author / Editor** |
| July 16th, 2018 | 0.1 | Initial Draft of Document | Marcus McInerney |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[1 Table of Contents 3](#_Toc519782907)

[2 Introduction 4](#_Toc519782908)

[3 References 4](#_Toc519782909)

[3.1 Boxilla documentation: 4](#_Toc519782910)

[3.2 InvisaPC documentation: 4](#_Toc519782911)

[4 Configuration requirements 5](#_Toc519782912)

[4.1 Enable HTTP on an InvisaPC Receiver that is managed by Boxilla 5](#_Toc519782913)

[4.2 Connection configuration for Boxilla managed Rx 5](#_Toc519782914)

[4.3 Connection configuration for unmanaged Rx 5](#_Toc519782915)

[5 REST interface authorization 5](#_Toc519782916)

[6 REST commands 5](#_Toc519782917)

[6.1 Force connection 6](#_Toc519782918)

[6.1.1 Request class JSON object 6](#_Toc519782919)

[6.1.2 Request class parameter formats 6](#_Toc519782920)

[6.2 Terminate connection 7](#_Toc519782921)

[6.2.1 Request class JSON object 7](#_Toc519782922)

[6.2.2 Request class format 7](#_Toc519782923)

[6.3 Make connection 7](#_Toc519782924)

[6.3.1 Request class JSON object 7](#_Toc519782925)

[7 REST interface status codes 8](#_Toc519782926)

[7.1 HTTP status codes 8](#_Toc519782927)

[7.2 Note on HTTP 200 status code 8](#_Toc519782928)

[8 APPENDIX 8](#_Toc519782929)

[8.1 Force connect issued to an InvisaPC Receiver with a connection active 8](#_Toc519782930)

[8.2 Force connect whereby the InvisaPC Transmitter or target VM is off-line 8](#_Toc519782931)

[8.3 Issuing a force\_connect for a shared mode connection 8](#_Toc519782932)

# Introduction

This document provides instructions on how to establish a connection between InvisaPC Receiver and InvisaPC Transmitter using the InvisaPC REST Interface.

# References

## Boxilla documentation:

<https://www.blackbox.com/en-us/store/Detail.aspx/Boxilla-Enterprise-Level-KVM-and-AV-IT-Manager/BXAMGR>

## InvisaPC documentation:

<https://www.blackbox.com/en-us/store/Detail.aspx/InvisaPC-KVM-Over-IP-Receiver---Dual-Head-DVI-D-USB-HID-USB2-0/DTX1002-R>

# InvisaPC Receiver configuration requirements

In order to establish a connection using the InvisaPC REST interface, a connection must be configured and associated with an InvisaPC Receiver user. Refer to section [3](#_References) for instructions on how to configure a connection.

## Enable HTTP on an InvisaPC Receiver that is managed by Boxilla

HTTP REST interface is disabled by default if the InvisaPC Receiver is managed by Boxilla. The Boxilla user manual (section [3.1](#_References)) provides instructions on how to enable HTTP REST Interface on a managed InvisaPC Receiver.

## Connection configuration for Boxilla managed Rx

The Boxilla user manual (section [3.1](#_References)) provides instructions on how to configure a connection and associate the connection with a user.

## Connection configuration for unmanaged Rx

The InvisaPC user manual (section [3](#_References).2) provides instructions on how to configure a connection and associate the connection with a user.

# Client HTTP PUT request settings

This section details the settings that must be configured in the client HTTP interface.

## Authorization

The REST interface requires authorization by means of Basic Auth using the following:

**username:** rest\_user **password:** 18\_22\_33\_AA

## Headers

The following header key-value pairs must be set in the client HTTP interface in order to execute a PUT request to the InvisaPC HTTP REST interface.

