

## My.jdownloader.org API Docu & implementation guidelines

### Introduction

This document specifies the development state of the MyJDownloader API. As the API changes regularly, changes in this document will happen every so often.

The API is designed to offer a secure communication between the JDownloader client and the request client, and to prevent any man in the middle listeners.

For this approach the API is using AES128CBC and HMAC-SHA256 for encryption. To be able to communicate with the API we would suggest to make sure you understand the procedure as this can get somewhat complicated for beginners.

The API is REST based, but currently only GET and POST routes are offered.

### Basics

Every JDownloader client configured to connect to the My.JDownloader service will register itself to the API.

There are two different main routes available.

1. Server API
2. Device API

The server API handles device registrations, account creation, account modification, password resets and so on. Furthermore it handles the handshake between the request client and the JDownloader client.

The Device API offers the routes you need to get data from the JDownloader client.

This includes:

1. Package/Link data
2. LinkGrabber data
3. Captcha data
4. Events
5. And much more

So in order to get data from a JDownloader client, you will need to do a handshake with the Server API first.

This documentation will guide you to be able to do said handshake, and to be able to gather data from a JDownloader client.

## Technical Specification

Api host: api.jdownloader.org

Api ports: 80, 10101

Currently no https is available.

There are some important things you need to keep in mind:

### Parameter Order

The parameter in GET/POST requests are **important**.

Use the documented order or you won't get a positive response.

### Content-Type Header

Make sure to use the correct Content-Type.

For most calls, this is application/json; charset=utf-8, see call description

### Always provide a new RequestID

The RequestID is required in almost every request.

It's a number that has to increase from one call to another.

You can either use a millisecond precise timestamp, or a self incrementing number.

The API will return the RequestID in the response.

**You should validate the response to make sure the answer is valid.**

### Encoding

Make sure all of url parameters are correctly urlencoded.

## Signature

Some calls use a signature parameter. To create a Signature, you have to

1. build the full queryString (incl. Rid)
2. hmac the queryString. The used Key depends. Some calls use serverEncryptionToken, others have to ask the user for email and password, create the loginSecret and use the loginsecret as key. email needs to be lower case!
3. hexformat the result
4. append the signature to the queryString &signature=...

Example:

```
queryString = "/my/connect?email=foo@bar.com&rid=1361982773157";  
queryString += "&signature=" + HmacSha256(utf8bytes(queryString), ServerEncryptionToken);
```

## Errors & Exceptions

All HTTP Responsecodes except 200 are errors or exceptions.

```
{  
  "src": "MYJD" | "DEVICE"  
  "type": <see errortypes below>  
  "data": <jsonobject>  
}
```

### Error Object Fields

Fieldname	Description and Values						
src							
	<table border="1"><thead><tr><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>MYJD</td><td>Error occured in the connection server</td></tr><tr><td>DEVICE</td><td>error occured on the remote device. e.g. a running jD</td></tr></tbody></table>	Value	Description	MYJD	Error occured in the connection server	DEVICE	error occured on the remote device. e.g. a running jD
	Value	Description					
	MYJD	Error occured in the connection server					
DEVICE	error occured on the remote device. e.g. a running jD						
type	@See ErrorTypes below						
data	Optional data or null						

### ErrorTypes:

src	type	Http Code	data	explain
-----	------	-----------	------	---------

MYJ D	OVERLOAD	503	null	Server overloaded. Try later
MYJ D	ERROR_EMAIL_NOT_CONFIRMED	401	null	Click Confirml Email, or resend email
MYJ D	TOKEN_INVALID	403	null	reconnect
MYJ D	OFFLINE	504	null	Device is not available
MYJ D	UNKNOWN			anything
MYJ D	AUTH_FAILED	403	null	Auth Failed during connect
MYJ D	TOO_MANY_REQUESTS	429	null	too many requests. Unlock via challenge
MYJ D	EMAIL_INVALID			email syntax error
MYJ D	CHALLENGE_FAILED			bad captcha response
MYJ D	EMAIL_FORBIDDEN			email probably exists
MYJ D	FAILED			the called action failed
MYJ D	MAINTENANCE	503	null	Server Maintenance. e.g. Rebooting

