



## TECH DOC

Date

10/2024

Version 1.0.2

# REST-API Guide



114, 3F, No. 21  
Ln. 168, Xingshan Rd.  
Neihu Dist., Taipei, Taiwan

**Taiwan**

20 Fairbanks,  
Ste, 170 Irvine, CA 92618  
California, U.S.A

**America**





# REST-API Guide

## Developer Guide

### Table of Contents

Introduction .....	4
Software requirements .....	4
Naming .....	5
Authentication .....	6
➤ Oauth2/token .....	6
➤ Authentication Without Password .....	8
Media Transfer .....	9
➤ Media Transfer- POST files/new .....	9
➤ Media Transfer- GET files/id .....	12
➤ Media Transfer- POST files/id .....	13
➤ Media Transfer- POST files/find .....	15
➤ Media Transfer- POST files/delete .....	17
Playback .....	19
➤ Playback- POST app/exec .....	19
➤ Playback- POST app/start .....	22
➤ Playback-POST app/switch .....	25
➤ Playback- POST v2/task/notify .....	27
Hardware Management .....	29
➤ Hardware Management- POST hardware/display .....	29
➤ Hardware Management- POST hardware/rs232 .....	32



# REST-API Guide

## Developer Guide

- Hardware Management- POST hardware/light/ ..... 35
- Hardware Management- POST hardware/videoOut ..... 38
- Hardware Management- - POST hardware/audioOut ..... 41
- Hardware Management- how to send hardware REST-APIs Without access token ..... 43
- System Management ..... 44
  - System Management- POST task/importConfiguration ..... 44
  - System Management- POST task/commitConfiguration ..... 46
  - System Management- GET task/exportConfiguration ..... 48
  - System Management- POST/ task/reboot ..... 50
  - System Management- GET task/screenshot ..... 52
  - System Management- POST app/settings/com.iadea.console/new ..... 54
  - System Management- POST app/settings/com.iadea.console/update ..... 56
  - System Management- POST security/users/admin ..... 58
- System Information ..... 60
  - System Information- GET system/playerInfo ..... 60
  - System Information- GET system/firmwareInfo ..... 62
  - System Information- GET system/modellInfo ..... 64
  - System Information- GET android.net.wifi.WifiManager/isWiFiEnabled ..... 66
  - System Information- GET android.net.wifi.WifiManager/get ConnectionInfo ..... 68
  - System Information- GET android.net.ethernet.EthernetManager/getSavedEthConfig ..... 71
  - System Information- GET android.net.ethernet.EthernetManager/getDhcpInfo ..... 73



# REST-API Guide

## Developer Guide

App Solutions .....	75
➤ App Solutions- POST /v2/app/solutions/list .....	75
➤ App Solutions- POST /v2/app/solutions/insert.....	77
➤ App Solutions- POST /v2/app/solutions/query .....	89
➤ App Solutions- POST /v2/app/solutions/update .....	92
➤ App Solutions- POST /v2/app/solutions/delete.....	96
➤ App Solutions- POST /v2/app/solutions/sync.....	98
Secure Access with HTTPS .....	101
Bypass Password During Authentication .....	101
Content Security Policy .....	103



# REST-API Guide

## Developer Guide

### Introduction

You may push SMIL scripts and media files into a player using a REST API as documented in the attachment.

IAdea firmware contains several Android architectural improvements to make IAdea devices robust to operate 24/7 with your APP. This guide provides all resources needed to get your APP up and run on IAdea devices as well as provide you some next steps options that can potentially provision your APP out of factory to save tremendous effort for large quantity deployment.

### Software requirements

- Restlet client extension tool or REST API client of your choice

### Naming

- The REST API can be accessed via HTTP prefix

```
http://(device_ip):8080/v2/
```

appended by API call. For example, for API

```
POST oauth2/token
```

the actual HTTP request should be sent to (device\_ip) at port 8080, with content similar to:

```
POST /v2/oauth2/token HTTP/1.1
Host: (device_ip)
Content-Type: application/x-www-form-urlencoded;charset=UTF-8

grant_type=password&username=...
```

Alternatively, you can also use HTTPS and send it to localhost at port 8443

```
http://localhost:8443/v2/
```

For more information on HTTPS pre-fix, please refer to the [Secure Access with HTTPS](#) section.



## REST-API Guide

### Developer Guide

#### Authentication

- Oauth2/token
  - Description: Turn on /off
  - Http request: POST
  - Input: multi-part form

Parameter	Function
grant_type	Must be the string "password"
username	Username requesting access, admin by default
password	The password configured on player. Leave this field empty by default.

➤ **Output:** JSON

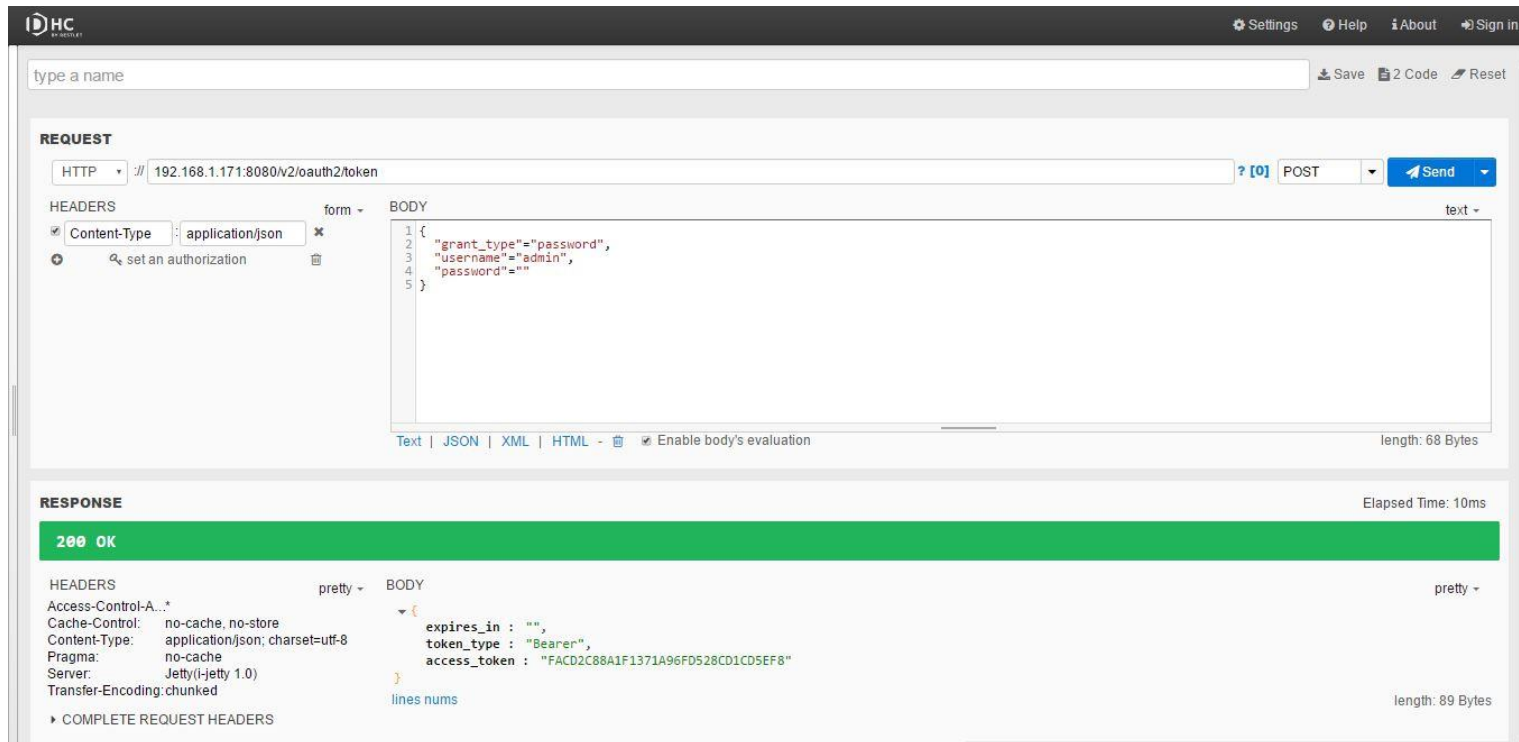
Parameter	Function
access_token	Authorization token for further API access
token_type	Always "Bearer"
expires_in	Number of seconds before access_token expires. If this field is returned as an empty string, then the access token does not expire.

**Using Access Token**

- Access token must be sent in the access\_token query parameter or an Authorization: Bearer HTTP header.
- Request with query parameter GET /v2/path/resource?access\_token=access\_token HTTP/1.1 HOST: player\_ip
- Request with HTTP header GET/v2/path/resource HTTP/1.1 HOST: player\_ip Authorization: Bearer access\_token



➤ Practical: Get the token of device



The screenshot shows a REST client interface with the following details:

- REQUEST:**
  - Method: HTTP
  - URL: // 192.168.1.171:8080/v2/oauth2/token
  - POST type: [0]
  - Headers: Content-Type: application/json
  - Body: 

```
1 {
2   "grant_type": "password",
3   "username": "admin",
4   "password": ""
5 }
```
  - Length: 68 Bytes
- RESPONSE:**
  - Status: 200 OK
  - Elapsed Time: 10ms
  - Headers: Access-Control-A..., Cache-Control: no-cache, no-store, Content-Type: application/json; charset=utf-8, Pragma: no-cache, Server: Jetty(i-jetty 1.0), Transfer-Encoding: chunked
  - Body: 

```
{
  expires_in : "",
  token_type : "Bearer",
  access_token : "FACD2C88A1F1371A96FD528CD1C05EF8"
}
```
  - Length: 89 Bytes

➤ Authentication Without Password

- You can obtain a token without including a password in the request body by using the `window.adapi.getToken()` function. This feature simplifies authentication on secure devices. For more information, refer to the [Bypass Password During Authentication](#) section.

### Media Transfer

- Media Transfer- POST files/new
  - **Description:** Create a file under *http://(device\_ip):8080/user-data* download path
  - **Input:** JSON or multi-part form

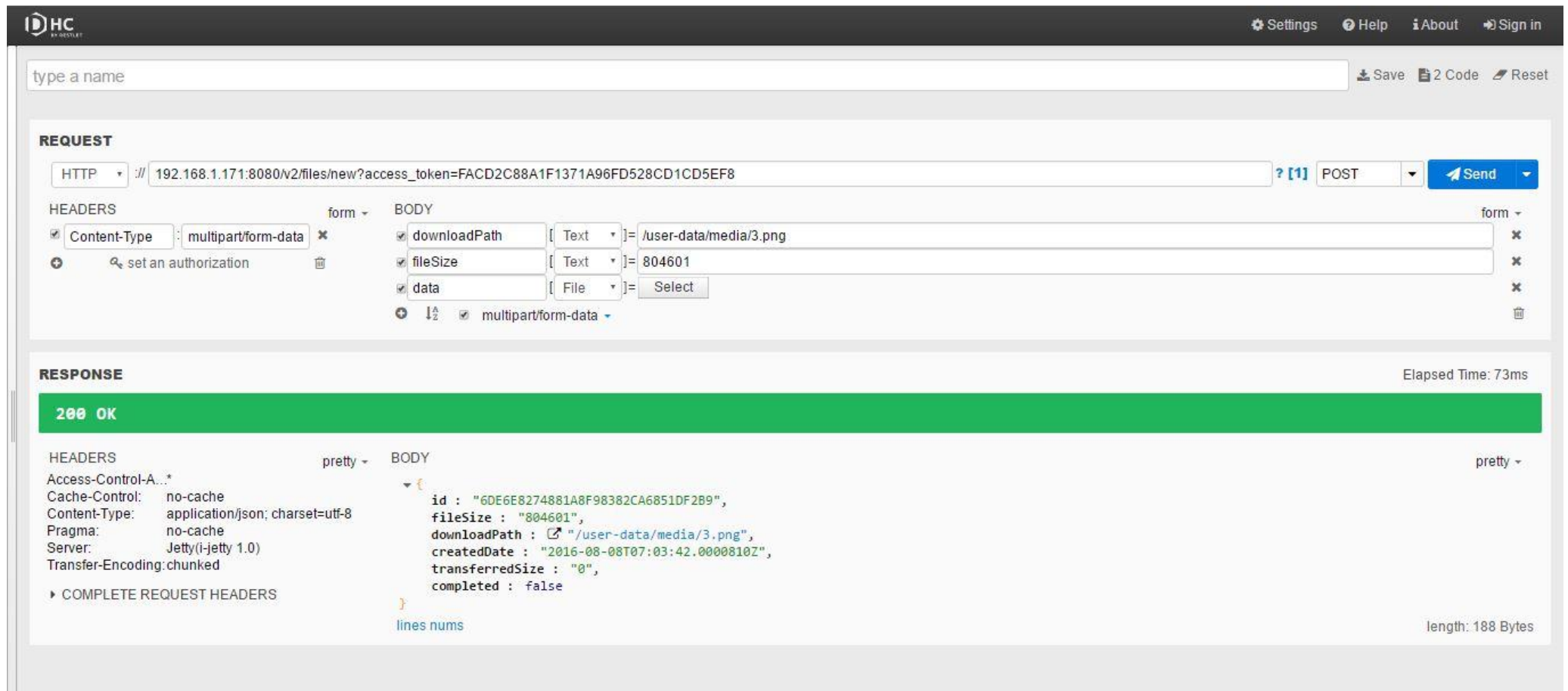
Parameter	Function
<b>downloadPath</b>	Path of file, must start with prefix /user-data/
<b>fileSize</b>	Size of file (must specify it using a positive integer larger than 0)
<b>data</b>	File encoded in multipart/form-data
<b>Etag</b>	(Optional) User-supplied unique string for version identification; default empty string
<b>mimeType</b>	(Optional) MIME type of the file; default empty string
<b>modifiedDate</b>	(Optional) ISO8601-encoded date time string indicating modified date time; default empty string

➤ **Output: JSON**

Parameter	Function
<b>id</b>	String for identifying the file in subsequent operations
<b>downloadPath</b>	Path of file to append after <a href="http://(device_ip):8080/v2">http://(device_ip):8080/v2</a>
<b>fileSize</b>	Size of file
<b>Etag</b>	Unique string for version identification
<b>mimeType</b>	MIME type of the file
<b>modifiedDate</b>	ISO8601-encoded date time string indicating modified date time; default empty string
<b>transferredSize</b>	Size uploaded into the player
<b>Completed</b>	Boolean (true or false) indicating whether file upload has completed

## Developer Guide

- **Practical:** Upload the file to device



The screenshot shows a REST client interface with the following details:

- REQUEST:**
  - Method: HTTP
  - URL: `192.168.1.171:8080/v2/files/new?access_token=FACD2C88A1F1371A96FD528CD1CD5EF8`
  - Method: POST
  - Headers:
    - Content-Type: multipart/form-data
  - Body (multipart/form-data):
    - downloadPath: /user-data/media/3.png
    - fileSize: 804601
    - data: [File]
- RESPONSE:**
  - Status: 200 OK
  - Elapsed Time: 73ms
  - Headers:
    - Access-Control-Allow-Origin: \*
    - Cache-Control: no-cache
    - Content-Type: application/json; charset=utf-8
    - Pragma: no-cache
    - Server: Jetty(i-jetty 1.0)
    - Transfer-Encoding: chunked
  - Body (JSON):

```
{
  "id": "60E6E8274881A8F98382CA6851DF2B9",
  "fileSize": "804601",
  "downloadPath": "/user-data/media/3.png",
  "createdDate": "2016-08-08T07:03:42.0000810Z",
  "transferredSize": "0",
  "completed": false
}
```
  - Length: 188 Bytes

## Developer Guide

- Media Transfer- GET files/id
  - **Description:** Get file information
  - **Input:** None
  - **Output:** JSON
  - **Practical:** Get file information by ID

**REQUEST**

HTTP // 192.168.1.171:8080/v2/files/6DE6E8274881A8F98382CA6851DF2B9?access\_token=FACD2C88A1F1371A96FD528CD1CD5EF8 ? [1] GET Send

HEADERS form BODY

Content-Type : multipart/form-data XHR does not allow an entity-body for GET request. or change a method definition in settings.

set an authorization

**RESPONSE** Elapsed Time: 11ms

200 OK

HEADERS pretty BODY pretty

Cache-Control: no-cache  
 Content-Type: application/json; charset=utf-8  
 Pragma: no-cache  
 Server: Jetty(i-jetty 1.0)  
 Transfer-Encoding: chunked

▶ COMPLETE REQUEST HEADERS

```

{
  fileSize : 804601,
  id : "6DE6E8274881A8F98382CA6851DF2B9",
  etag : "",
  downloadPath : "/user-data/media/3.png",
  createDate : "2016-08-08T07:03:42.0000810Z",
  transferredSize : 0,
  modifiedDate : "",
  mimeType : "",
  completed : false
}
      
```

length: 226 Bytes

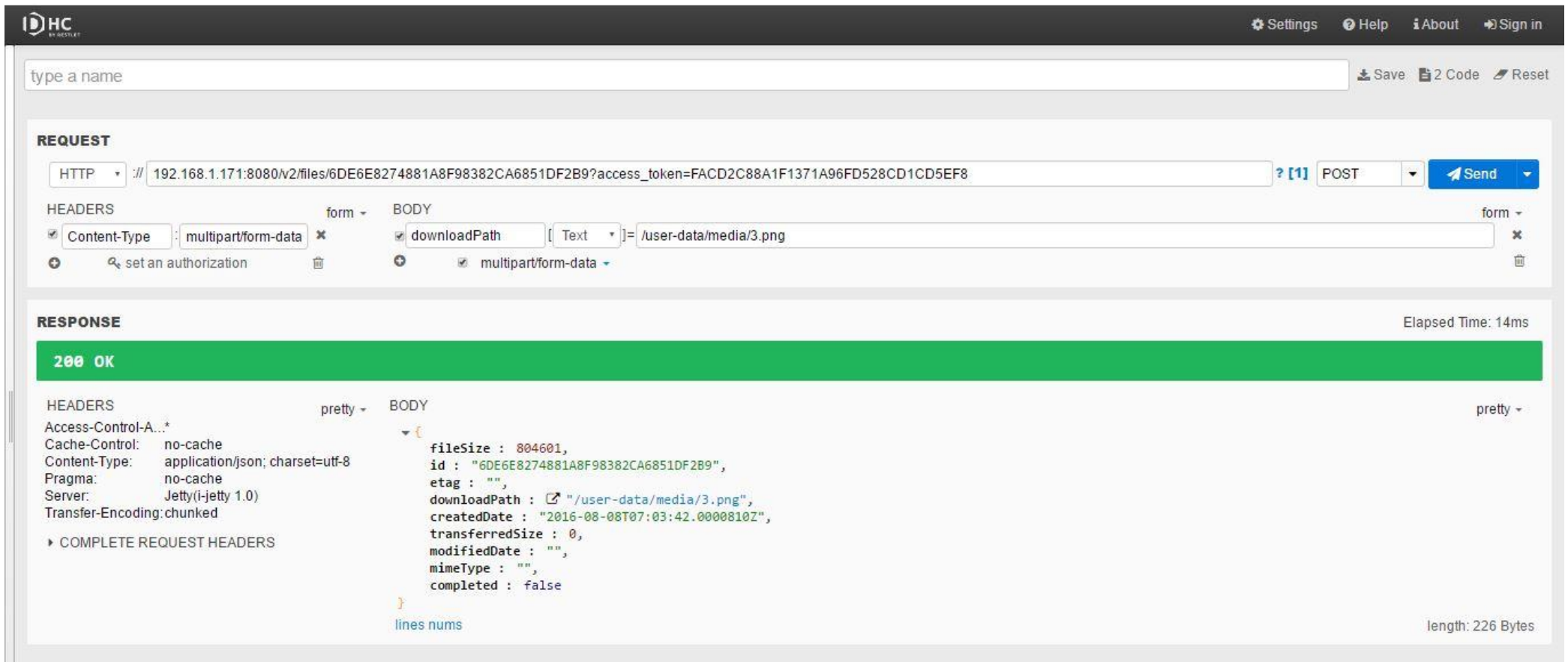
## Developer Guide

- Media Transfer- POST files/id
  - **Description:** Get file information
  - **Http request:** POST
  - **Input:** JSON or multi-part form