### Content-Type

**Key:** Content-Type **Value:** application/json

### Authorization

**Key:** Authorization **Value:** Basic cmVzdF91c2VyOjE4XzIyXzMzX0FB

### Cache-control

**Key:** cache-control **Value:** no-cache

# REST commands

This section describes supported HTTP URI’s. In the examples provided, the xxx.xxx.xxx.xxx field represents the IP address of the InvisaPC Receiver to which the command is being issued. The commands described are supported by InvisaPC Receiver and are not supported by InvisaPC Transmitter. The force\_connection URI establishes a connection based on connection options that are set in the configuration

## Force connection

 (PUT) URI: http://xxx.xxx.xxx.xxx:7778/control/connections

This URI launches a connection between an InvisaPC Receiver and target appliance (an InvisaPC Transmitter or a supported VM). The force\_connect URI is supported in InvisaPC 4.3 and later. Requirements:

* The connection must be configured in order to be launched. For further information, see section [4](#_Configuration_requirements). Note: If a connection is configured via the Boxilla user interface then the user associated with the connection must log out and log back in to the InvisaPC OSD in order to allow the connection to be established using the Force connection. The connection configuration will allow the user to determine the following connection options:
	+ Connection type (private or shared)
	+ Extended desktop
	+ USB redirection
	+ Audio
	+ Persistent connection
* The JSON object specified in [6.1.1](#_Request_class_json) must be included in the PUT request.

### Request class JSON object

The force connection URI requires a JSON request object in the following format:



The action parameter is used by the interface as a switch and remains fixed. The connection parameter corresponds to the connection name chosen when configuring the connection. The user parameter corresponds to the user that is associated with the connection.

### Request class parameter formats

| **Parameter** | **Format** |
| --- | --- |
| action | “force\_connection” |
| connection | 32 byte char array |
| user | 32 byte char array |

Table 2. Force connection request class formats

The connection and user parameters can have a maximum of 32 characters.

## Terminate connection

(PUT) URL: http:// xxx.xxx.xxx.xxx:7778/control/connections

This URI terminates any active connection on an InvisaPC Receiver. Note that InvisaPC Receiver does not support concurrent connections, i.e. there can be no more than one active connection running on the appliance at any time.

### Request class JSON object

The terminate connection URI requires a JSON request object in the following format:



The action parameter is used by the interface as a switch and remains fixed.

### Request class format

| **Parameter** |  **Format** |
| --- | --- |
| action | “terminate\_connection”  |

Table 3. Force connection request class formats

## Make connection

(PUT) URL: http:// xxx.xxx.xxx.xxx:7778/control/connections

This URI launches a connection between an InvisaPC Receiver and target appliance (InvisaPC Transmitter or supported VM) with the following **fixed** parameters:

* Resolution 1920x1080
* 32 bit color depth
* Shared connection type
* NLA (Network Level Authentication) disabled
* Hotkey set to default (PrintScrn key)

### Request class JSON object



The action parameter is used by the interface as a switch and remains fixed. The transmitter parameter corresponds to the IP address of the InvisaPC Transmitter or supported VM. The user parameter corresponds to the user associated with the connection.

# REST interface status codes

## HTTP status codes

Table 1 specifies the HTTP response status codes used by the REST interface.

| Code | Description |
| --- | --- |
| 200 OK | The command was received and understood and is being processed. |
| 400 Bad Request | The client issued an unsupported command to the server |
| 401 Unauthorized | Authentication failure |
| 500 Internal Server Error  | An error occurred during the processing of the command |

Table 1. HTPP status codes used by REST interface

## Note on HTTP 200 status code

Status code 200 is returned to the client provided that:

* The HTTP REST Interface is enabled on the InvisaPC Receiver
* URI is correct
* The authorization details are correct
* JSON object request parameters are valid

It is important to note that the status 200 indicates that the command was received and understood by the InvisaPC Receiver. It does not necessarily indicate that the command was executed. The onus is on the client to ensure that the InvisaPC Receiver and Transmitter or target VM is in the correct state before and after the command is issued.

# APPENDIX

This appendix provides usage examples of the REST commands described in section 6.

## Force connect issued to an InvisaPC Receiver with a connection active

If a force\_connect command is issued to an InvisaPC Receiver which has an active connection, the active connection will be terminated and the connection specified in the force\_connect request JSON body will be established. The same principle applies to make\_connection.

## Force connect whereby the InvisaPC Transmitter or target VM is off-line

On issuing a force\_connection request to an InvisaPC Transmitter or VM, the client must first ensure that the InvisaPC Transmitter is online. This means that a force\_connect request to an InvisaPC Receiver whereby the target InvisaPC Transmitter is offline will receive a status code 200. This status codes does not indicate that the connection was established but rather that the command was received and is being processed.

## Issuing a force\_connect for a shared mode connection

Issuing a force connect command for a shared mode connection will behave identically to launching a shared mode connection from InvisaPC OSD or Boxilla user interface.