## Account Management

### Signup for a New Account

#### 1. Get Challenge

GET [api.jdownloader.org/captcha/getCaptcha](http://api.jdownloader.org/captcha/getCaptcha)

Response Content

```
{
  "captchaChallenge" : "13650058317(...)8299d97cf5",
  "image" : "(...)"
}
```

Possible Errors

OVERLOAD,MAINTENANCE,TOO\_MANY\_REQUESTS

## 2. Register

POST

```
/my/requestregistrationemail?email=<email>&captchaResponse=<captchacode>&captchaChallenge=<captchaChallenge>&referrer=<Your RefererID - or AppKey>
```

-No Postdata-

Response Content

true

Possible Errors

OVERLOAD,MAINTENANCE,TOO\_MANY\_REQUESTS,EMAIL\_INVALID,CHALLENGE\_FAILED,EMAIL\_FORBIDDEN

## 3. Finish Registration

After Step 2, the user will get an Email. This email contains the registerKeyString. registerKeyString has a length of 64 and is hexformatted. Unhex this registerKeyString and you get a 32 byte registerKey

Common Mistakes:

- Make sure that you don't use urlencode für loginSecret
- Make sure to always submit **the email adress in lower case!**

```
a) loginSecret=hex(sha256( utf8bytes( email.lowercase + pass + "server" )));
```

```
b) encryptedLoginSecret= hex(AES128CBC(loginSecret))
```

```
iv=registerKey.firsthalf
```

```
key=registerKey.secondhalf
```

```
/my/finishregistration?email=<email>&loginsecret=<encryptedLoginSecret>&signature=<Signature>
```

The signature is a bit different in this case, because the used key is the registrationKey

```
queryString=/my/finishregistration?email=<email>&encryptedLoginSecret=<encryptedLoginSecret>
```

```
signature= HmacSha256(utf8bytes(queryString),registerKey);
```

Response:

```
Encryption: AES128CBC iv=registerKey.firstHalf, key=registerKey.secondHalf
```

```
Decrypted Content:
```

```
{
```

```
    "rid": 1361963793949 //VALIDATE @See RequestID
```

```
}
```

## Connect

This call has to ask the user for email+password

Common Mistakes:

- Make sure that you don't use urlencode für loginSecret
- Make sure to always submit the email adress in lower case!
- The Signature is calculated with the loginSecret for this call.

```
loginSecret=hex(sha256( utf8bytes( email.lowercase + pass + "server" )));  
deviceSecret=hex(sha256( utf8bytes( email.lowercase + pass + "device" )));
```

Encryption Key for Signature and Response Encryption: loginSecret

Path: /my/connect?email=<email>&appkey=<appkey>&rid=<@See  
RequestID>&signature=HmacSha256(utf8bytes(queryString),loginSecret)

-No Postdata-

Response:

Encryption: AES128CBC iv=loginSecret.firstHalf, key=loginSecret.secondHalf

Decrypted Content:

```
{  
  "sessiontoken": "af42...",  
  "regaintoken": "ca35...",  
  "rid": 1361963793949 //VALIDATE @See RequestID  
}
```

**Do NOT Store the loginSecret but calculate:**

Common Mistakes:

- In Javascript: loginSecret, deviceSecret and sessionToken are byte/word arrays. you cannot use + to merge them!

```
serverEncryptionToken=sha256(loginSecret+sessiontoken);
```

```
deviceEncryptionToken=sha256(deviceSecret+sessiontoken);
```

ServerEncryptionToken, deviceEncryptionToken, loginSecret and deviceSecret have 32 bytes. The are used for aes encryption. In this case byte 0-15 are used as initvector and the second half 16-31 are used as key.