Parameter	Function
<b>seek</b>	(Optional) Offset from beginning of file to replace data. Default 0
<b>data</b>	(Optional) File encoded in multipart/form-data
<b>downloadPath</b>	Path of file, must start with prefix /user-data/
<b>etag</b>	(Optional) User-supplied unique string for version identification; default empty string
<b>mimeType</b>	(Optional) MIME type of the file; default empty string
<b>modifiedDate</b>	(Optional) ISO8601-encoded date time string indicating modified date time; default empty string

- **Output:** JSON

➤ **Practical:** Get file information by ID



The screenshot shows the REST Client interface with the following details:

- REQUEST**
  - Method: HTTP
  - URL: `192.168.1.171:8080/v2/files/6DE6E8274881A8F98382CA6851DF2B9?access_token=FACD2C88A1F1371A96FD528CD1CD5EF8`
  - Method: POST
  - Headers: Content-Type: multipart/form-data
  - Body: downloadPath [Text] = /user-data/media/3.png
- RESPONSE**
  - Status: 200 OK
  - Elapsed Time: 14ms
  - Headers: Access-Control-Allow-Origin: \*, Cache-Control: no-cache, Content-Type: application/json; charset=utf-8, Pragma: no-cache, Server: Jetty(i-jetty 1.0), Transfer-Encoding: chunked
  - Body (JSON):

```
{  "filePath": "/user-data/media/3.png",  "id": "6DE6E8274881A8F98382CA6851DF2B9",  "etag": "",  "downloadPath": "/user-data/media/3.png",  "createdDate": "2016-08-08T07:03:42.0000810Z",  "transferredSize": 0,  "modifiedDate": "",  "mimeType": "",  "completed": false}
```
  - length: 226 Bytes

## Developer Guide

- Media Transfer- POST files/find
  - **Description:** List files on the system
  - **Input:** JSON or multi-part form

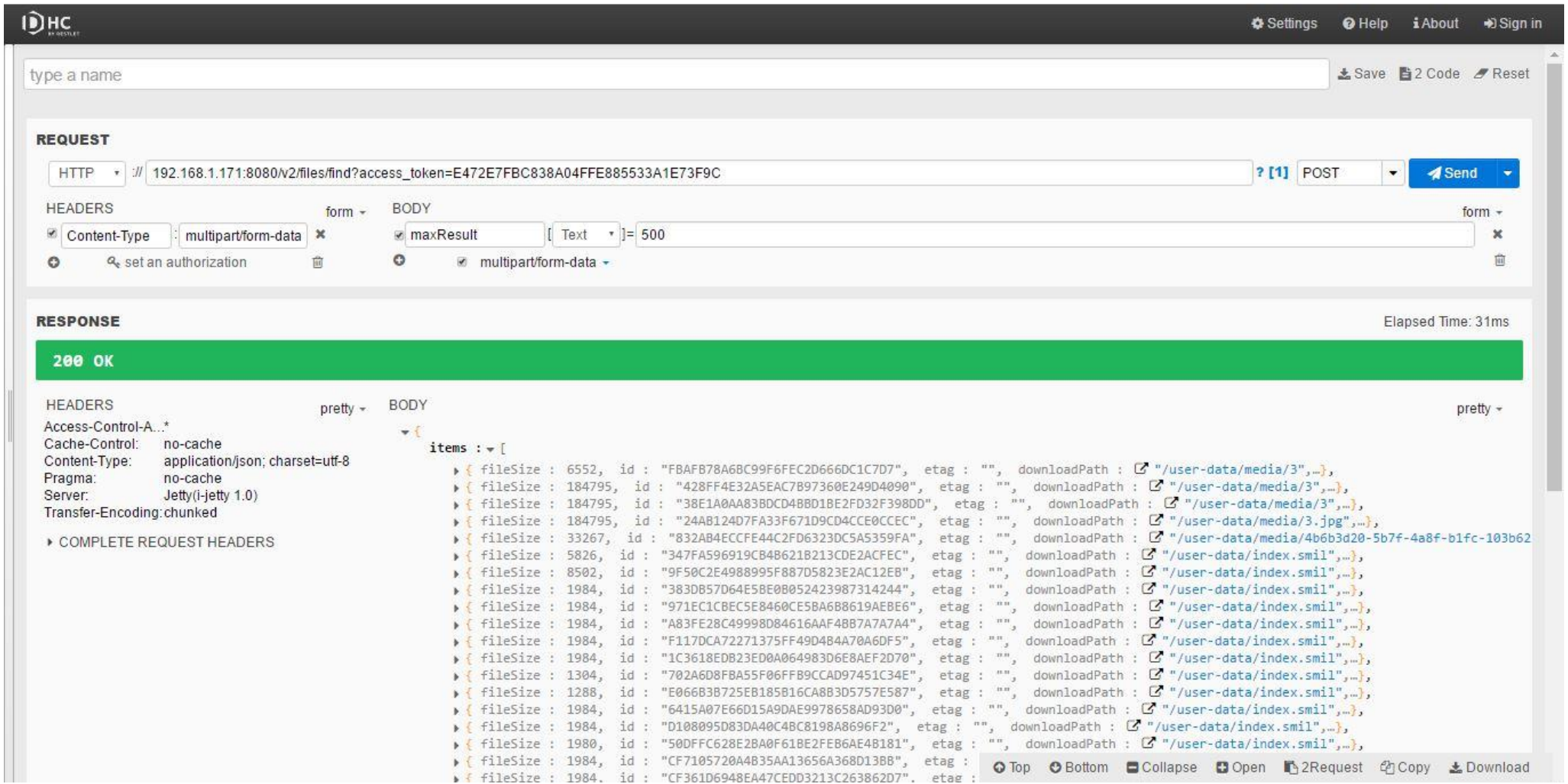
Parameter	Function
<b>maxResults</b>	(Optional) Max number of file records to return
<b>pageToken</b>	(Optional) Continuation record
<b>query</b>	(Optional) Reserved

- **Output:** JSON

Parameter	Function
<b>nextPageToken</b>	null of an identifier for report continuation
<b>items</b>	Array of <i>FileResource</i> structures



- **Practical:** List the file in device and set maximum to 500



The screenshot shows a REST client interface with the following details:

- REQUEST:**
  - Method: HTTP
  - URL: `192.168.1.171:8080/v2/files/find?access_token=E472E7FBC838A04FFE885533A1E73F9C`
  - Method: POST
  - Headers:
    - `Content-Type: multipart/form-data`
  - Body:
    - `maxResult: 500`
- RESPONSE:**
  - Status: 200 OK
  - Elapsed Time: 31ms
  - Headers:
    - `Access-Control-Allow-Origin: *`
    - `Cache-Control: no-cache`
    - `Content-Type: application/json; charset=utf-8`
    - `Pragma: no-cache`
    - `Server: Jetty(i-jetty 1.0)`
    - `Transfer-Encoding: chunked`
  - Body (JSON):

```

items : [
  {
    fileSize : 6552, id : "FBAFB78A6BC99F6FEC2D666DC1C7D7", etag : "", downloadPath : "/user-data/media/3", ...
  },
  {
    fileSize : 184795, id : "428FF4E32A5EAC7B97360E249D4090", etag : "", downloadPath : "/user-data/media/3", ...
  },
  {
    fileSize : 184795, id : "38E1A0AA83BDCD48BD1BE2FD32F398DD", etag : "", downloadPath : "/user-data/media/3", ...
  },
  {
    fileSize : 184795, id : "24AB124D7FA33F671D9CD4CCE0CCEC", etag : "", downloadPath : "/user-data/media/3.jpg", ...
  },
  {
    fileSize : 33267, id : "832AB4ECCFE44C2FD6323DC5A5359FA", etag : "", downloadPath : "/user-data/media/4b6b3d20-5b7f-4a8f-b1fc-103b62", ...
  },
  {
    fileSize : 5826, id : "347FA596919CB4B621B213CDE2ACFEC", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 8502, id : "9F50C2E4988995F887D5823E2AC12E8", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1984, id : "383DB57D64E58E08052423987314244", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1984, id : "971EC1CBEC5E8460CE5BA6B8619AEBE6", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1984, id : "A83FE28C49998D84616AAF48B7A7A7A4", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1984, id : "F117DCA72271375FF49D4B4A70A6DF5", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1984, id : "1C3618EDB23ED0A064983D6E8AEF2D70", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1304, id : "702A6D8FBA55F06FFB9CCAD97451C34E", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1288, id : "E06683B725EB185B16CA8B3D5757E587", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1984, id : "6415A07E66D15A9DAE9978658AD93D0", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1984, id : "D108095D83DA40C48C8198A8696F2", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1980, id : "50DFFC628E2BA0F61BE2FEB6AE4B181", etag : "", downloadPath : "/user-data/index.smil", ...
  },
  {
    fileSize : 1984, id : "CF7105720A4B35AA13656A368D1388", etag : ""
  },
  {
    fileSize : 1984, id : "CF361D6948EA47CEDD3213C263862D7", etag : ""
  }
]

```

## Developer Guide

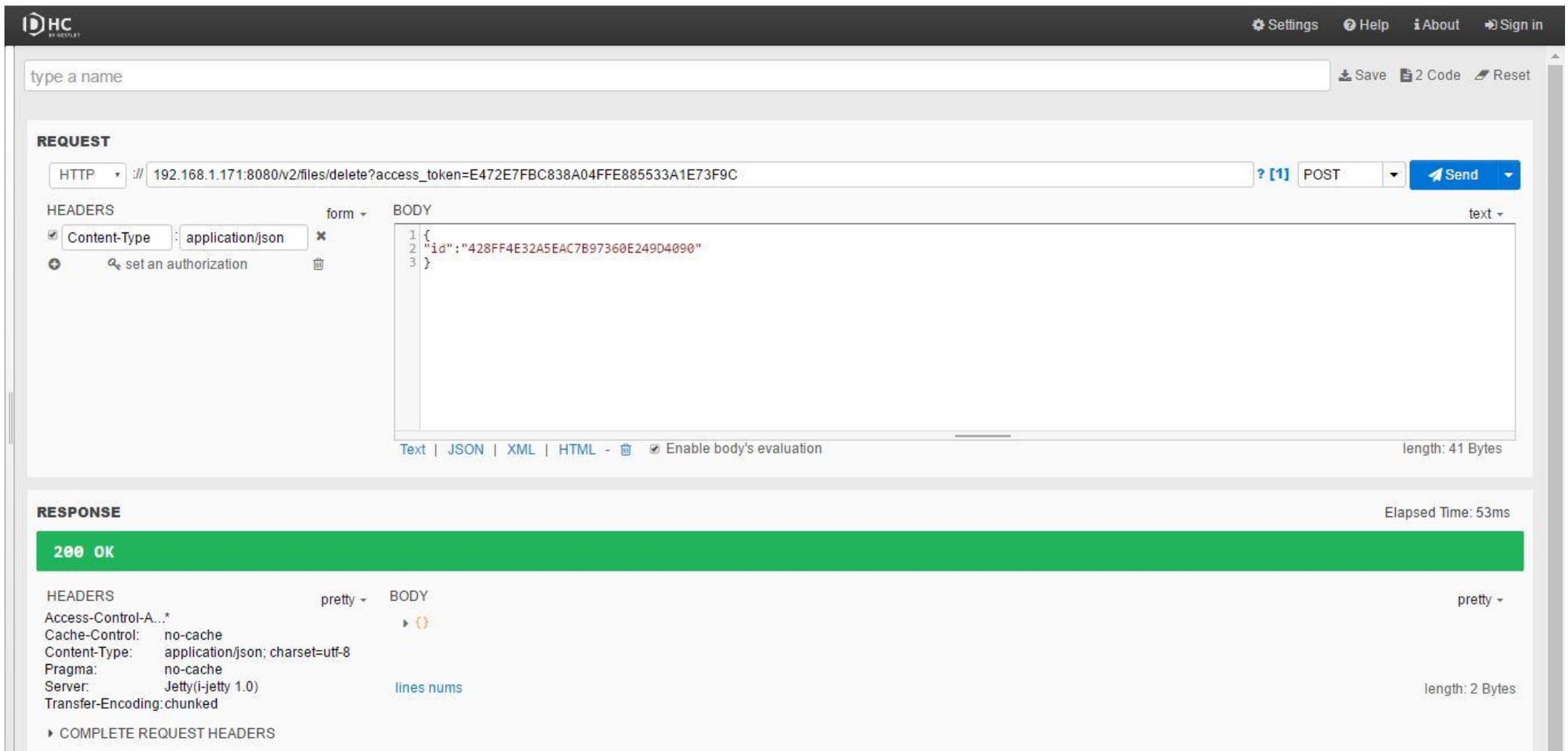
- Media Transfer- POST files/delete
  - **Description:** Delete a file
  - **Input:** JSON or multi-part form

Parameter	Function
id	ID string of file returned by files/new

- **Output:** JSON

Parameter	Function
null	null

- **Practical:** delete the file by ID



The screenshot shows a REST client interface with a dark header bar containing 'HC RESTLET' and navigation links for Settings, Help, About, and Sign in. Below the header is a search bar with the text 'type a name' and buttons for Save, 2 Code, and Reset.

The main area is divided into two sections: REQUEST and RESPONSE.

**REQUEST**

- Method: HTTP
- URL: `:// 192.168.1.171:8080/v2/files/delete?access_token=E472E7FBC838A04FFE885533A1E73F9C`
- Method dropdown: POST
- Send button: **Send**
- HEADERS: Content-Type: application/json
- BODY: 

```
1 {  
2 "id": "428FF4E32A5EAC7B97360E249D4090"  
3 }
```
- Footer: Text | JSON | XML | HTML -  Enable body's evaluation length: 41 Bytes

**RESPONSE**

- Status: **200 OK**
- Elapsed Time: 53ms
- HEADERS: Access-Control-A..., Cache-Control: no-cache, Content-Type: application/json; charset=utf-8, Pragma: no-cache, Server: Jetty(i-jetty 1.0), Transfer-Encoding: chunked
- BODY: 

```
▶ { }
```
- Footer:  COMPLETE REQUEST HEADERS length: 2 Bytes

### Playback

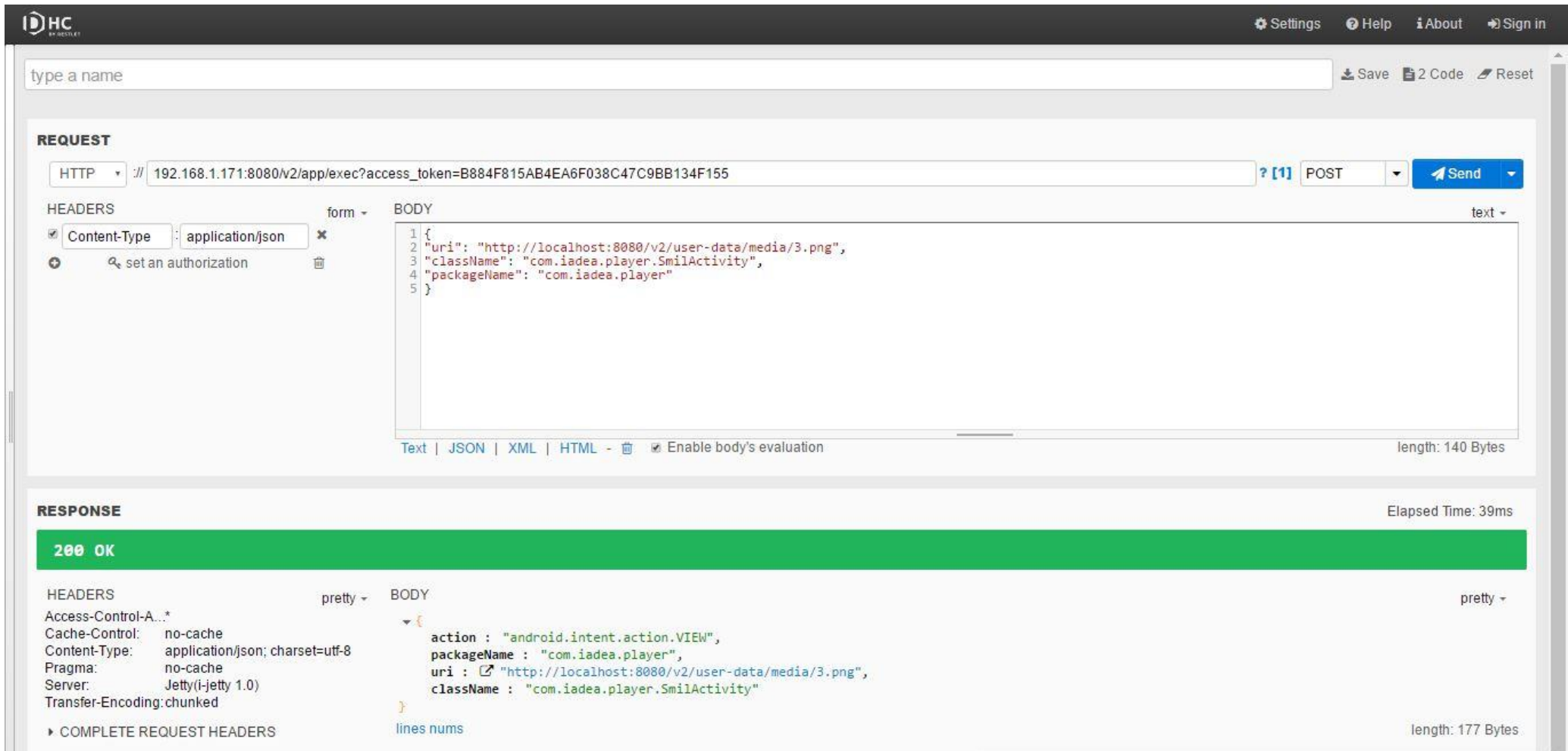
- Playback- POST app/exec
  - **Description:** Play Content once
  - **Input:** JSON

Parameter	Function
<b>uri</b>	Location of content. May be http://localhost:8080/v2/user-data/...
<b>packageName</b>	Android package to launch
<b>className</b>	Android class name
<b>action</b>	Android action
<b>type</b>	(Optional) Android intent type
<b>extras</b>	(Optional) Array of Android intent extra parameters

➤ **Output: JSON**

Parameter	Function
<b>Uri</b>	Location of content. May be http://localhost:8080/user-data/...
<b>packageName</b>	(Optional) Android package to launch
<b>className</b>	(Optional) Android class name
<b>Action</b>	(Optional) Android action
<b>Type</b>	(Optional) Android intent type

- **Practical:** Play the content set in URL



The screenshot shows a REST client interface with the following details:

**REQUEST**

- Method: HTTP
- URL: `:// 192.168.1.171:8080/v2/app/exec?access_token=B884F815AB4EA6F038C47C9BB134F155`
- POST
- Send button
- Headers:
  - Content-Type: `application/json`
  - set an authorization
- Body:
 

```
1 {
2   "uri": "http://localhost:8080/v2/user-data/media/3.png",
3   "className": "com.iadea.player.SmilActivity",
4   "packageName": "com.iadea.player"
5 }
```
- length: 140 Bytes

**RESPONSE**

- Status: **200 OK**
- Elapsed Time: 39ms
- Headers:
  - Access-Control-A...
  - Cache-Control: no-cache
  - Content-Type: `application/json; charset=utf-8`
  - Pragma: no-cache
  - Server: Jetty(i-jetty 1.0)
  - Transfer-Encoding: chunked
- Body:
 

```
{
  "action": "android.intent.action.VIEW",
  "packageName": "com.iadea.player",
  "uri": "http://localhost:8080/v2/user-data/media/3.png",
  "className": "com.iadea.player.SmilActivity"
}
```
- length: 177 Bytes

## Developer Guide

- ▶ Playback- POST app/start
  - ▶ **Description:** Play Content once
  - ▶ **Input:** JSON

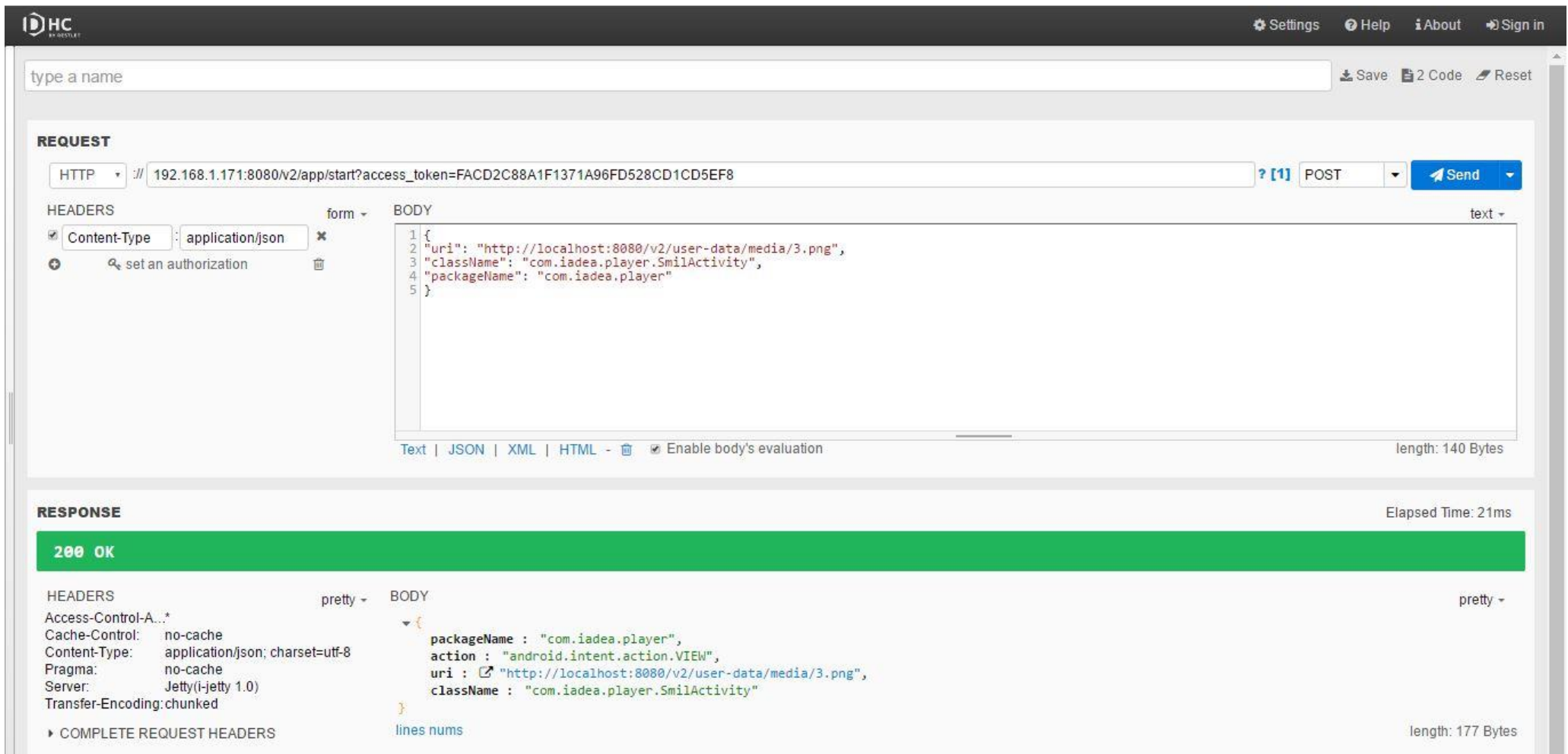
Parameter	Function
<b>uri</b>	Location of content. May be http://localhost:8080/v2/user-data/...
<b>packageName</b>	Android package to launch
<b>className</b>	Android class name
<b>action</b>	(Optional) Android action
<b>type</b>	(Optional) Android intent type
<b>extras</b>	(Optional) Array of Android intent extra parameters

➤ **Output: JSON**

Parameter	Function
<b>uri</b>	Location of content. May be <code>http://localhost:8080/v2/user-data/...</code>
<b>packageName</b>	Android package to launch
<b>className</b>	Android class name
<b>action</b>	(Optional) Android action
<b>type</b>	(Optional) Android intent type



### ➤ Practical: Set the default content



The screenshot displays the DHC REST client interface. At the top, there is a search bar with the text "type a name" and navigation links for "Settings", "Help", "About", and "Sign in". Below the search bar, the "REQUEST" section is active. The URL is set to "http://192.168.1.171:8080/v2/app/start?access\_token=FACD2C88A1F1371A96FD528CD1CD5EF8" and the method is "POST". The "Content-Type" header is set to "application/json". The request body is a JSON object: 

```
{
  "uri": "http://localhost:8080/v2/user-data/media/3.png",
  "className": "com.iadea.player.SmilActivity",
  "packageName": "com.iadea.player"
}
```

. The response section shows a "200 OK" status with a response time of 21ms. The response headers include "Access-Control-Allow-Origin", "Cache-Control: no-cache", "Content-Type: application/json; charset=utf-8", "Pragma: no-cache", "Server: Jetty(i-jetty 1.0)", and "Transfer-Encoding: chunked". The response body is a JSON object: 

```
{
  "packageName": "com.iadea.player",
  "action": "android.intent.action.VIEW",
  "uri": "http://localhost:8080/v2/user-data/media/3.png",
  "className": "com.iadea.player.SmilActivity"
}
```

## Developer Guide

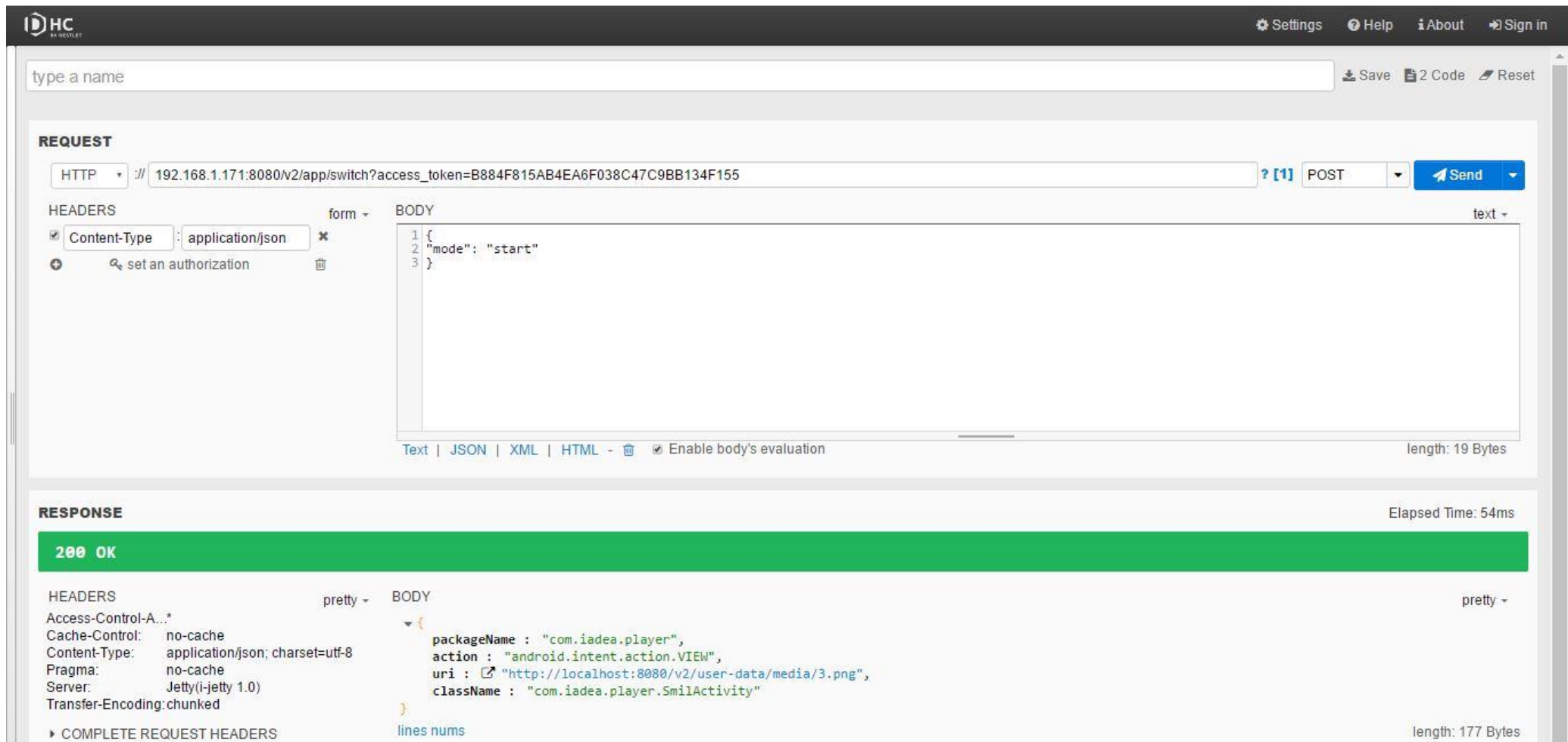
- Playback-POST app/switch
  - **Description:** Switch to play default Content
  - **Input:** JSON

Parameter	Function
<b>mode</b>	Set to "start" to play default content
<b>ifModifiedSince</b>	(Optional) Switch to content if it is modified after this ISO8601 date/time

- **Output:** JSON

Parameter	Function
<b>uri</b>	Location of content. May be http://localhost:8080/v2/user-data/...
<b>packageName</b>	(Optional) Android package to launch
<b>className</b>	(Optional) Android class name
<b>action</b>	(Optional) Android action
<b>type</b>	(Optional) Android intent type

- **Practical:** Switch to play the default content



The screenshot shows the IDE's REST client interface. At the top, there's a search bar with the text "type a name" and buttons for "Save", "2 Code", and "Reset". Below this is the "REQUEST" section. The method is set to "POST" and the URL is "http://192.168.1.171:8080/v2/app/switch?access\_token=B884F815AB4EA6F038C47C9BB134F155". The "Content-Type" header is set to "application/json". The request body is a JSON object: 

```
{ "mode": "start" }
```

. Below the request section, the "RESPONSE" section shows a "200 OK" status. The response headers include "Access-Control-Allow-Origin", "Cache-Control: no-cache", "Content-Type: application/json; charset=utf-8", "Pragma: no-cache", "Server: Jetty(i-jetty 1.0)", and "Transfer-Encoding: chunked". The response body is a JSON object: 

```
{ "packageName": "com.iadea.player", "action": "android.intent.action.VIEW", "uri": "http://localhost:8080/v2/user-data/media/3.png", "className": "com.iadea.player.SmilActivity" }
```

. The interface also shows options for "Enable body's evaluation" and "length: 19 Bytes" for the request and "length: 177 Bytes" for the response.



## REST-API Guide

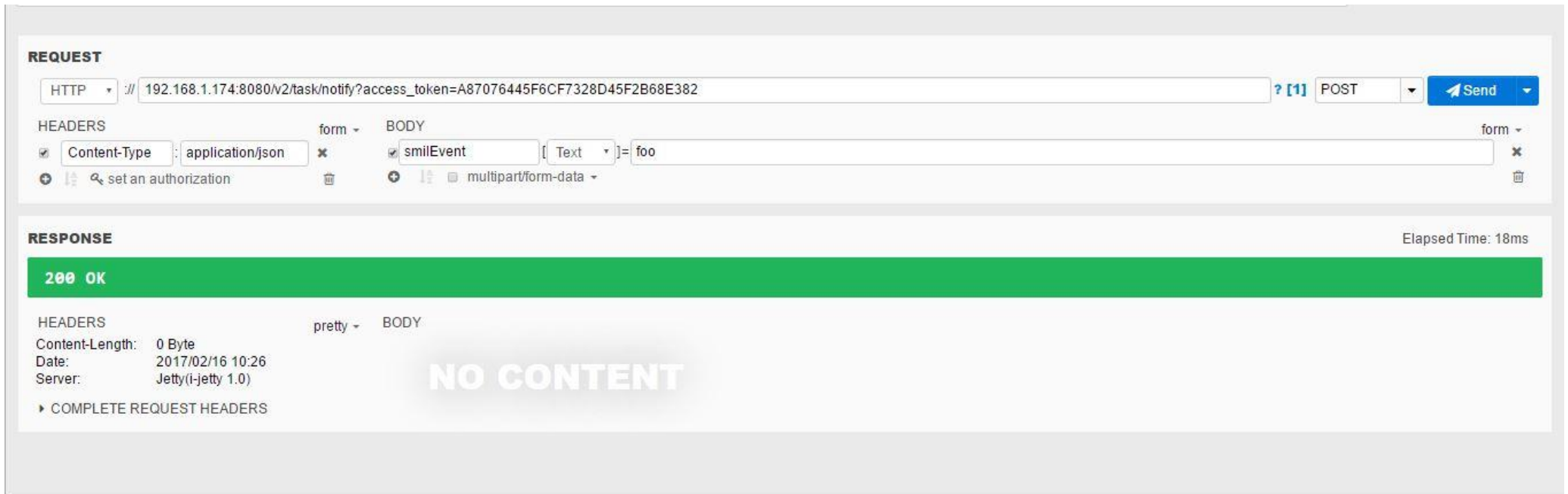
### Developer Guide

- ▶ Playback- POST v2/task/notify
  - ▶ **Description:** Network event Trigger
  - ▶ **Input:** JSON

Parameter	Function
smilEvent	foo

- ▶ **Output:** none
- ▶ **Note:** if user needs more than one trigger, please use different name of function e.g. foo1,foo2...etc.

### ➤ Practical:



The screenshot displays a REST client interface with the following details:

- REQUEST**
  - Method: HTTP
  - URL: `192.168.1.174:8080/v2/task/notify?access_token=A87076445F6CF7328D45F2B68E382`
  - Method: POST
  - Buttons: ? [1], Send
  - HEADERS**
    - Content-Type: application/json
    - set an authorization
  - BODY**
    - Form type: form
    - Field: smilEvent (Text) = foo
    - Form type: multipart/form-data
- RESPONSE**
  - Status: 200 OK
  - Elapsed Time: 18ms
  - HEADERS**
    - Content-Length: 0 Byte
    - Date: 2017/02/16 10:26
    - Server: Jetty(i-jetty 1.0)
  - Body: NO CONTENT
  - Link: ▶ COMPLETE REQUEST HEADERS

### Hardware Management

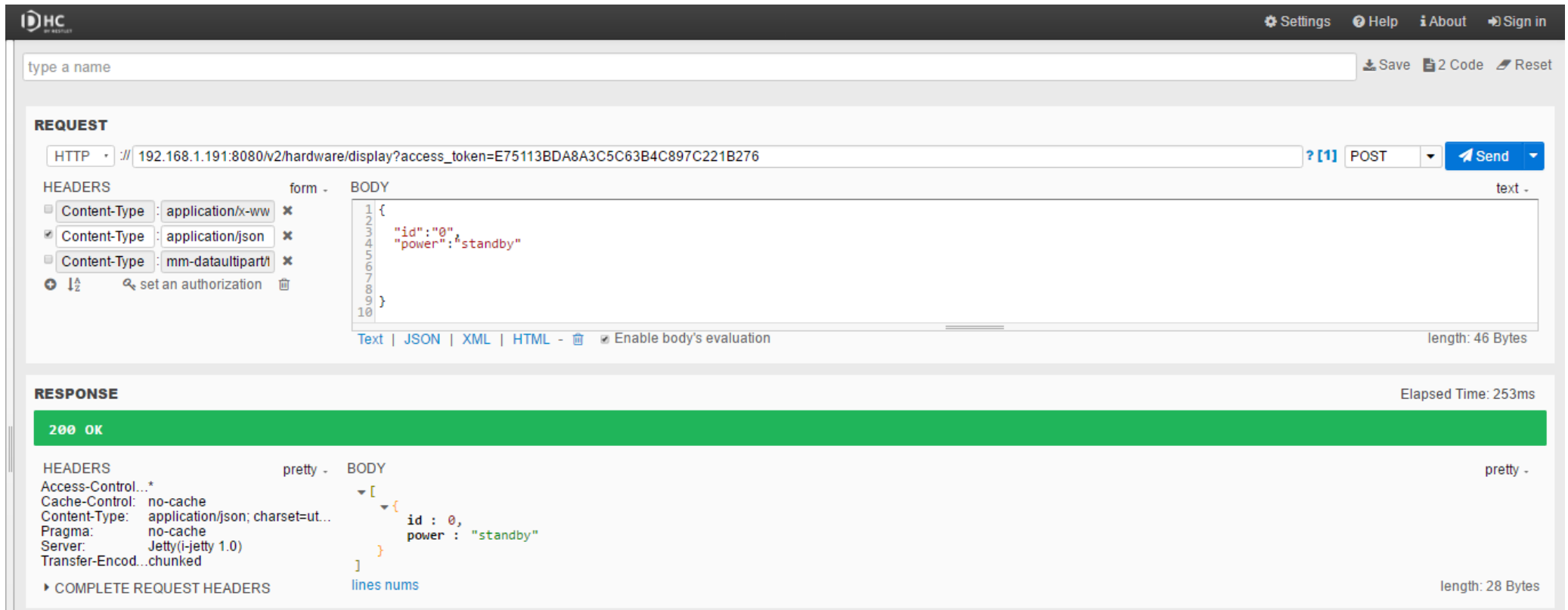
- Hardware Management- POST hardware/display
  - **Description:** Turn on/off display
  - **Input:** JSON or multi-part form

Parameters	Function
<b>Id</b>	the display id, always set to "0" (refer to number zero)
<b>Power</b>	Set to "on" to turn on display; Set to "standby" to turn off display

- **Output:** JSON

Parameters	Function
<b>Id</b>	the display id.
<b>Power</b>	return current display setting.

➤ **Practical:** Turn off display



The screenshot shows the REST Client interface with a POST request to `http://192.168.1.191:8080/v2/hardware/display?access_token=E75113BDA8A3C5C63B4C897C221B276`. The request body is a JSON object: `{ "id": "0", "power": "standby" }`. The response is a `200 OK` with headers including `Access-Control-Allow-Origin: *`, `Cache-Control: no-cache`, `Content-Type: application/json; charset=utf-8`, `Pragma: no-cache`, `Server: Jetty(i-jetty 1.0)`, and `Transfer-Encoding: chunked`. The response body is the same JSON object as the request body.