After the Connect Process, you have the following fields:

<b>serverEncryptionToken</b>	Server<-->Client Encryption (iv=token.firstHalf key=firstHalf.secondhalf)
<b>deviceEncryptionToken</b>	Server<-->Device Encryption (iv=token.firstHalf key=firstHalf.secondhalf)
<b>sessiontoken</b>	Sessiontoken for all further calls
<b>regaintoken</b>	Token to get a New sessiontoken if the old one is invalid
<b>deviceSecret</b>	Needed to generate new deviceEncryptionToken in case of reconnect (@see reconnect)

Possible Errors:

OVERLOAD,MAINTENANCE,TOO\_MANY\_REQUESTS,AUTH\_FAILED,ERROR\_EMAIL\_NOT\_CONFIRMED

The Token is used as auth token for all further requests. the regain token can be used once to get a new token (@see reconnect). Applications should not store email and password, but token and regaintoken only. If token AND regaintoken are invalid or outdated, ask the user to enter his logins.

## RECONNECT

### Parameters

sessiontoken	sessiontoken (@see connect)
regaintoken	regaintoken (@see connect)
rid	@see timestamp above
signature	@see Signature above

Path: /my/reconnect?sessiontoken=<sessiontoken>&regaintoken=<regaintoken>&rid=<@See RequestID>&signature=<@See Signature>

Response:

Encryption: AES128CBC iv=serverEncryptionToken.firstHalf,

key=serverEncryptionToken.secondHalf

Decrypted Content:

```
{
  "sessiontoken": "af42...",
  "regainToken": "ca35...",
  "rid": 1361963793949 //VALIDATE @See RequestID
}
```

### Common Mistakes:

- In Javascript: serverEncryptionToken, deviceSecret and sessionToken are byte/word arrays. you cannot use + to merge them!

```
serverEncryptionToken=sha256(serverEncryptionToken+sessiontoken);
```

```
deviceEncryptionToken=sha256(deviceSecret+sessiontoken);
```

Possible Errors:

OVERLOAD,MAINTENANCE,TOO\_MANY\_REQUESTS,AUTH\_FAILED,ERROR\_EMAIL\_NOT\_CONFIRMED

## **Request Passwordchange Email Confirmation**

This call has no signature!

Path:

/my/requestpasswordresetemail?email=<email>&captchaResponse=<captchacode>&captchaChallenge=<captchaChallenge>

-No Postdata-

Response Content

true

Possible Errors

OVERLOAD,MAINTENANCE,TOO\_MANY\_REQUESTS,EMAIL\_INVALID,CHALLENGE\_FAILED,EMAIL\_FORBIDDEN

## **Change Password**

### 1. Request a Password Change (@See above)

The user will get an email containing a key. YOU need this key for the next Step.

### 2. Do a password change

After Step 1, the user will get an Email. This email contains the passwordchangeKeyString. passwordchangeKeyString has a length of 64 and is hexformatted. Unhex this passwordchangeKeyString and you get a 32 byte passwordchangeKey

Common Mistakes:

- Make sure that you don't use urlencode für loginSecret

a) loginSecret=hex(sha256( utf8bytes( email.lowercase + newpass + "server" )));

b) encryptedLoginSecret= hex(AES128CBC(loginSecret))

iv=passwordchangeKey.firsthalf

key=passwordchangeKey.secondhalf

/my/finishpasswordreset?email=<email>&loginsecret=<encryptedLoginSecret>&signature=<Signature>

The signature is a bit different in this case, because the used key is the passwordchangeKey

queryString=/my/finishregistration?email=<email>&encryptedLoginSecret=<encryptedLoginSecret>

signature= HmacSha256(utf8bytes(queryString),passwordchangeKey);

Response:

Encryption: AES128CBC iv=passwordchangeKey.firstHalf, key=passwordchangeKey.secondHalf

Decrypted Content:

{

    "rid": 1361963793949 //VALIDATE @See RequestID

```
}
```

After a password change, the current Session becomes invalid. Do a full connect to get a new session.