**REQUEST**

HTTP :// 192.168.1.191:8080/v2/hardware/display?access\_token=E75113BDA8A3C5C63B4C897C221B276 [1] POST Send

**HEADERS** form - BODY text -

- Content-Type : application/x-www-form-urlencoded
- Content-Type : application/json
- Content-Type : multipart/form-data

```
{
  "id": "0",
  "power": "standby"
}
```

length: 46 Bytes

**RESPONSE** Elapsed Time: 253ms

**200 OK**

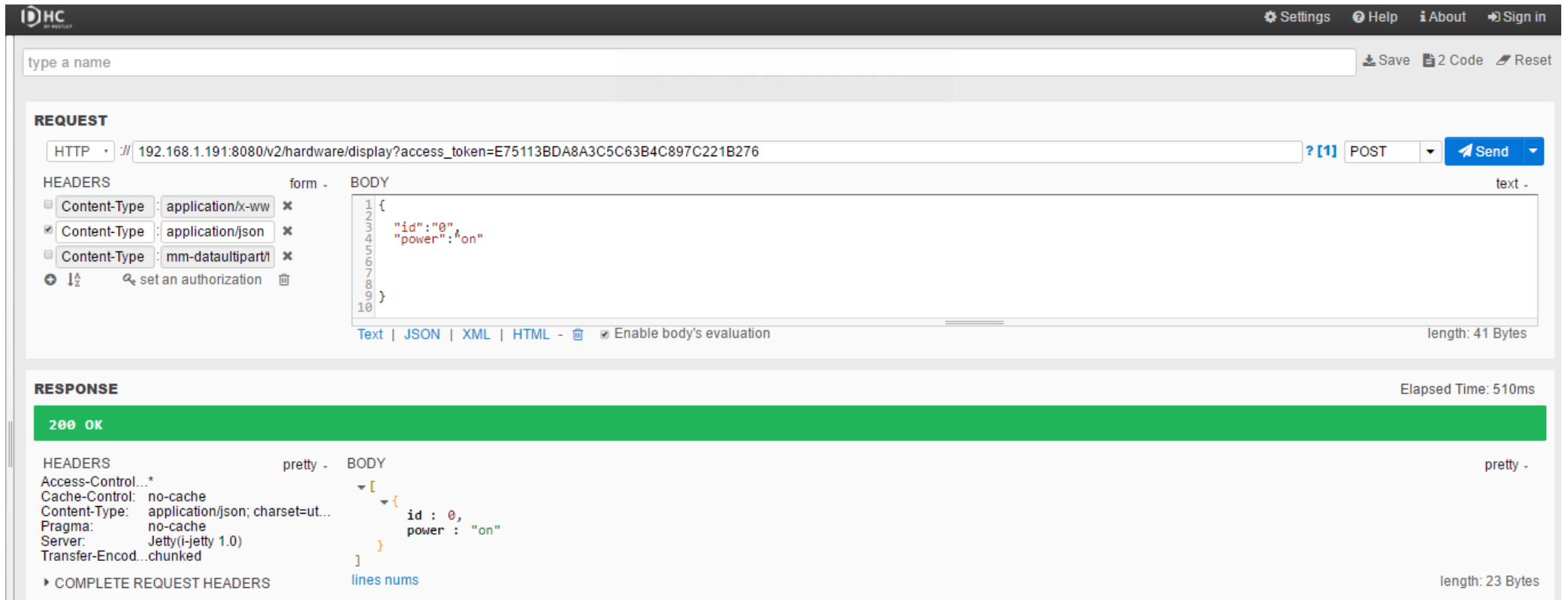
**HEADERS** pretty - BODY pretty -

- Access-Control-Allow-Origin: \*
- Cache-Control: no-cache
- Content-Type: application/json; charset=utf-8
- Pragma: no-cache
- Server: Jetty(i-jetty 1.0)
- Transfer-Encoding: chunked

```
[
  {
    id : 0,
    power : "standby"
  }
]
```

length: 28 Bytes

➤ **Practical:** Turn on display



The screenshot displays the REST Client interface with the following details:

- Request:**
  - Method: HTTP
  - URL: `://192.168.1.191:8080/v2/hardware/display?access_token=E75113BDA8A3C5C63B4C897C221B276`
  - Method: POST
  - Body: 

```
{
  "id": "0",
  "power": "on"
}
```
  - Headers: Content-Type: application/x-www-form-urlencoded, application/json, mm-datamultipart/form-data
  - Length: 41 Bytes
- Response:**
  - Status: 200 OK
  - Elapsed Time: 510ms
  - Headers: Access-Control-Allow-Origin: \*, Cache-Control: no-cache, Content-Type: application/json; charset=utf-8, Pragma: no-cache, Server: Jetty(i-jetty 1.0), Transfer-Encoding: chunked
  - Body: 

```
[
  {
    "id": "0",
    "power": "on"
  }
]
```
  - Length: 23 Bytes



## Developer Guide

- [Hardware Management- POST hardware/rs232](#)
  - **Description:** Send serial command through RS232 port.
  - **Input:** JSON or multi-part form

Parameters	Function
<b>baudrate</b>	the external RS232 device baud rate setting, e.g. 9600,38400,115200.
<b>repeatCount</b>	set default(global) repeat count, defines how many times user like to repeat sending the RS232 command
<b>repeatInterval</b>	set default(global) repeat interval while repeat sending RS232 commands if configured
<b>controls</b>	<p>defines individual control id and it corresponds RS232 command, repeatCount and repeatInterval. When repeatCount and repeatInterval are not defined, default setting will be applied. E.g., 'controls': the player <b>only receives hexadecimal code with the end character.</b></p> <pre>[{'id':'display.power.on','command':'38 39 39 73 21 30 30 31','repeatCount':'2','repeatInterval':'200'}, {'id':'display.power.off','command':'38 39 39 73 21 30 30 31','repeatCount':'1','repeatInterval':'500'}]</pre>



## REST-API Guide

### Developer Guide

➤ **Output: JSON**

Parameter	Function
<b>controls</b>	returns the control id configured
<b>repeatInterval</b>	return the repeat interval configured
<b>repeatCount</b>	return the repeat count configured
<b>baudrate</b>	return the baud rate configured

- **Practical:** send serial command through RS232 port

The screenshot shows a REST client interface with the following details:

- REQUEST:**
  - Method: **POST**
  - URL: `http://192.168.1.113:8080/v2/hardware/rs232?access_token=EB8850CBCCF7583EDB05BA4FF52FD2D`
  - Header: `Content-Type: application/json`
  - Body (JSON):

```
1 {
2   "baudrate": "9600",
3   "repeatCount": "2",
4   "repeatInterval": "500",
5   "controls": [
6     {
7       "id": "display.power.on", "command": "38 39 39 73 21 30 30 31 0D", "repeatCount": "2", "repeatInterval": "200"},
8     {
9       "id": "display.power.off", "command": "38 39 39 73 21 30 30 30 0D", "repeatCount": "1", "repeatInterval": "500"}
10  ]
11 }
```
- RESPONSE:**
  - Status: **200 OK** (highlighted in green)
  - Elapsed Time: 30ms
  - Header: `Content-Type: application/json; charset=utf-8`
  - Body (JSON):

```
{
  "controls": [
    {
      "id": "display.power.on",
      "command": "38 39 39 73 21 30 30 31 0D",
      "repeatInterval": "200",
      "repeatCount": "2"
    },
    {
      "id": "display.power.off",
      "command": "38 39 39 73 21 30 30 30 0D",
      "repeatInterval": "500",
      "repeatCount": "1"
    }
  ]
}
```

## Developer Guide

- Hardware Management- POST hardware/light/
  - **Description:** Control GPO through GPIO board and light bar on digital signboard
  - **Input:** JSON or multi-part form

Parameter	Function
<b>Name</b>	name for touch buttons: touchButtonN where N = A, B, C, X, Y, Z, L1, L2, R1, R2, ... name for general purpose output: GPOn where n = 0, 1, 2, 3, ... name for light-bar on frame: frame, frameLeft, frameTop, frameRight, frameBottom name for status light on front: front
<b>Brightness</b>	brightness level: 0...1, set to "0" to turn off light (GPO); set to "1" to turn on light (GPO).
<b>Color</b>	color string is common HTML color value, 6-digit hex color code: "#RRGGBB"; To turn off LED, set color code as "#000000". If the digital signboard supports only 12-bit color input, such as IAdea digital signboard XDS-1078, signboard will return the supported color code. For example, if set "#FF0000", device will return "#F00000".



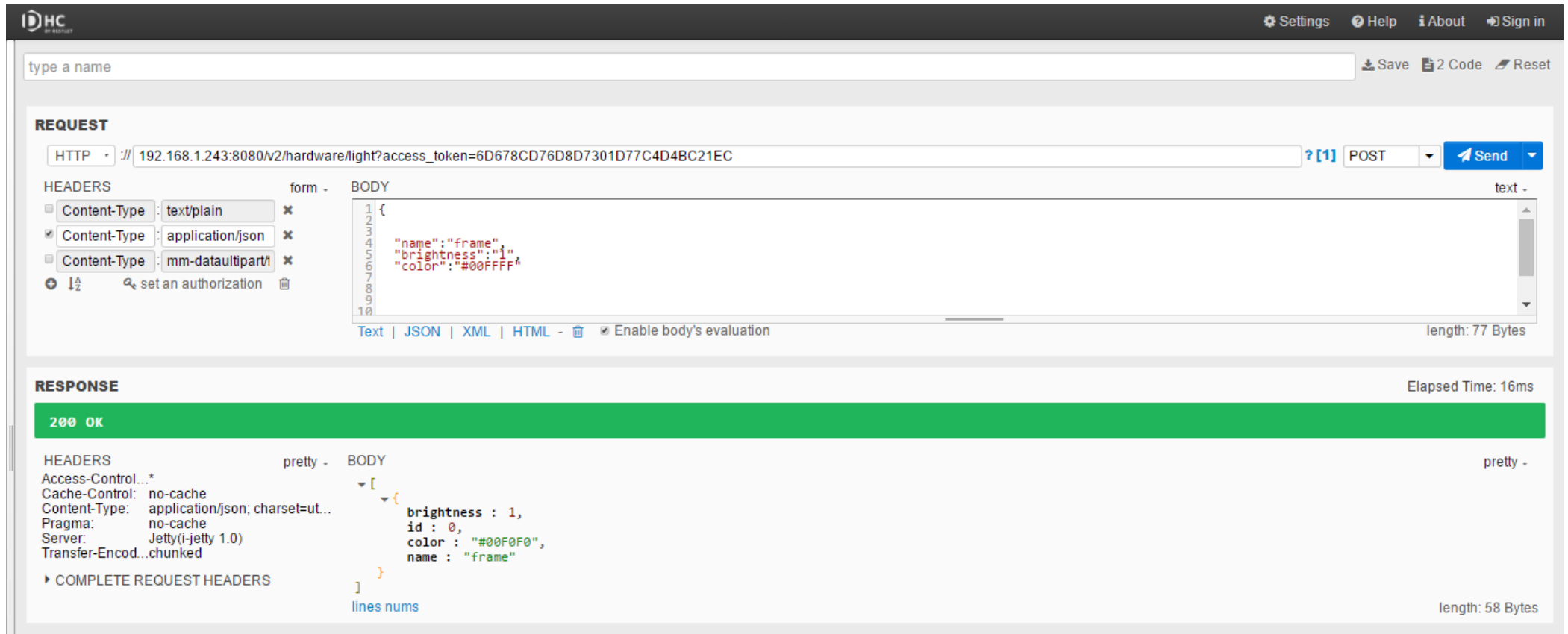
# REST-API Guide

## Developer Guide

➤ **Output: JSON**

Paremeter	Function
Id	return interface id
Name	return interface name
Brightness	return brightness level
Color	return color value if available

- **Practical:** Turn on light bar on digital signboard.



The screenshot shows a REST client interface with the following details:

- Request:** Method: POST, URL: `http://192.168.1.243:8080/v2/hardware/light?access_token=6D678CD76D8D7301D77C4D4BC21EC`. Headers include `Content-Type: text/plain`, `Content-Type: application/json`, and `Content-Type: mm-datamultipart`. The body is a JSON object: `{ "name": "frame", "brightness": 1, "color": "#00FFFF" }`. Length: 77 Bytes.
- Response:** Status: 200 OK, Elapsed Time: 16ms. Headers include `Access-Control-Allow-Origin: *`, `Cache-Control: no-cache`, `Content-Type: application/json; charset=utf-8`, `Pragma: no-cache`, `Server: Jetty(i-jetty 1.0)`, and `Transfer-Encoding: chunked`. The body is a JSON array: `[ { "brightness": 1, "id": 0, "color": "#00F0F0", "name": "frame" } ]`. Length: 58 Bytes.

## Developer Guide

- Hardware Management- POST hardware/videoOut
  - **Description:** Control HDMI-out settings on device
  - **Input:** JSON or multi-part form

Parameter	Function
<b>id</b>	the display id, always set to "0" (refer to number zero)
<b>rotation</b>	Rotation-string: "0 90 180 270" (rotate specified degrees clockwise) Rotation-string: "auto" (rotate according to accelerometer)
<b>format</b>	The resolution setting on device MBR-1100, XMP-6250/6400, XDS-1078 support: AUTO/CEA_480P60/CEA_576P50/CEA_720P60/CEA_720P50/CEA_1080P60/CEA_1080P50 XMP-7300 support: AUTO/CEA_480P60/CEA_576P50/CEA_720P60/CEA_720P50/CEA_1080P60/CEA_1080P50/CEA_2160P60/CEA_2160P50/CEA_2160P30
<b>margin</b>	The space around content to compensate display's over scanning Value: "{number}%" or "{number}px"
<b>cecEnabled</b>	True: Enable CEC, False: Disable CEC
<b>hdcpEnabled</b>	True: Enable HDCP, False: Disable HDCP

➤ **Output: JSON**

<b>Parameter</b>	<b>Function</b>
<b>Id</b>	return display id
<b>Format</b>	return the solution settings
<b>Margin</b>	return the space around content
<b>Rotation</b>	return screen orientation setting
<b>cecEnabled</b>	return CEC status
<b>hdcpEnabled</b>	return HDCP status



## Developer Guide

### ➤ Practical: The HDMI-out settings on device

METHOD: POST | SCHEME // HOST [:" PORT ] [ PATH [ "?" QUERY ] ] | http://192.168.1.209:8080/v2/hardware/videoOut | length: 46 bytes | Send

QUERY PARAMETERS

HEADERS

- Content-Type: application/json
- Authorization: Bearer 1B31CDBACFD5DFB22A8D91511FF3

ADD header | ADD authorization

BODY

```
1 { "id": 0, "format": "CEA_720P50", "margin": "10 10 10 10", "rotation": "0",
2   "cecEnabled": false }
```

200 OK

HEADERS

Access-Control-Allow-Origin: \*

Cache-Control: no-cache

Pragma: no-cache

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Server: Jetty(i-jetty 1.0)

▶ COMPLETE REQUEST HEADERS

BODY

```
[
  {
    id: 0,
    margin: "10px 10px 10px 10px",
    format: "CEA_720p50",
    rotation: "0",
    hdcpEnabled: "true",
    cecEnabled: "false"
  }
]
```

## Developer Guide

- Hardware Management- - POST hardware/audioOut
  - **Description:** The sound level on device
  - **Input:** JSON or multi-part form

Parameter	Function
id	the display id, always set to "0" (refer to number zero)
masterSound Level	soundLevel: "mute" or "0.0..100.0%" or 0..1.0 or -number..0dB Note: All numbers are in logarithmic scale, i.e. no logarithmic function should be applied to these numbers

- **Output:** JSON

Paremeter	Function
Id	return display id
soundLevel	return value of sound level

## Developer Guide

- **Practical:** Set the sound level on device

The screenshot displays a REST client interface with the following details:

- Request:**
  - Method: POST
  - URL: `http://192.168.1.253:8080/v2/hardware/audioOut`
  - Length: 46 bytes
  - Headers:
    - Content-Type: `application/json`
    - Authorization: `Bearer 82269A2E2C74B20C7ADC720EB3E1f`
  - Body: `1 { "id": 0, "masterSoundLevel": 80%}`
- Response:**
  - Status: 200 OK
  - Cache Detected - Elapsed Time: 25ms
  - Headers:
    - Access-Control-Allow-Origins: \*
    - Cache-Control: no-cache
    - Pragma: no-cache
    - Content-Type: application/json; charset=utf-8
  - Body: `[{"masterSoundLevel": "0.25118864315"}]`

## Developer Guide

- Hardware Management- how to send hardware REST-APIs Without access token
  - For the REST APIs under hardware management, you can use them without getting the token in advance. This method is limited to use in the **HTML or Android app** which will run on IAdea device.
  - The examples to call the REST-API of light bar on XDS-1078 without access token.
    1. Leave the token to be blank. [Sample Code Download](#)  
**POST http://localhost:8080/v2/hardware/light?access\_token=**
    2. Remove '?access\_token =' from command. [Sample Code Download](#)  
**POST http://localhost:8080/v2/ hardware/light**
  - Supported model and firmware version:
    - MBR-1100 & XMP-6200 (TBC)
    - XMP6250 & XMP-6400 (1.2.93.621 or later)
    - XDS-1078 (1.2.91.624 or later)

## System Management

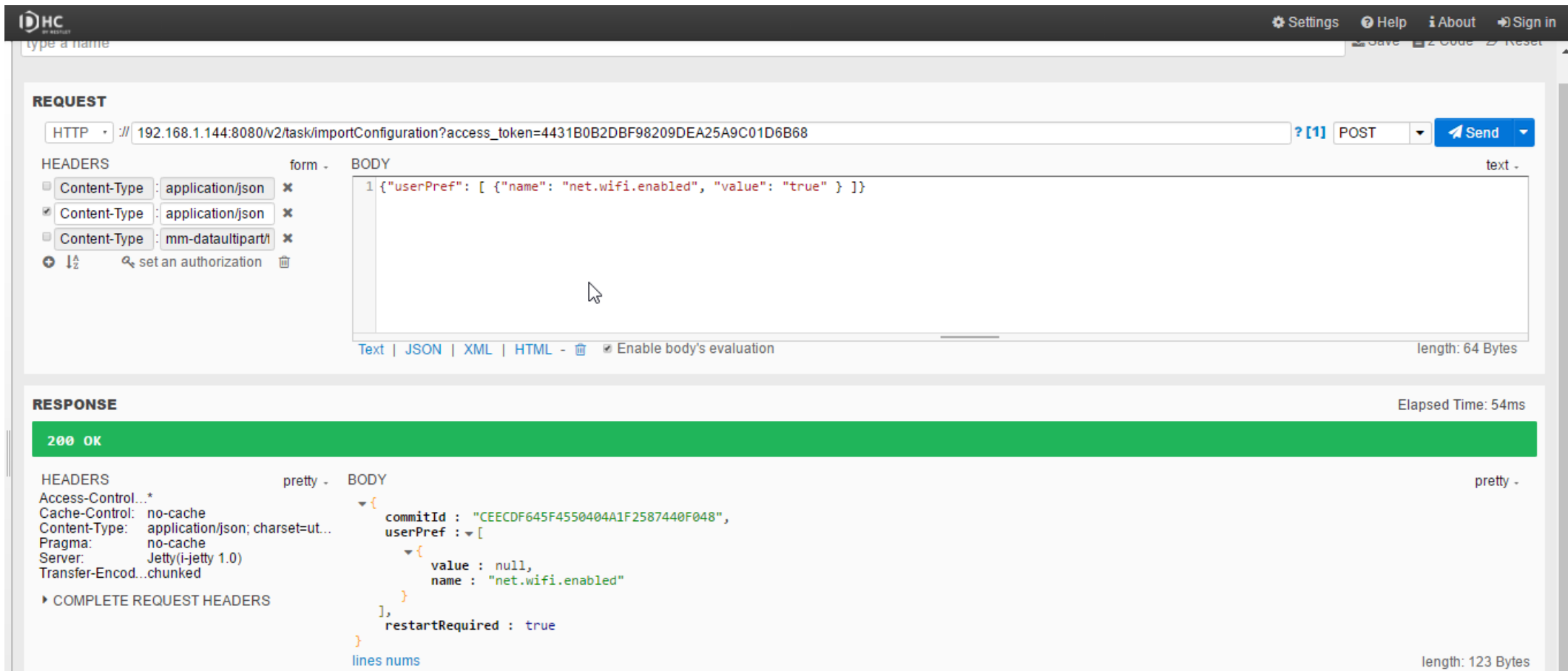
- System Management- POST task/importConfiguration
  - **Description:** Import new configuration to player
  - **Input:** JSON or multi-part form

Parameter	Function
<b>userPref</b>	new configuration object ex. {"userPref": [ {"name": "string", "value": "string" }, {...} ] } Parameter refer <a href="#">Device configuration</a>

- **Output:** JSON

Parameter	Function
<b>commitId</b>	return the ID to be committed when commitConfiguration
<b>userPref</b>	return newly imported configuration
<b>restartRequired</b>	true/false, if restart is required for changes to take effect, restartRequired is true

- **Practical:** Import new configuration to player



The screenshot shows a REST client interface with the following details:

- REQUEST:**
  - Method: **POST**
  - URL: `http://192.168.1.144:8080/v2/task/importConfiguration?access_token=4431B0B2DBF98209DEA25A9C01D6B68`
  - Headers:
    - Content-Type: application/json
    - Content-Type: application/json
    - Content-Type: mm-dataulpart/
  - Body: `1 [{"userPref": [{"name": "net.wifi.enabled", "value": "true"}]}]`
- RESPONSE:**
  - Status: **200 OK**
  - Elapsed Time: 54ms
  - Headers:
    - Access-Control-...\*
    - Cache-Control: no-cache
    - Content-Type: application/json; charset=ut...
    - Pragma: no-cache
    - Server: Jetty(i-jetty 1.0)
    - Transfer-Encod...: chunked
  - Body:

```

{
  commitId : "CEECDF645F4550404A1F2587440F048",
  userPref : [
    {
      value : null,
      name : "net.wifi.enabled"
    }
  ],
  restartRequired : true
}

```

## Developer Guide

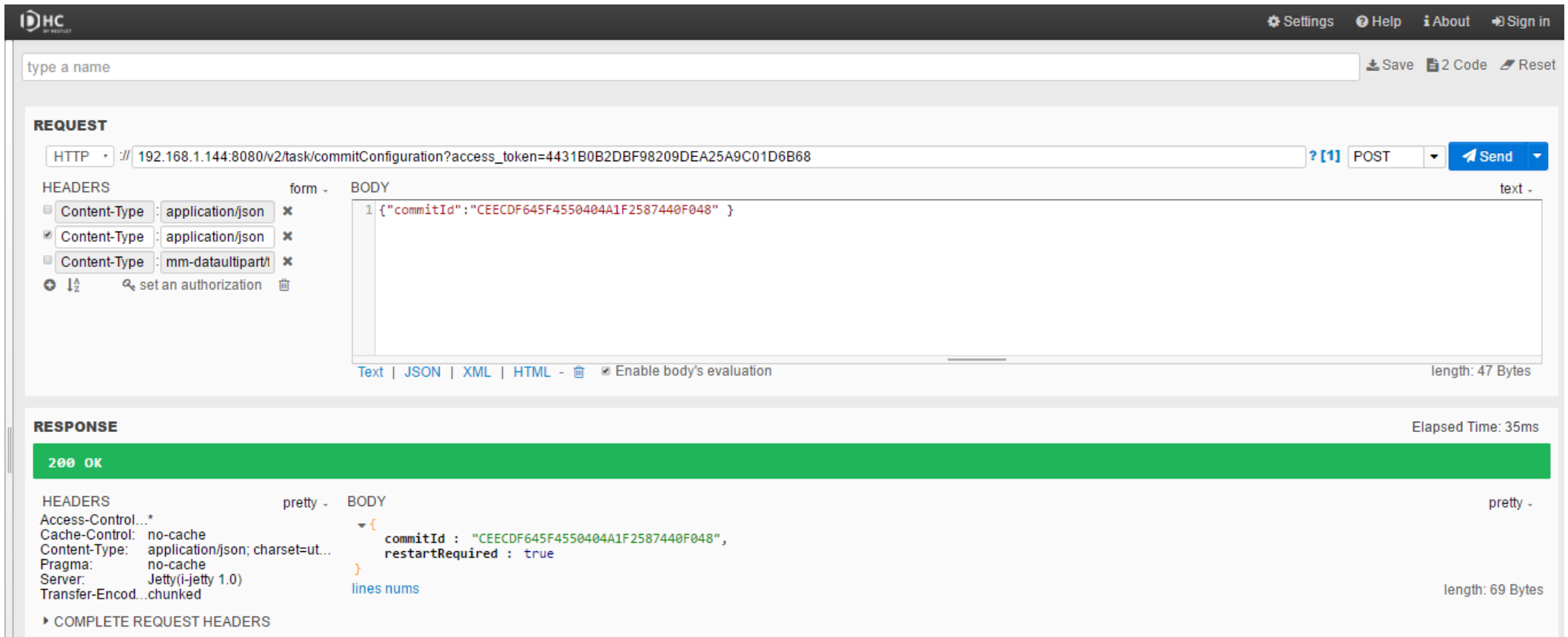
- System Management- POST task/commitConfiguration
  - **Description:** Commit new configuration to player
  - **Input:** JSON or multi-part form

Parameter	Function
commitId	the ID returned when importConfiguration

- **Output:** JSON

Parameter	Function
commitId	return the ID just committed
restartRequired	true/false, if restart is required for changes to take effect, restartRequired is true

- **Practical:** Commit configuration to new player



The screenshot shows the REST Client interface with the following details:

- Request:**
  - Method: **POST**
  - URL: `http://192.168.1.144:8080/v2/task/commitConfiguration?access_token=4431B0B2DBF98209DEA25A9C01D6B68`
  - Headers:
    - Content-Type: application/json
    - Content-Type: application/json
    - Content-Type: mm-dataupload
  - Body: `{ "commitId": "CEEEDF645F4550404A1F2587440F048" }`
- Response:**
  - Status: **200 OK**
  - Elapsed Time: 35ms
  - Headers:
    - Access-Control-Allow-Origin: \*
    - Cache-Control: no-cache
    - Content-Type: application/json; charset=utf-8
    - Pragma: no-cache
    - Server: Jetty(i-jetty 1.0)
    - Transfer-Encoding: chunked
  - Body: `{ "commitId": "CEEEDF645F4550404A1F2587440F048", "restartRequired": true }`



## Developer Guide

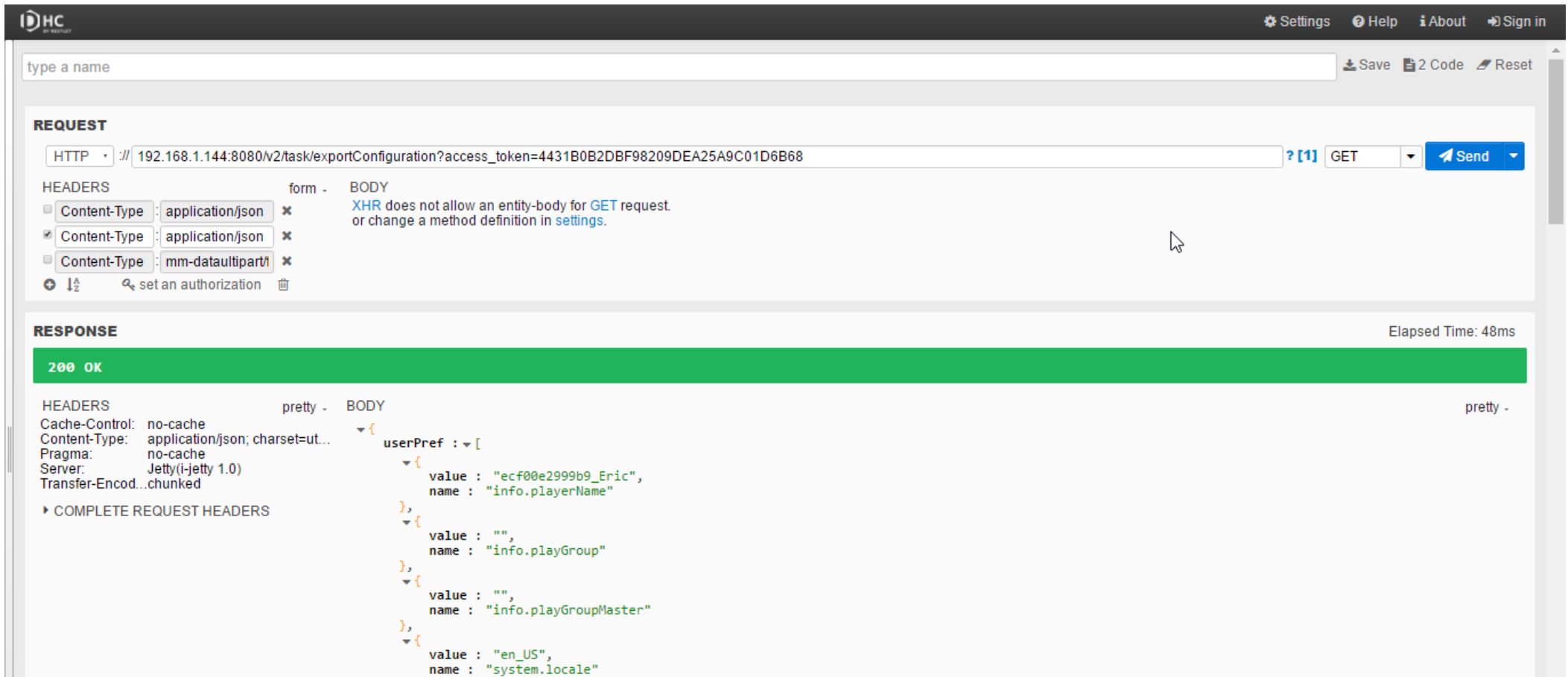
- System Management- GET task/exportConfiguration
  - **Description:** Get configuration from player
  - **Input:** JSON or multi-part form

Parameter	Function
null	null

- **Output:** JSON

Parameter	Function
userPref	return the player configuration { "userPref": [{"value": "", "value"; ""}], {...}... } }

- **Practical:** Get configuration from player



The screenshot shows a REST client interface with the following details:

- REQUEST:**
  - Method: HTTP
  - URL: `://192.168.1.144:8080/v2/task/exportConfiguration?access_token=4431B0B2DBF98209DEA25A9C01D6B68`
  - Method: GET
  - Buttons: Save, Code, Reset, Send
  - HEADERS:
    - Content-Type: application/json
    - Content-Type: application/json
    - Content-Type: mm-datamultipart/
  - Message: "XHR does not allow an entity-body for GET request. or change a method definition in settings."
- RESPONSE:**
  - Status: 200 OK
  - Elapsed Time: 48ms
  - HEADERS:
    - Cache-Control: no-cache
    - Content-Type: application/json; charset=ut...
    - Pragma: no-cache
    - Server: Jetty(i-jetty 1.0)
    - Transfer-Encod...chunked
  - COMPLETE REQUEST HEADERS
  - BODY (JSON):

```
userPref : [
  {
    value : "ecf00e2999b9_Eric",
    name : "info.playerName"
  },
  {
    value : "",
    name : "info.playGroup"
  },
  {
    value : "",
    name : "info.playGroupMaster"
  },
  {
    value : "en_US",
    name : "system.locale"
  }
]
```

## Developer Guide

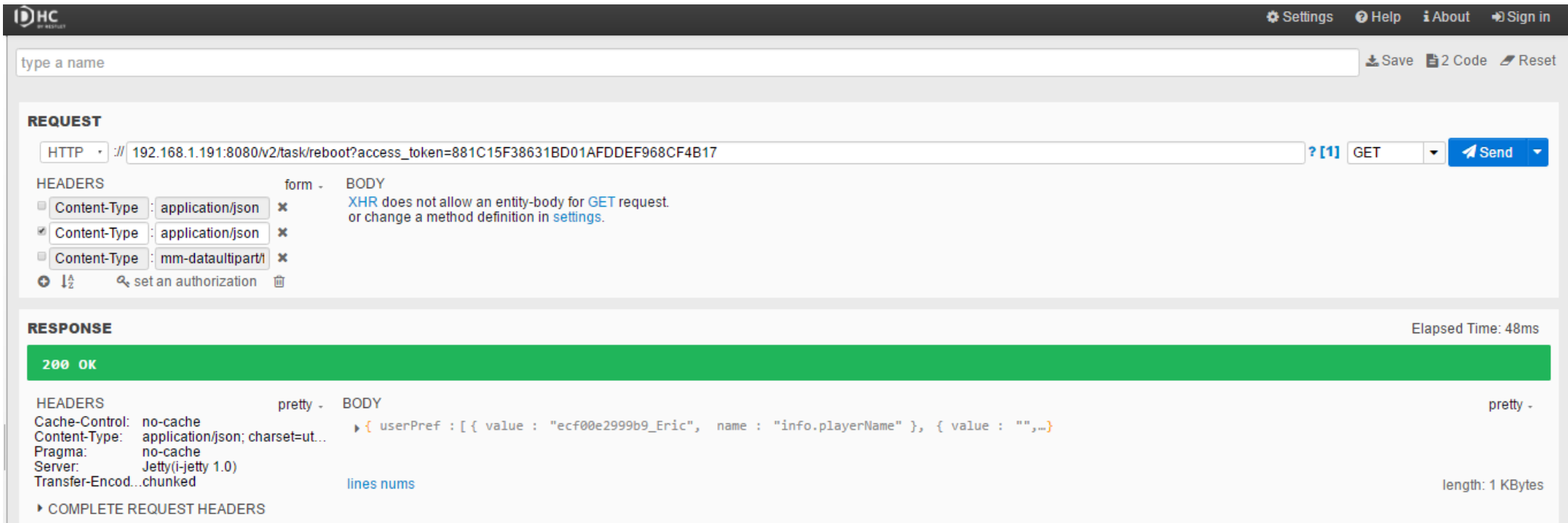
- System Management- POST/ task/reboot
  - **Description:** Reboot player immediately
  - **Input:** JSON or multi-part form

Parameter	Function
Null	Null

- **Output:** JSON

Parameter	Function
Null	Null

- **Practical:** Reboot player immediately



The screenshot shows a REST client interface with the following details:

- Request:**
  - Method: HTTP GET
  - URL: `192.168.1.191:8080/v2/task/reboot?access_token=881C15F38631BD01AFDDEF968CF4B17`
  - Headers:
    - Content-Type: application/json
    - Content-Type: application/json
    - Content-Type: mm-datauploadpart1
  - Body: XHR does not allow an entity-body for GET request. or change a method definition in settings.
- Response:**
  - Status: 200 OK
  - Elapsed Time: 48ms
  - Headers:
    - Cache-Control: no-cache
    - Content-Type: application/json; charset=utf-8
    - Pragma: no-cache
    - Server: Jetty(i-jetty 1.0)
    - Transfer-Encoding: chunked
  - Body: `{ userPref : [ { value : "ecf00e2999b9_Eric", name : "info.playerName" }, { value : "", ... } ] }`
  - Length: 1 KBytes

## Developer Guide

- System Management- GET task/screenshot
  - **Description:** Get screenshot from player
  - **Input:** JSON or multi-part form

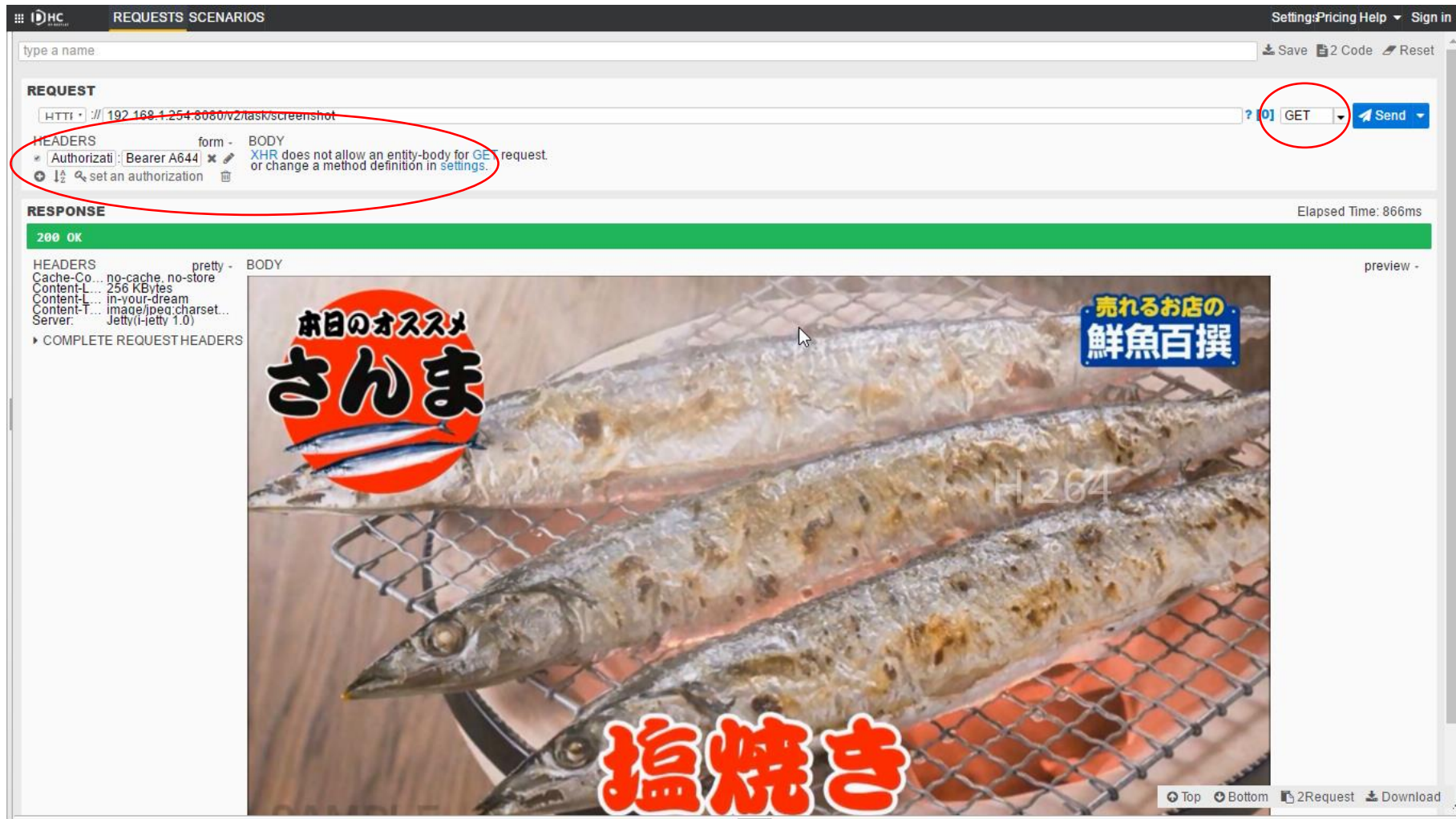
Parameter	Function
Authorization	Verify correctness of player's ""token_type" &" access_token" format as below "Bearer player's access_token"

- **Output:** JSON

Notice: There is a space between "token\_type" and "access token"

Parameter	Function
Null	Null

- **Practical:** Get screenshot from player



## Developer Guide

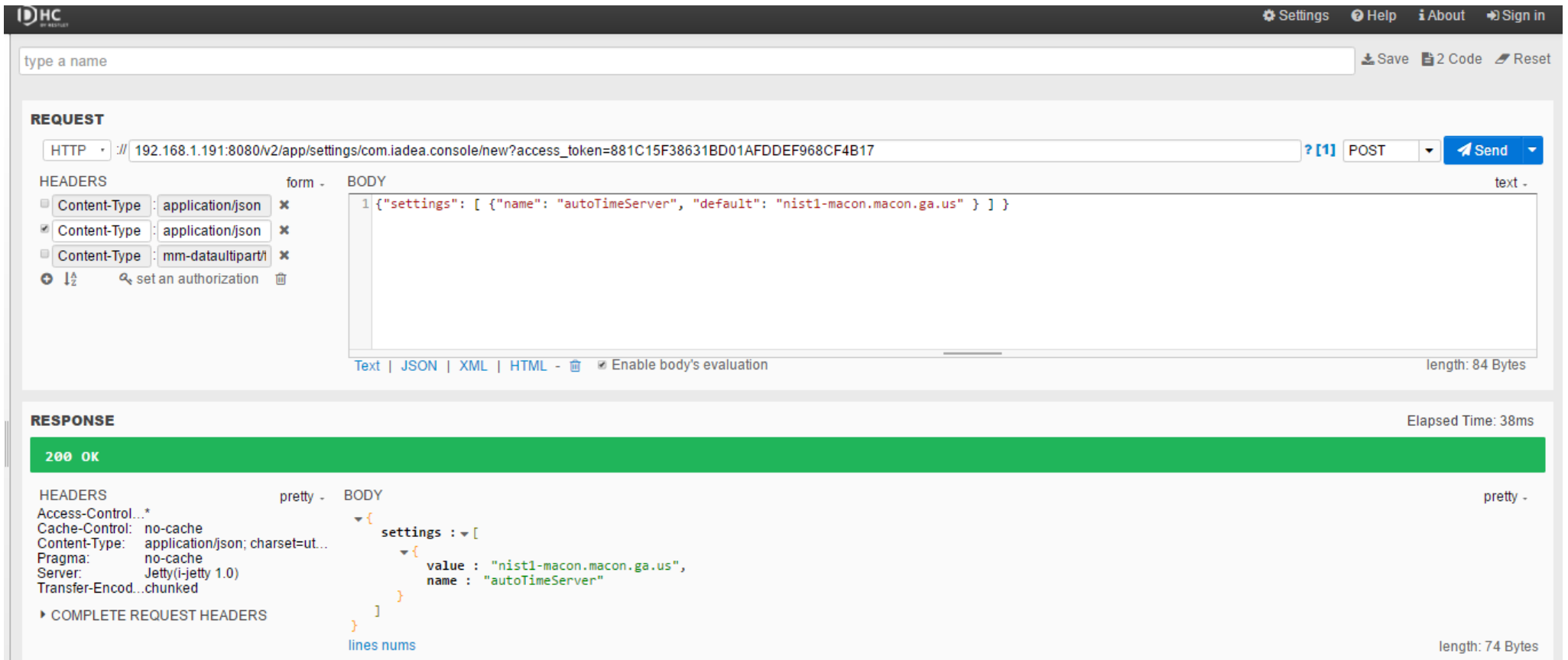
- System Management- POST app/settings/com.iadea.console/new
  - **Description:** Add time server configuration
  - **Input:** JSON

Parameter	Function
Settings	new setting object ex. {"settings": [ {"name": "autoTimeServer", "default": "ntp://host{:port}" } ] } "name": must be autoTimeServer "default": default   ntp://host{:port} set to default player synchronize time with pool.ntp.org, otherwise synchronize with specified ntp host

- **Output:** JSON

Parameter	Function
Value	return the default value configured above
Name	return name

➤ **Practical:** Add time server configuration



The screenshot shows a REST client interface with the following details:

- Request:**
  - Method: HTTP
  - URL: `:// 192.168.1.191:8080/v2/app/settings/com.idea.console/new?access_token=881C15F38631BD01AFDDEF968CF4B17`
  - Method: POST
  - Body: `{ "settings": [ { "name": "autoTimeServer", "default": "nist1-macon.macon.ga.us" } ] }`
  - Headers:
    - Content-Type: application/json
    - Content-Type: application/json
    - Content-Type: mm-dataupload
- Response:**
  - Status: 200 OK
  - Elapsed Time: 38ms
  - Body: `{ "settings": [ { "value": "nist1-macon.macon.ga.us", "name": "autoTimeServer" } ] }`
  - Headers:
    - Access-Control-Allow-Origin: \*
    - Cache-Control: no-cache
    - Content-Type: application/json; charset=utf-8
    - Pragma: no-cache
    - Server: Jetty(i-jetty 1.0)
    - Transfer-Encoding: chunked



## Developer Guide

- System Management- POST app/settings/com.iadea.console/update
  - **Description:** Update time server configuration
  - **Input:** JSON

Parameter	Function
Settings	new setting object ex. {"settings": [ {"name": "autoTimeServer", "value": "ntp://host{port}" } ] } "name": must be autoTimeServer "value" : default   ntp://host{port} set to default player synchronize time with pool.ntp.org, otherwise synchronize with specified ntp host

- **Output:** JSON

Parameter	Function
Value	return the default value configured above
Name	return name

➤ **Practical:** Update time server configuration

The screenshot shows the IDE's REST client interface. At the top, there's a search bar with the text "type a name" and buttons for "Save", "2 Code", and "Reset". Below this, the "REQUEST" section is active, showing a POST request to the URL `://192.168.1.191:8080/v2/app/settings/com.idea.console/update?access_token=881C15F38631BD01AFDDEF968CF4B17`. The request body is a JSON array: `[{"settings": [{"name": "autoTimeServer", "value": "nisttime.carsoncity.k12.mi.us"}]}]`. The response section shows a `200 OK` status with an elapsed time of 19ms. The response body is a JSON object: `{ settings: [ { value: "nisttime.carsoncity.k12.mi.us", name: "autoTimeServer" } ] }`.

## Developer Guide

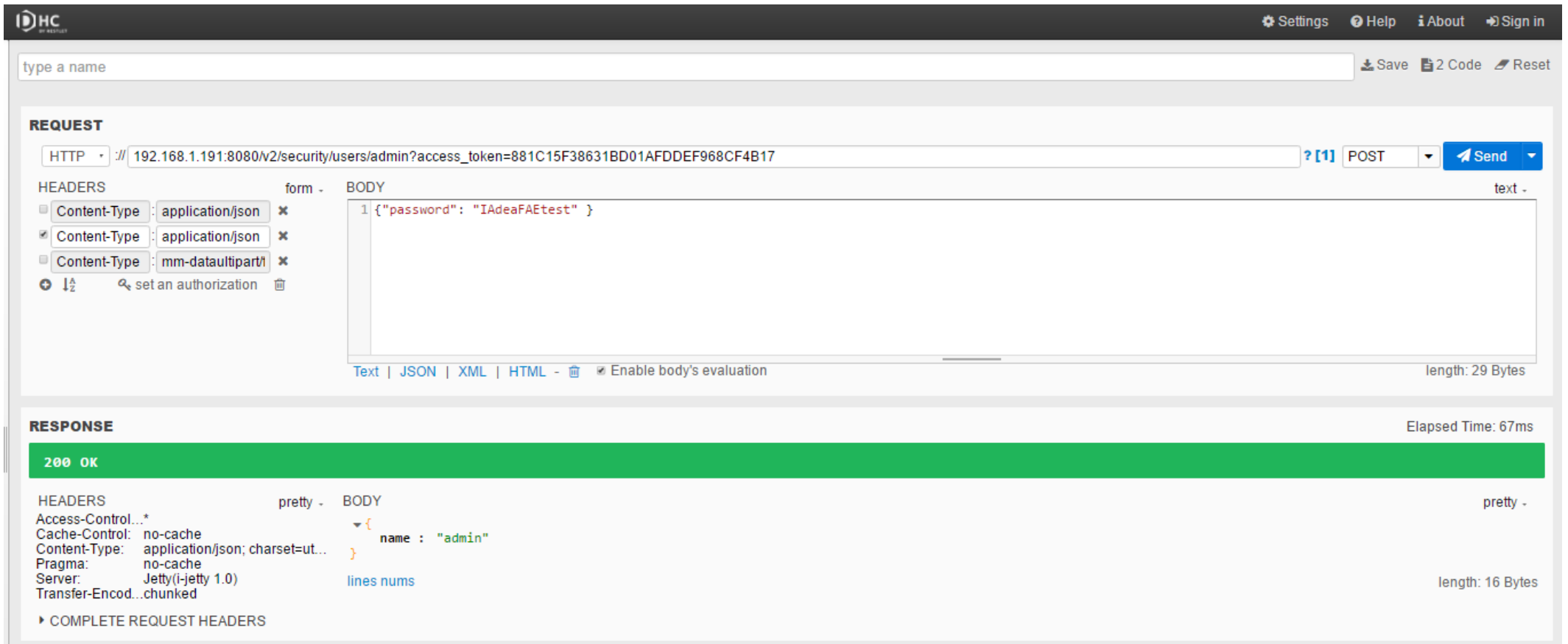
- System Management- POST security/users/admin
  - **Description:** Update device password
  - **Input:** JSON

Parameter	Function
Password	String

- **Output:** JSON

Parameter	Function
Name	return admin by default

➤ **Practical:** Update device password



The screenshot shows a REST client interface with the following details:

- Request:**
  - Method: `POST`
  - URL: `:// 192.168.1.191:8080/v2/security/users/admin?access_token=881C15F38631BD01AFDDEF968CF4B17`
  - Headers:
    - `Content-Type: application/json` (3 instances)
    - `mm-dataupload`
  - Body: `{ "password": "IAdeaFAEtest" }`
  - Length: 29 Bytes
- Response:**
  - Status: `200 OK`
  - Elapsed Time: 67ms
  - Headers:
    - `Access-Control-Allow-Origin: *`
    - `Cache-Control: no-cache`
    - `Content-Type: application/json; charset=utf-8`
    - `Pragma: no-cache`
    - `Server: Jetty(i-jetty 1.0)`
    - `Transfer-Encoding: chunked`
  - Body: `{ "name": "admin" }`
  - Length: 16 Bytes

### System Information

- System Information- GET system/playerInfo
  - **Description:** Get player information
  - **Input:** null

Parameter	Function
Null	Null

- **Output:** JSON

Parameter	Function
playerId	Returns the player Id
serialNumber	Returns Ethernet MAC Id address
factoryDefaults	Returns 'add-on' package name

## Developer Guide

- **Practical:** Get the Ethernet information of network connection via Static IP



The screenshot shows a REST client interface with the following configuration:

- METHOD:** GET
- URL:** http://10.0.10.126:8080/v2/system/playerInfo?access\_token=D2BB69CEC6B363F891B89E4C7C363D38
- QUERY PARAMETERS:** access\_token = D2BB69CEC6B363F891B89E4C7C363D38
- Buttons:** Send, Add query parameter

```
{
  playerId : "9e7df0ed-2a5c-4a19-bec7-2cc54800900c",
  serialNumber : "2cc54800900c",
  factoryDefaults : []
}
```

## Developer Guide

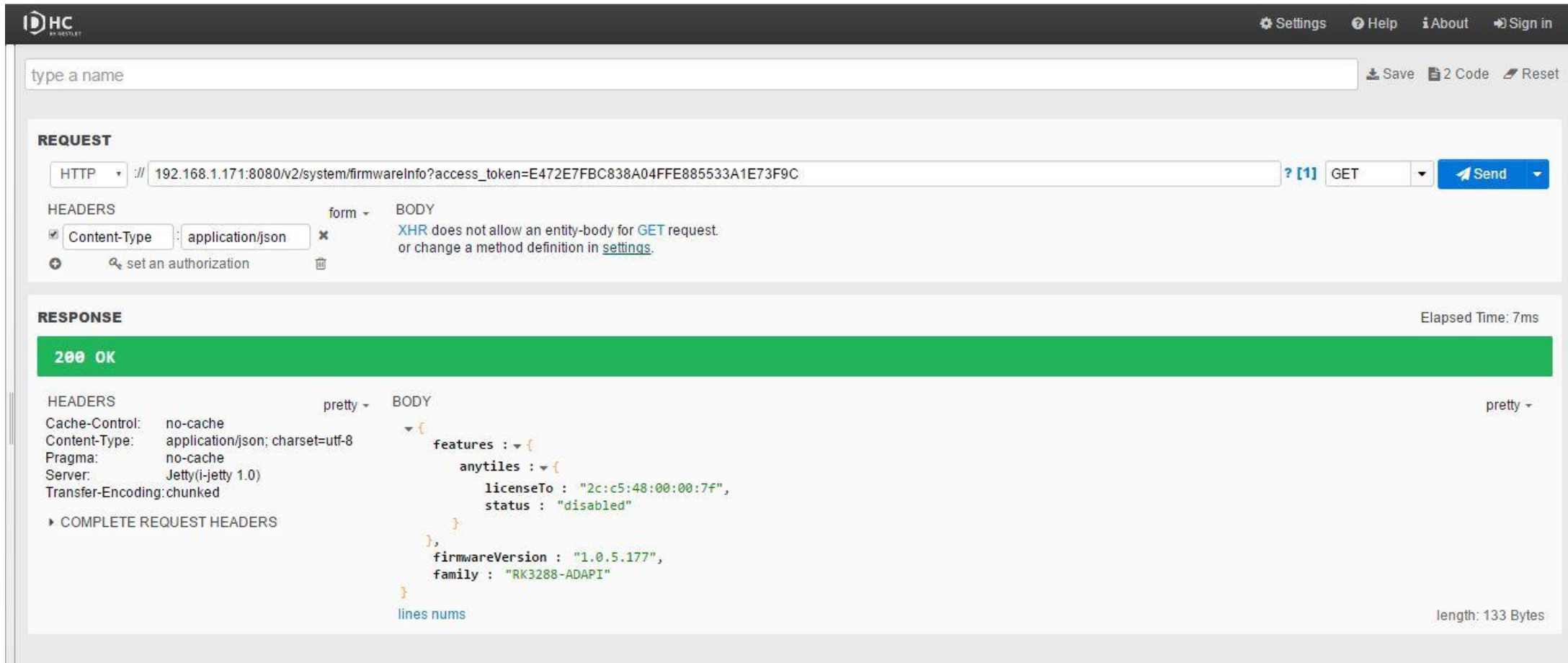
- System Information- GET system/firmwareInfo
  - **Description:** Get firmware information from player
  - **Input:** null

Parameter	Function
Null	Null

- **Output:** JSON

Parameter	Function
firmware version	Device firmware version information
family	Product model family, such as "AML8726M3-ADAPI"

➤ Practical:



The screenshot shows a REST client interface with the following details:

- Request:** Method: HTTP, URL: `:// 192.168.1.171:8080/v2/system/firmwareInfo?access_token=E472E7FBC838A04FFE885533A1E73F9C`, Method: GET. Headers include `Content-Type: application/json`. A message states: "XHR does not allow an entity-body for GET request. or change a method definition in settings."
- Response:** Status: 200 OK, Elapsed Time: 7ms. Headers include `Cache-Control: no-cache`, `Content-Type: application/json; charset=utf-8`, `Pragma: no-cache`, `Server: Jetty(i-jetty 1.0)`, and `Transfer-Encoding: chunked`. The body is a JSON object: 

```
{  "features": {    "anytiles": {      "licenseTo": "2c:c5:48:00:00:7f",      "status": "disabled"    }  },  "firmwareVersion": "1.0.5.177",  "family": "RK3288-ADAPI"}
```



## Developer Guide

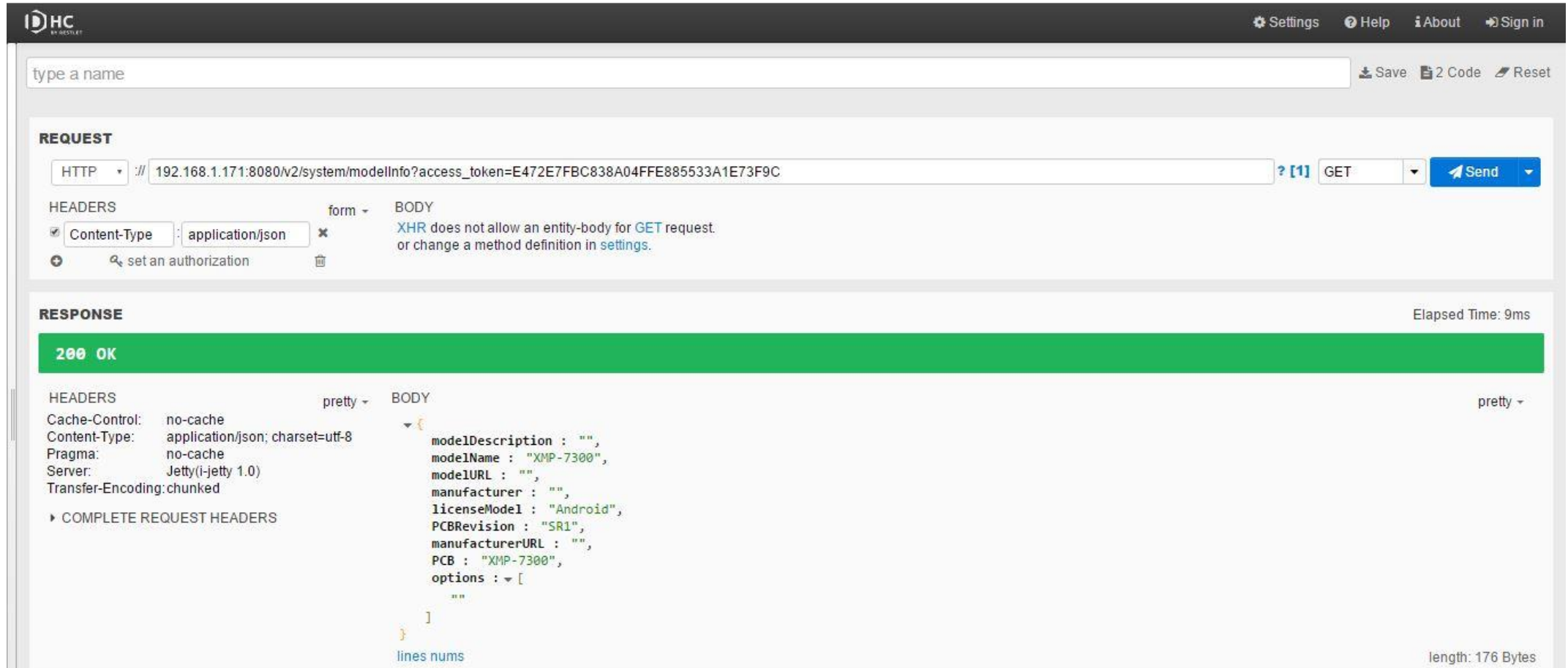
- System Information- GET system/modelInfo
  - **Description:** Get player model name and other manufacturer use only information
  - **Input:** null

Parameter	Function
Null	Null

- **Output:** JSON

Parameter	Function
modelName	Device model name

➤ **Practical:** Get the system information on device



The screenshot shows a REST client interface with the following details:

- Request:**
  - Method: HTTP
  - URL: `192.168.1.171:8080/v2/system/modelInfo?access_token=E472E7FBC838A04FFE885533A1E73F9C`
  - Method: GET
  - Headers: Content-Type: application/json
  - Body: XHR does not allow an entity-body for GET request. or change a method definition in settings.
- Response:**
  - Status: 200 OK
  - Elapsed Time: 9ms
  - Headers:
    - Cache-Control: no-cache
    - Content-Type: application/json; charset=utf-8
    - Pragma: no-cache
    - Server: Jetty(i-jetty 1.0)
    - Transfer-Encoding: chunked
  - Body (JSON):

```

{
  "modelDescription": "",
  "modelName": "XMP-7300",
  "modelURL": "",
  "manufacturer": "",
  "licenseModel": "Android",
  "PCBRevision": "SR1",
  "manufacturerURL": "",
  "PCB": "XMP-7300",
  "options": [
    ""
  ]
}