## Resend Email Confirmation

Use Step 2. Register /my/requestregisteremail....

## List Devices

Encryption Key for Signature and Response Encryption: serverEncryptionToken

```
POST api.jdownloader.org/my/listdevices?sessiontoken=<sessionToken>&rid=<@See
RequestID>&signature=<@see signature>
```

Response Encryption:

```
aes128cbc(iv=serverEncryptionToken.firstHalf, key=serverEncryptionToken.secondHalf )
```

Decrypted Response:

```
{
  "rid": 1361963793949 //Timestamp des Requests.
  "list": [
    {
      "id": "uniquedeviceid", //given by the server
      "type": "devicetype", //given by the device
      "name": "My Device Name" //given by the device
      ...//other fields are possible
    },...
  ]
}
```

Possible Errors:

```
OVERLOAD,MAINTENANCE,TOO_MANY_REQUESTS,TOKEN_INVALID
```

## Disconnect

```
POST api.jdownloader.org/my/disconnect?sessiontoken=<sessionToken>&rid=<@See
RequestID>&signature=<@see signature>
```

Response Encryption:

```
aes128cbc(iv=secretServer.firstHalf, key=secretServer.secondHalf )
```

Decrypted Response:

```
{
    "rid": 1361963793949 //Timestamp des Requests.
}
```

Possible Errors:

OVERLOAD,MAINTENANCE,TOO\_MANY\_REQUESTS,TOKEN\_INVALID

## General API Calls

Encryption Key for Signature and Response Encryption: deviceEncryptionToken

POST api.jdownloader.org/t\_<sessiontoken>\_<deviceid>/action

Content-Type: application/aesjson-jd; charset=utf-8

```
Request Content: aes128cbc(data, iv=deviceEncryptionToken.firstHalf,
                           key=deviceEncryptionToken.secondHalf )
```

NOTE: data is just a variable for this document. The actual content is JSON {"apiVer": 1..}

data =

```
{
    "apiVer": 1, //1 is the current API-Version
    "url": "/action",
    "params": [
        "param1",
        "param2"
    ],
    "rid": timestamp_milliseconds
}
```

HTTP/1.1 200 OK

Content-Type: application/aesjson-jd; charset=utf-8

```
Response Content: aes128cbc(data, iv=deviceEncryptionToken.firstHalf,
                           key=deviceEncryptionToken.secondHalf )
```

Die URL wird doppelt transmittiert, da der ConnectServer PremiumURLs filtern können soll. JD muss daher überprüfen ob data.url der Action in der URL entspricht. Sollte der Server eine Fehlermeldung kommunizieren wollen, antwortet er mit

HTTP/1.1 4xx/5xx ERRORMSG

Content-Type: application/aesjson-server; charset=utf-8

```
Response Content: aes128cbc( data, iv=deviceEncryptionToken.firstHalf,
                           key=deviceEncryptionToken.secondHalf )
```

Possible HTTP ErrorCodes:

503 Overload  
407 Token Invalid  
g  
429 Too Many Requests. @see #unlock

### API Data Transport

Set Request *Accept-Encoding* Header to “gzip\_aes” to bypass base64 (needed for browser compatibility) and compressed response content.

```
Check Response Code == 200 {  
Check Response Content-Encoding Header for “gzip_aes”:  
true: you can read the inputStream as aesinputstream(gzipinputstream(inputStream))  
false: you can read the inputStream as base64inputStream(aesinputstream(inputStream))  
}else{  
read the inputStream normal  
}
```

## EVENTS API - Server Push/Long Polling

### **Namespace: //events**

This API enables you to get informed by Events thrown by the JDownloader.