```
  - Length: 176 Bytes

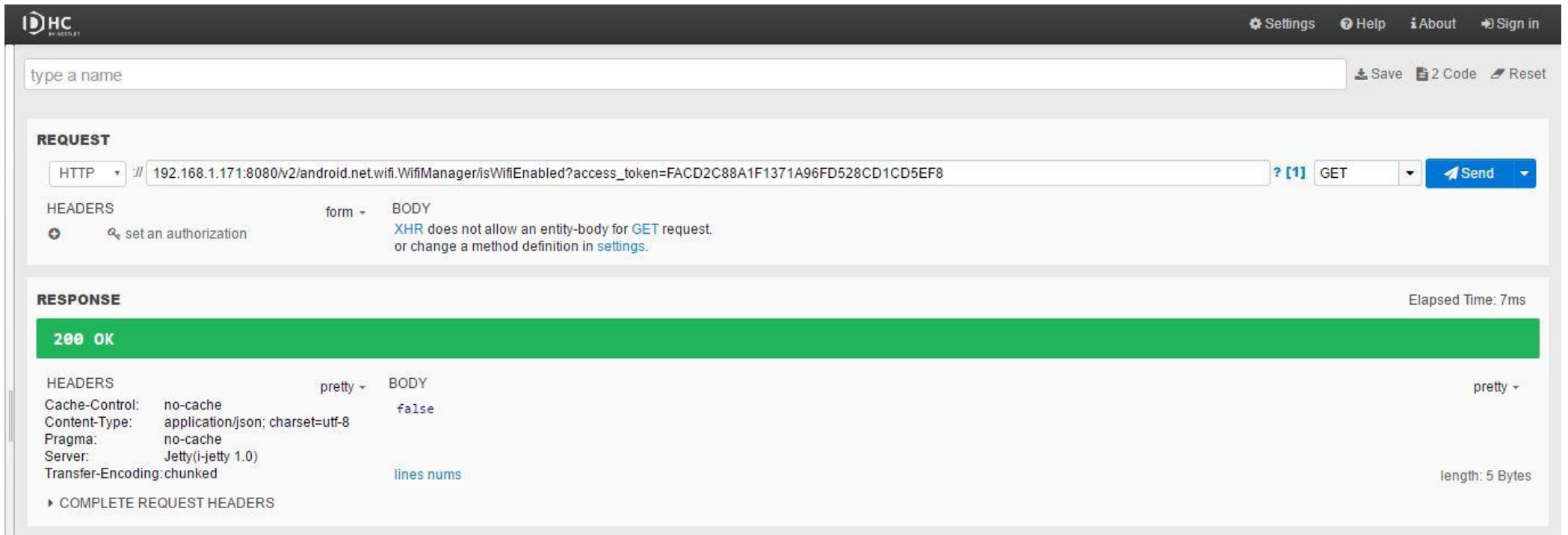
## Developer Guide

- System Information- GET `android.net.wifi.WifiManager/isWiFiEnabled`
  - **Description:** Get Wi-Fi enable/disable information
  - **Input:** null

Parameter	Function
Null	Null

- **Output:** True/False

### ➤ Practical: Checking Wi-Fi Status



The screenshot shows a REST client interface with the following details:

- REQUEST:**
  - Method: HTTP
  - URL: // 192.168.1.171:8080/v2/android.net.wifi.WifiManager/isWifiEnabled?access\_token=FACD2C88A1F1371A96FD528CD1CD5EF8
  - Method: GET
  - Buttons: Save, 2 Code, Reset, Send
  - HEADERS: set an authorization
  - form: BODY
  - Message: XHR does not allow an entity-body for GET request. or change a method definition in settings.
- RESPONSE:**
  - Status: 200 OK
  - Elapsed Time: 7ms
  - HEADERS:
    - Cache-Control: no-cache
    - Content-Type: application/json; charset=utf-8
    - Pragma: no-cache
    - Server: Jetty(i-jetty 1.0)
    - Transfer-Encoding: chunked
  - BODY: false
  - length: 5 Bytes
  - Buttons: pretty, lines nums, COMPLETE REQUEST HEADERS



## REST-API Guide

### Developer Guide

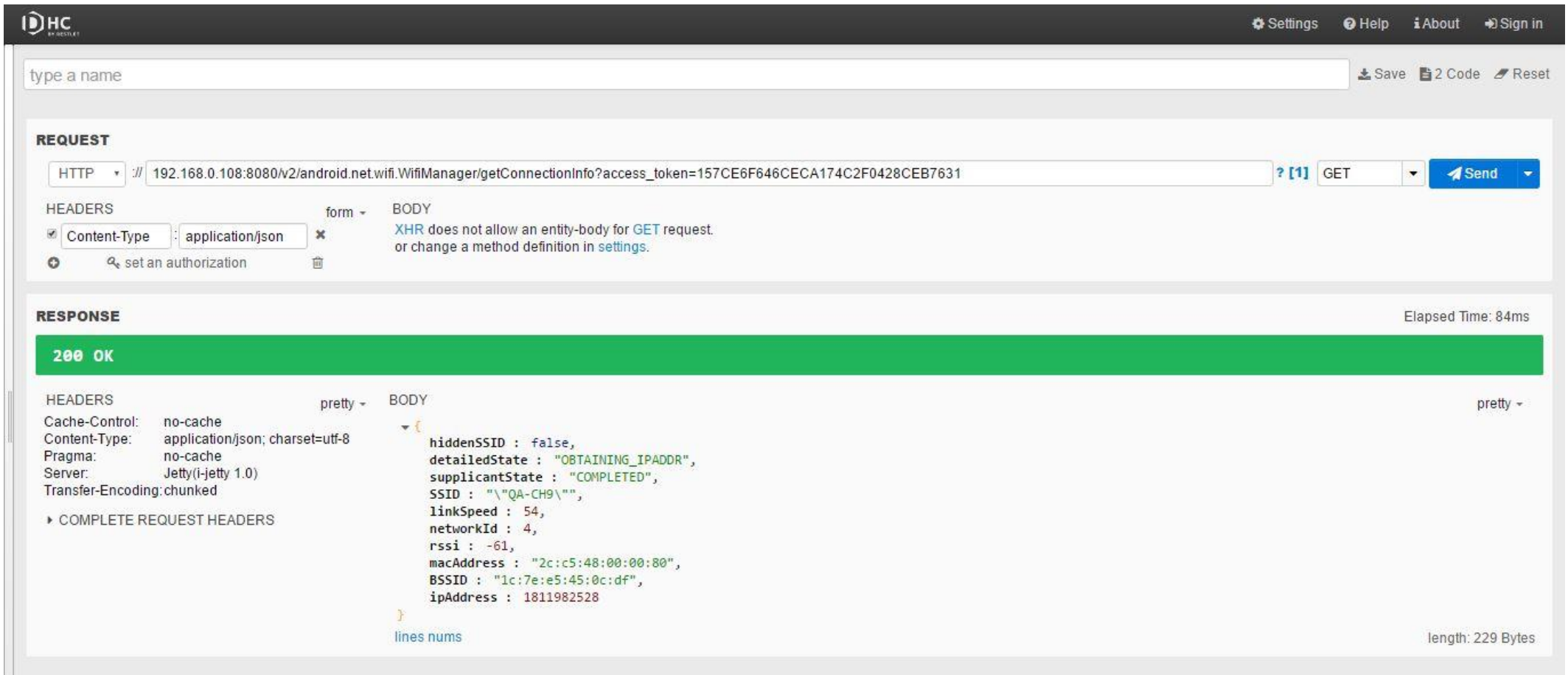
- System Information- GET android.net.wifi.WifiManager/get ConnectionInfo
  - **Description:** Get Wi-Fi information
  - **Input:** null

Parameter	Function
Null	Null

➤ **Output: JSON**

Parameter	Function
<b>BSSID</b>	BSSID
<b>detailedState</b>	Detailed State
<b>hiddenSSID</b>	Whether SSID is hidden
<b>ipAddress</b>	IP Address
<b>linkSpeed</b>	Link Speed
<b>macAddress</b>	MAC Address
<b>networkId</b>	Network ID
<b>rsi</b>	RSSI info
<b>SSID</b>	SSID
<b>supplicantState</b>	Supplicant State

- **Practical:** Get the information of network connection



The screenshot shows a REST client interface with the following details:

- Request:**
  - Method: HTTP
  - URL: `192.168.0.108:8080/v2/android.net.wifi.WifiManager/getConnectionInfo?access_token=157CE6F646CECA174C2F0428CEB7631`
  - Method: GET
  - Headers: Content-Type: application/json
  - Body: XHR does not allow an entity-body for GET request. or change a method definition in settings.
- Response:**
  - Status: 200 OK
  - Elapsed Time: 84ms
  - Headers:
    - Cache-Control: no-cache
    - Content-Type: application/json; charset=utf-8
    - Pragma: no-cache
    - Server: Jetty(i-jetty 1.0)
    - Transfer-Encoding: chunked
  - Body (JSON):

```

{
  "hiddenSSID": false,
  "detailedState": "OBTAINING_IPADDR",
  "supplicantState": "COMPLETED",
  "SSID": "\"QA-CH9\"",
  "linkSpeed": 54,
  "networkId": 4,
  "rssi": -61,
  "macAddress": "\"2c:c5:48:00:00:80\"",
  "BSSID": "\"1c:7e:e5:45:0c:df\"",
  "ipAddress": 1811982528
}

```
  - length: 229 Bytes

## Developer Guide

- System Information- GET `android.net.ethernet.EthernetManager/getSavedEthConfig`
  - **Description:** Get Ethernet information
  - **Input:** null

Parameter	Function
Null	Null

- **Output:** JSON

Parameter	Function
<b>dnsAddr</b>	Returns the DNS 1 and DNS 2 in Static IP settings
<b>netmask</b>	Returns net mask Address in Static IP settings
<b>connectMode</b>	Returns 'DHCP' or 'manual'(Static IP)
<b>ipAddress</b>	Returns the IP address in Static IP settings
<b>ifName</b>	Returns the interface name from the saved configuration

Note. The parameters listed above except connectMode are all for Static IP.

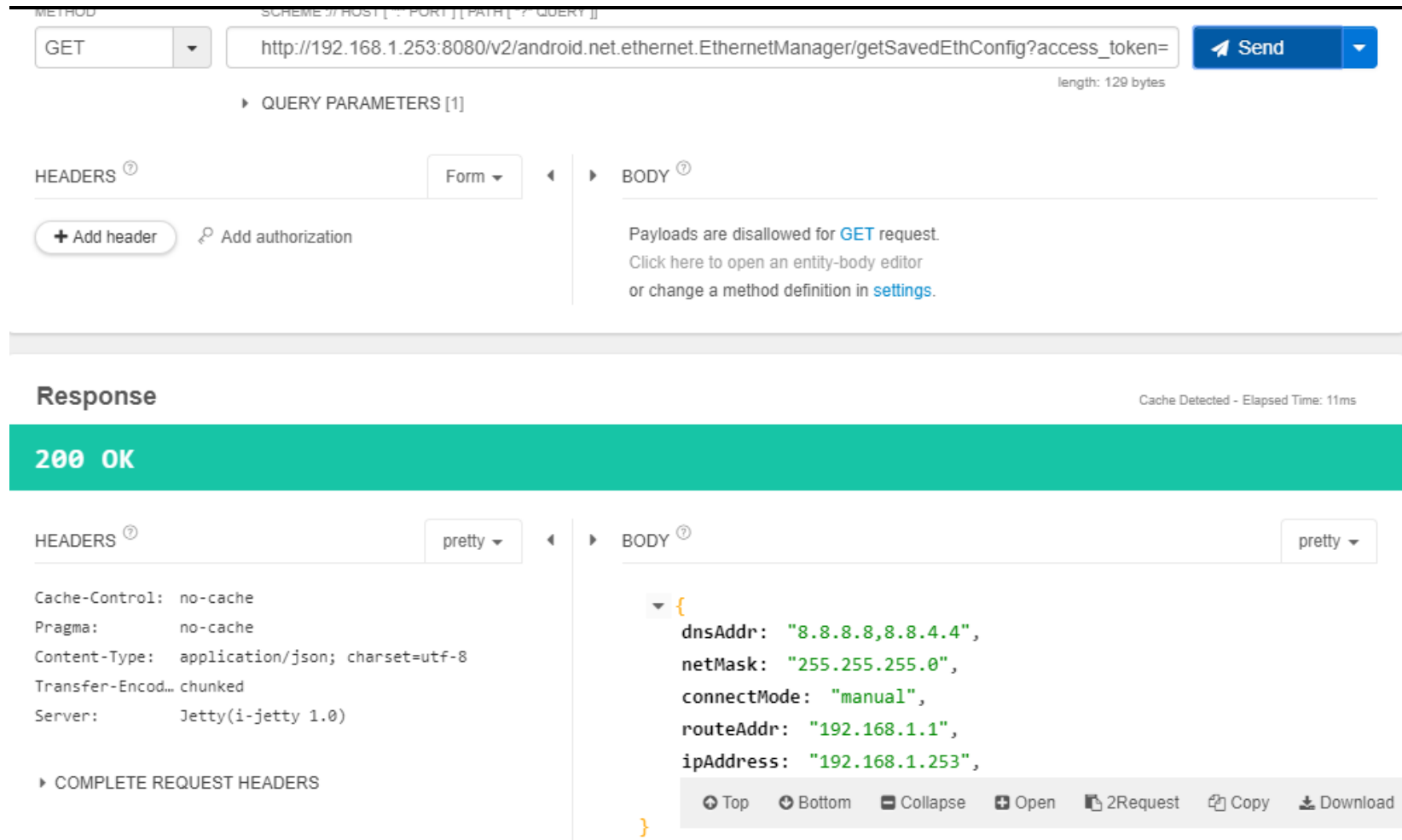
If device is via DHCP, you will see connectMode: DHCP, then please go next command

GET `android.net.ethernet.EthernetManager/getDhcpInfo`



## Developer Guide

- **Practical:** Get the Ethernet information of network connection via Static IP



The screenshot displays a REST client interface with the following details:

- Request:**
  - Method: GET
  - URL: `http://192.168.1.253:8080/v2/android.net.ethernet.EthernetManager/getSavedEthConfig?access_token=`
  - Query Parameters: 1
  - Length: 129 bytes
  - Body: Payloads are disallowed for GET request.
- Response:**
  - Status: 200 OK
  - Cache: Cache Detected - Elapsed Time: 11ms
  - Headers:
    - Cache-Control: no-cache
    - Pragma: no-cache
    - Content-Type: application/json; charset=utf-8
    - Transfer-Encod... chunked
    - Server: Jetty(i-jetty 1.0)
  - Body (pretty):

```
{
  dnsAddr: "8.8.8.8,8.8.4.4",
  netMask: "255.255.255.0",
  connectMode: "manual",
  routeAddr: "192.168.1.1",
  ipAddress: "192.168.1.253",
}
```

## Developer Guide

- System Information- GET `android.net.ethernet.EthernetManager/getDhcpInfo`
  - **Description:** Get Ethernet information (via DHCP)
  - **Input:** null

Parameter	Function
Null	Null

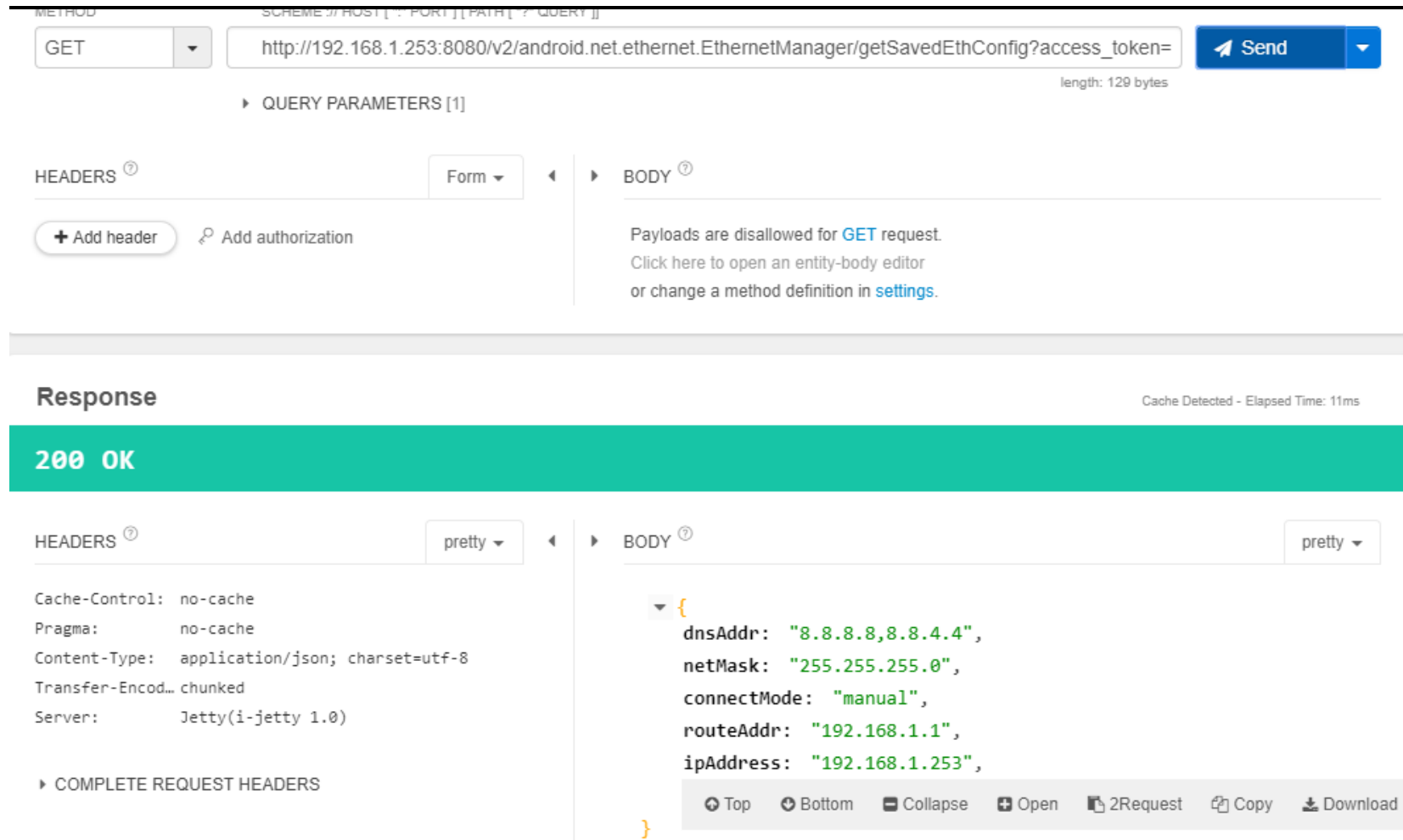
- **Output:** JSON

Parameter	Function
<b>dns1</b>	Returns the IP address of DNS 1
<b>dns2</b>	Returns the IP address of DNS 2
<b>leaseDuration</b>	Returns the duration that a device reserves an IP address on your network.
<b>serverAddress</b>	Returns IP Address of DHCP server
<b>gateway</b>	Returns the IP address of gateway
<b>netmask</b>	Returns Net mask Address
<b>ipAddress</b>	Returns the IP address of IAdea device

Note. After getting the info, you'll have to convert IPV4 to string. To convert IPV4 to string, please refer this article: <https://gist.github.com/werbet/2643813>

## Developer Guide

- **Practical:** Get the Ethernet information of network connection via DHCP



The screenshot displays a REST client interface with the following details:

- Request:**
  - Method: GET
  - URL: `http://192.168.1.253:8080/v2/android.net.ethernet.EthernetManager/getSavedEthConfig?access_token=`
  - Length: 129 bytes
  - Query Parameters: [1]
  - Headers: Form
  - Body: Payloads are disallowed for GET request. Click here to open an entity-body editor or change a method definition in settings.
- Response:**
  - Status: 200 OK
  - Cache Detected - Elapsed Time: 11ms
  - Headers:
    - Cache-Control: no-cache
    - Pragma: no-cache
    - Content-Type: application/json; charset=utf-8
    - Transfer-Encod... chunked
    - Server: Jetty(i-jetty 1.0)
  - Body (pretty):

```
{
  dnsAddr: "8.8.8.8,8.8.4.4",
  netMask: "255.255.255.0",
  connectMode: "manual",
  routeAddr: "192.168.1.1",
  ipAddress: "192.168.1.253",
}
```

## App Solutions

IAdea App solutions APIs enable developer to manage their APK, calendar and media components. Developers can use solutions API to update, delete and sync these components.

- App Solutions- [POST /v2/app/solutions/list](#)
  - **Description:** See the list of the installed apps.
  - **Input:** JSON

Parameter	Function
Null	Null

- **Output:** JSON

Parameter	Function
name	App name, string
version	Version number, string
description	App description, string
state	Possible values: "SYNCING", "READY", "ERROR", "UNREADY"

## Developer Guide

➤ **Practical:** See what app is installed.

**solutions/list**

METHOD: POST | SCHEME://HOST [":" PORT] [PATH [ "?" QUERY ]]

http://172.20.10.12:8080/v2/app/solutions/list?access\_token=19B602DAFBD6099AC98D646D8F6D0AF length: 91 byte(s)

QUERY PARAMETERS [1]

HEADERS <sup>Ⓢ</sup> ↓  Content-Type : application/json  Add header  Add authorization

BODY <sup>Ⓢ</sup>   Enable body evaluation length: 2 bytes

---

**200 OK**

HEADERS <sup>Ⓢ</sup> pretty

```
Access-Control-Allow-Origin: *
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Server: Jetty(i-jetty 2.0.0-78+bundle)
```

BODY <sup>Ⓢ</sup> pretty

```
[
  {
    name: "XXXXXXXXXX",
    version: "9.2",
    description: {
      default: "XXXXXXXXXX Solution"
    },
    state: "READY"
  }
]
```

[lines nums](#) length: 99 bytes

## Developer Guide

- App Solutions- POST /v2/app/solutions/insert
  - **Description:** Install App
  - **Input:** JSON

Parameter	Function
<b>name</b>	App name, string
<b>version</b>	Version number, string
<b>description</b>	App description, string
<b>components</b>	Component-key is a string for this component <pre> {   "component-key":{     "type":"APK",     "packageName":"string",     "versionCode":"number   decimal-string",     "versionName":"string",     "link":"string",     "allowUninstall":"boolean"   } }           </pre>

➤ **Output: JSON**

Parameter	Function
<b>name</b>	App name, string
<b>version</b>	Version number, string
<b>description</b>	App description, string
<b>components</b>	<p>Component-key is a string for this component</p> <pre>{   "component-key":{     "type":"APK",     "packageName":"string",     "versionCode":"number   decimal-string",     "versionName":"string",     "link":"string",     "allowUninstall":"boolean"   } }</pre>
<b>state</b>	Possible values: <b>“SYNCING”</b> , <b>“READY”</b> , <b>“ERROR”</b> , <b>“UNREADY”</b>
<b>detail</b>	<pre>"details":{   "firstInstallTime":iso8601-datetime-string,   "lastModifyTime":iso8601-datetime-string,   "lastSyncTime":iso8601-datetime-string,   "signingInfo":{     "..."   } }</pre>

## Developer Guide

```
    },  
    "componentStatus":{  
      "component-key":{  
        "state":"string",  
        "progress":"number",  
        "contentUri":"string"  
      },  
      "..."  
    }  
  }  
}
```



## Developer Guide

### ➤ Practical: Install a 3<sup>rd</sup> party App

**solutions/insert**

METHOD: POST | SCHEME // HOST [ : PORT ] [ PATH [ ? QUERY ] ]

http://10.0.10.177:8080/v2/app/solutions/insert?access\_token=80F026A05D8423CE0E17BBBC1B5D4

➤ QUERY PARAMETERS [1]

HEADERS [1] | Form | BODY

Content-Type: application/json

[+ Add header](#) | [Add authorization](#)

```

1 {
2   "name": "Go...t",
3   "version": "9.2",
4   "description": "Go...t Solution",
5   "components": {
6     "Go...t APK": {
7       "type": "APK",
8       "packageName": "com.g...t.br...ng.display",
9       "versionCode": 92,
10      "versionName": "9.2",
11      "link": "https://install.g...t.cloud/mwdeviceapp/?mode=download&version=9.2"
12    }
13  }
14 }

```

Text JSON XML HTML | [Format body](#) |  Enable body evaluation

**Response**

**200 OK**

HEADERS [1] | pretty | BODY

Access-Control-Allow-Origin: \*

Cache-Control: no-cache

Pragma: no-cache

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Server: Jetty(1-jetty 2.0.0-76+platform)

▼ COMPLETE REQUEST HEADERS | pretty

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36

Content-Type: application/json

Accept: /\*

Origin: chrome-extension://aejoelaoggenbcahagindiliamicdfm

Accept-Encoding: gzip, deflate

Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7,zh-CN;q=0.6

Note: XHR automatically adds headers like Accept, Accept-Language, Cookie, User-Agent, etc.

```

{
  name : "Go...t",
  version : "9.2",
  description : {
    default : "Go...t Solution"
  },
  components : {
    Go...t APK : { type : "APK", packageName : "com.g...t.br...ng.display", versionCode : "92",...}
  },
  state : "SYNCING",
  details : {
    firstInstallTime : "2020-09-18T05:31:14Z",
    lastModifyTime : "2020-09-18T05:31:14Z",
    componentStatus : {
      Go...t APK : {
        state : "PENDING"
      }
    }
  }
}

```

lines nums

## Developer Guide

- **Description:** Insert an iCalendar
- **Input:** JSON

Parameter	Function
<b>name</b>	Calendar name, string
<b>version</b>	Version number, string
<b>description</b>	Calendar description, string
<b>components</b>	<p>Component-key is a string for this component</p> <pre>{   "component-key":{     "type":"CALENDAR",     "link":"string",     "checksum":" string",     "checksumAlgorithm":" string",     "mimeType":" string",     "etag":" string",     "event":" string"   } }</pre>

➤ **Output: JSON**

Parameter	Function
<b>name</b>	App name, string
<b>version</b>	Version number, string
<b>description</b>	App description, string
<b>components</b>	<p>Component-key is a string for this component</p> <pre>{   "name": "string",   "version": "number   decimal-string",   "description": {     "default": "string"   },   "components": {     "string": {       "type": "CALENDAR",       "link": "string",       "event": {         "defaultContent": {           "packageName": "string",           "className": "string",           "data": "string"         }       },       "defaultPriority": "number   decimal-string"     }   } }</pre>

## Developer Guide

	<pre>}, "state": "SYNCING", "details": {   "firstInstallTime": "string",   "lastModifyTime": "string",   "componentStatus": {     "string": {       "state": "PENDING"     }   } } }</pre>
<b>state</b>	Possible values: <b>“SYNCING”</b> , <b>“READY”</b> , <b>“ERROR”</b> , <b>“UNREADY”</b>

## Developer Guide

### ➤ Practical: Insert an iCalendar

solutions/insert Calendar

Save

METHOD SCHEME // HOST [:" PORT ] [ PATH [?" QUERY ] ]

POST http://172.20.10.12:8080/v2/app/solutions/insert?access\_token=84911FEDF8C24172E6D3F35A3CBEFF

Send

length: 93 byte(s)

▶ QUERY PARAMETERS [1]

HEADERS ⓘ

Form

▶ BODY ⓘ

Text

Content-Type : application/json

+ Add header Add authorization

```

1 {
2   "name": "Calendar_Google",
3   "version": "1.0",
4   "description": "Calendar for Google",
5   "components": {
6     "CalendarGoogle": {
7       "type": "CALENDAR",
8       "link": "http://www.google.com/Test.ics",
9       "event":
10      {
11        defaultContent :
12        {
13          uri = "http://www.google.com",
14          packageName = "com.iadea.player",
15          className = "com.iadea.player.SmilActivity",
16          Action = "android.intent.action.VIEW"
17        },
18        "defaultPriority": 1
19      }
20    }
21  }
22 }
23 }
24
25

```

Cache Detected - Elapsed Time: 85ms

Response

200 OK

HEADERS ⓘ

pretty

▶ BODY ⓘ

pretty

```

access-control-allow- *
cache-control: no-cache
content-type: application/json; charset=utf-8
pragma: no-cache
server: Jetty(i-jetty 2.0.0-78+bundle)
transfer-encoding: chunked

```

▶ COMPLETE REQUEST HEADERS

```

{
  name: "Calendar_Google",
  version: "1.0",
  description: { default: "Calendar for Google"},
  components: { CalendarGoogle: { type: "CALENDAR", link: "http://www.google.com/Test.ics",
  state: "SYNCING",
  details: { firstInstallTime: "2020-11-01T07:15:43Z", lastModifyTime: "2020-11-01T07:15:43Z" } } }
}

```

Description: Insert a Media

➤ **Input: JSON**

Parameter	Function
<b>name</b>	Calendar name, string
<b>version</b>	Version number, string
<b>description</b>	Media description, string
<b>components</b>	<p>Component-key is a string for this component</p> <pre>{   "component-key":{     "type":"MEDIA",     "link":"string",     "checksum":" string",     "checksumAlgorithm":" string",     "mimeType":" string",     "etag":" string",   } }</pre>

➤ **Output: JSON**

Parameter	Function
<b>name</b>	App name, string
<b>version</b>	Version number, string
<b>description</b>	App description, string
<b>components</b>	<p>Component-key is a string for this component</p> <pre>{   "name": "string",   "version": "number   decimal-string",   "description": {     "default": "string"   },   "components": {     "string": {       "type": "MEDIA",       "link": "string"     }   } }, "state": "SYNCING", "details": {   "firstInstallTime": "string",   "lastModifyTime": "string",   "componentStatus": {     "string": {</pre>

## Developer Guide

	<pre>    "state": "PENDING"   } } }</pre>
<b>state</b>	Possible values: <b>“SYNCING”</b> , <b>“READY”</b> , <b>“ERROR”</b> , <b>“UNREADY”</b>



## Developer Guide

### ➤ Practical: Insert a media

#### solutions/insert Media

METHOD: POST

SCHEME // HOST [\*\*PORT] [PATH [\*\* QUERY ]]

http://172.20.10.12:8080/v2/app/solutions/insert?access\_token=84911FEDF8FC24172E6D3F35A3CBEFF length: 93 byte(s) Send

QUERY PARAMETERS [1]

HEADERS <sup>Ⓞ</sup> Form

Content-Type : application/json

+ Add header Ⓞ Add authorization

BODY <sup>Ⓞ</sup> Text

```

1 {
2   "name": "Media Test",
3   "version": "1.0",
4   "description": "Upload an image",
5   "components": {
6     "Image": {
7       "type": "MEDIA",
8       "link": "https://drive.google.com/file/1...
9     }
10  }
11 }
12

```

#### Response

200 OK Cache Detected - Elapsed Time: 311ms

HEADERS <sup>Ⓞ</sup> pretty

Access-Control-Allow-\*

Cache-Control: no-cache

Pragma: no-cache

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Server: Jetty(1-jetty 2.0.0-78+bundle)

COMPLETE REQUEST HEADERS

BODY <sup>Ⓞ</sup> pretty

```

{
  name: "Media Test",
  version: "1.0",
  description: {
    default: "Upload an image"
  },
  components: {
    Image: {
      type: "MEDIA",
      link: "https://drive.google.com/file/1...
    }
  },
  state: "SYNCING",
  details: {
    firstInstallTime: "2020-11-01T08:07:39Z",
    lastModifyTime: "2020-11-01T08:07:39Z",
    componentStatus: {
      Image: {
        state: "PENDING"
      }
    }
  }
}

```

## Developer Guide

- App Solutions- POST /v2/app/solutions/query
  - **Description:** Verify App install status
  - **Input:** JSON

Parameter	Function
name	App name, string

- **Output:** JSON

Parameter	Function
name	App name, string
version	Version number, string
description	App description, string
components	Component-key is a string for this component <pre>{   "component-key":{     "type":"APK",     "packageName":"string",     "versionCode":"number   decimal-string",     "versionName":"string",     "link":"string",     "allowUninstall":"boolean"   } }</pre>
state	Possible values: “ <b>SYNCING</b> ”, “ <b>READY</b> ”, “ <b>ERROR</b> ”, “ <b>UNREADY</b> ”

## Developer Guide

**detail**

```
"details":{
  "firstInstallTime":iso8601-datetime-string,
  "lastModifyTime":iso8601-datetime-string,
  "lastSyncTime":iso8601-datetime-string,
  "signingInfo":{
    "...
  },
  "componentStatus":{
    "component-key":{
      "state":"string",
      "progress":"number",
      "contentUri":"string"
    },
    "...
  }
}
```

## Developer Guide

### ➤ Practical: Query installation status

solutions/query

METHOD: POST | SCHEME // HOST [ : PORT ] [ PATH [ ? QUERY ] ]

http://172.20.10.12:8080/v2/app/solutions/sync?access\_token=19B602DAFBD6099AC98D646D8F6D0AF

QUERY PARAMETERS [1]

HEADERS: Content-Type: application/json

BODY: { "name": "GoBright" }

---

200 OK

HEADERS: Access-Control-Allow-Origin: \*, Cache-Control: no-cache, Pragma: no-cache, Content-Type: application/json; charset=utf-8, Transfer-Encoding: chunked, Server: Jetty(i-jetty 2.0.0-78+bundle)

BODY: { name: "GoBright", version: "9.2", description: { default: "GoBright Solution" }, components: { GoBrightAPK: { type: "APK", packageName: "com.goBright.apk", display: "GoBright" } }, state: "SYNCING", details: { firstInstallTime: "2020-10-14T14:22:26Z", lastModifyTime: "2020-10-14T14:22:26Z", lastSyncTime: "2020-10-14T14:22:36Z", componentStatus: { GoBrightAPK: { state: "PENDING" } } } } }

## Developer Guide

- App Solutions- POST /v2/app/solutions/update
  - **Description:** Update an installed app.
  - **Input:** JSON

Parameter	Function
<b>name</b>	App name, string
<b>version</b>	Version number, string
<b>description</b>	App description, string
<b>components</b>	Component-key is a string for this component <pre> {   "component-key":{     "type":"APK",     "packageName":"string",     "versionCode":"number   decimal-string",     "versionName":"string",     "link":"string",     "allowUninstall":"boolean"   } }           </pre>

➤ **Output: JSON**

Parameter	Function
<b>name</b>	App name, string
<b>version</b>	Version number, string
<b>description</b>	App description, string
<b>components</b>	Component-key is a string for this component <pre>{   "component-key":{     "type":"APK",     "packageName":"string",     "versionCode":"number   decimal-string",     "versionName":"string",     "link":"string",     "allowUninstall":"boolean"   } }</pre>
<b>state</b>	Possible values: <b>“SYNCING”</b> , <b>“READY”</b> , <b>“ERROR”</b> , <b>“UNREADY”</b>
<b>detail</b>	<pre>"details":{   "firstInstallTime":iso8601-datetime-string,   "lastModifyTime":iso8601-datetime-string,   "lastSyncTime":iso8601-datetime-string,   "signingInfo":{</pre>

## Developer Guide

```
"..."
},
"componentStatus":{
  "component-key":{
    "state":"string",
    "progress":"number",
    "contentUri":"string"
  },
  "..."
}
}
```

## Developer Guide

- **Practical:** Update an installed app.

**solutions/update**

METHOD: POST  
 SCHEME://HOST [":" PORT] [PATH [ "?" QUERY ]]

http://172.20.10.12:8080/v2/app/solutions/update?access\_token=19B602DAFBD6099AC98D646D8F6D0AF  
length: 93 byte(s)

▶ QUERY PARAMETERS [1]

HEADERS [1] | Form

Content-Type : application/json

+ Add header | Add authorization

BODY | Text

```
1 {
2   "name": "my-app"
3 }
4
```

Text JSON XML HTML | Format body |  Enable body evaluation  
length: 25 bytes

**200 OK**

HEADERS | pretty

```
access-control-allow-origins: *
cache-control: no-cache
content-type: application/json; charset=utf-8
pragma: no-cache
server: Jetty(i-jetty 2.0.0-78+bundle)
transfer-encoding: chunked
```

BODY | pretty

```
{
  name: "my-app",
  state: "SYNCING",
  details: {
    firstInstallTime: "2020-10-14T14:22:26Z",
    lastModifyTime: "2020-10-14T14:30:25Z",
    lastSyncTime: "2020-10-14T14:23:23Z"
  }
}
```

lines nums length: 169 bytes



## Developer Guide

- App Solutions- POST /v2/app/solutions/delete
  - **Description:** Verify App install status
  - **Input:** JSON

Parameter	Function
name	App name, string

- **Output:** JSON

Parameter	Function
null	null

## Developer Guide

### ➤ Practical: Delete an app.

**solutions/delete**

METHOD: **POST** | SCHEME://HOST [:" PORT ] [ PATH [ "?" QUERY ] ]

URL: `http://172.20.10.12:8080/v2/app/solutions/delete?access_token=19B602DAFBD6099AC98D646D8F6D0AF` length: 93 byte(s)

QUERY PARAMETERS [1]

HEADERS <sup>ⓘ</sup> ↓ **Form**

- Content-Type : `application/json`

+ Add header | Add authorization

BODY <sup>ⓘ</sup> **Text**

```

1 {
2   "name": "XXXXXXXXXX"
3 }
4

```

Text | JSON | XML | HTML | Format body |  Enable body evaluation | length: 25 bytes

**200 OK**

HEADERS <sup>ⓘ</sup> **pretty**

```

access-control-allow-headers: *
cache-control: no-cache
content-type: application/json; charset=utf-8
pragma: no-cache
server: Jetty(i-jetty 2.0.0-78+bundle)
transfer-encoding: chunked

```

BODY <sup>ⓘ</sup> **pretty**

```

null

```

[lines nums](#) | length: 4 bytes

## Developer Guide

- App Solutions- POST /v2/app/solutions/sync
  - **Description:** Sync an installed app.
  - **Input:** JSON

Parameter	Function
name	App name, string

- **Output:** JSON

Parameter	Function
name	App name, string
version	Version number, string
description	App description, string
components	Component-key is a string for this component <pre> {   "component-key":{     "type":"APK",     "packageName":"string",     "versionCode":"number   decimal-string",     "versionName":"string",     "link":"string",     "allowUninstall":"boolean"   } }           </pre>

## Developer Guide

<b>state</b>	Possible values: "SYNCING", "READY", "ERROR", "UNREADY"
<b>detail</b>	<pre> "details":{   "firstInstallTime":iso8601-datetime-string,   "lastModifyTime":iso8601-datetime-string,   "lastSyncTime":iso8601-datetime-string,   "signingInfo":{     "..."/&gt; </pre>

## Developer Guide

### ➤ Practical: Sync an installed app

**solutions/sync**

METHOD:  SCHEME://HOST [:" PORT ][ PATH [ "?" QUERY ]]

length: 93 byte(s)

▶ QUERY PARAMETERS [1]

HEADERS <sup>Ⓢ</sup> ↓

Content-Type  ×

+ Add header Add authorization

BODY <sup>Ⓢ</sup>

```

1 {
2   "name": "xxxxxxxxxxxx"
3 }
4

```

Text JSON XML HTML | Format body |  Enable body evaluation length: 25 bytes

---

**200 OK** ✕

HEADERS <sup>Ⓢ</sup>

Access-Control-Allow... \*

Cache-Control: no-cache

Pragma: no-cache

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Server: Jetty(i-jetty 2.0.0-78+bundle)

BODY <sup>Ⓢ</sup>

null

lines nums length: 4 bytes

## Secure Access with HTTPS

- For enhanced security, access the REST API through HTTPS.

Unlike HTTP (port 8080) HTTPS requires:

- **Protocol:** HTTPS
- **Port:** 8443
- **Host:** localhost (or your client device must trust the device certificate)
- **Query:** <https://localhost:8443/v2/> appended by API call.

## Bypass Password During Authentication

- Using `window.adapi.getToken()` enables password-free command execution, even on password-protected devices.

### Enabling the required settings:

- To use `window.adapi.getToken()`, enable the [Trust] option in AppStart's advanced settings.
  1. On the IAdea device, go to the [Basic Settings] page.
  2. Click [Advanced] > [System] > [AppStart].
  3. Expand options by clicking [Advanced].
  4. Enable the [Trust] option and click [Set].

Example: [Playing default content](#) via JavaScript Console:

```
// Get the token
var token = window.adapi.getToken(); // Retrieve the token

// Make the POST request
fetch('https://localhost:8443/v2/app/switch', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'Authorization': `Bearer ${token}` // Append the token to the
Authorization header
  },
  body: JSON.stringify({
    "mode": "start" // Body content
  })
})
.then(response => response.json()) // Parse the JSON response
.then(data => console.log(data)) // Log the response data
.catch(error => console.error('Error:', error)); // Handle any errors
(optional)
```

**Notes:** For HTTP, replace the URL with <http://localhost:8080/v2/app/switch>

### Content Security Policy

- **As a developer you can specify the Content Security Policy through a HTTP response header called Content-Security-Policy.**
- **To allow inline scripts and all scripts loaded from IAdea players, please whitelist <http://localhost:8080> to your web application.**
- **For further information about Content Security Policy, please refer to the page below. <https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>**