*Subscription Lifecycle:*

### **Start Subscription**

Call: subscribe



### **Listen to Events**

Call: listen



### **End Subscription**

Call: unsubscribe



### **ManageSubscription**

Calls: addSubscription, removeSubscription, setSubscription

//TODO: Unuglify graphic ;)

---

## START SUBSCRIPTION

Call	Request/Response	Description
/events/subscribe	<p><b>Params (POST):</b></p> <pre>{[   String[]:SUBSCRIPTIONS,   String[]:EXCLUSIONS ]}</pre> <p><b>Response</b> @see SubscriptionResponse</p>	<p>Starts new subscription and returns SubscriptionResponse containing the subscription Id used for further interaction with this subscription.</p> <p><b>SUBSCRIPTIONS</b> Array of Strings used as Regex to query for events.</p> <p><b>EXCLUSIONS</b> Array of Strings used as Regex to enable to further narrow down the subscribed events. Optional.</p>

## MANAGE SUBSCRIPTION

Call	Request/Response	Description
/events/addSubscription	<p><b>Params (POST):</b></p> <pre>{[   Long:SUBSCRIPTION_ID,   String[]:SUBSCRIPTIONS,   String[]:EXCLUSIONS ]}</pre> <p><b>Response</b> @see SubscriptionResponse</p>	<p>Add Subscriptions and/or Exclusions to the subscription with the given id.</p> <p><b>SUBSCRIPTION_ID</b> id that identified the subscription the call should be applied to</p> <p><b>SUBSCRIPTIONS</b> Array of Strings used as Regex to query for events.</p> <p><b>EXCLUSIONS</b> Array of Strings used as Regex to enable to further narrow down the subscribed events. Optional.</p>

<p><b>/events/removeSubscription</b></p>	<p><b>Params (POST) :</b></p> <pre>{   Long:SUBSCRIPTION_ID,   String[]:SUBSCRIPTIONS,   String[]:EXCLUSIONS }</pre> <p><b>Response</b> @see SubscriptionResponse</p>	<p>Remove Subscriptions and/or Exclusions to the subscription with the given id.</p> <p><b>SUBSCRIPTION_ID</b> id that identified the subscription the call should be applied to</p> <p><b>SUBSCRIPTIONS</b> Array of Strings used as Regex to query for events.</p> <p><b>EXCLUSIONS</b> Array of Strings used as Regex to enable to further narrow down the subscribed events. Optional.</p>
<p><b>/events/setSubscription</b></p>	<p><b>Params (POST) :</b></p> <pre>{   Long:SUBSCRIPTION_ID,   String[]:SUBSCRIPTIONS,   String[]:EXCLUSIONS }</pre> <p><b>Response</b> @see SubscriptionResponse</p>	<p>Set Subscriptions and/or Exclusions to the subscription with the given id.</p> <p><b>SUBSCRIPTION_ID</b> id that identified the subscription the call should be applied to</p> <p><b>SUBSCRIPTIONS</b> Array of Strings used as Regex to query for events.</p> <p><b>EXCLUSIONS</b> Array of Strings used as Regex to enable to further narrow down the subscribed events. Optional.</p>

<p><b>/events/changesubscriptiontimeouts</b></p>	<p><b>Params (POST) :</b>  <pre>{   Long:SUBSCRIPTION_ID,   Long:POLL_TIME_OUT,   Long:MAX_KEEP_ALIVE }</pre></p> <p><b>Response</b>  @see SubscriptionResponse</p>	<p>Set Subscriptions and/or Exclusions to the subscription with the given id.</p> <p><b>SUBSCRIPTION_ID</b>  id that identified the subscription the call should be applied to</p> <p><b>SUBSCRIPTIONS</b>  Array of Strings used as Regex to query for events.</p> <p><b>EXCLUSIONS</b>  Array of Strings used as Regex to enable to further narrow down the subscribed events.  Optional.</p>
--	---	---

**LISTEN TO THE SUBSCRIPTION**

Call	Request/Response	Description
<p><b>/events/listen</b></p>	<p><b>Params (POST) :</b>  <pre>{   Long:SUBSCRIPTION_ID }</pre></p> <p><b>Response</b>  @see EventObject[]</p>	<p>This call returns an array of EventObject for all events that occurred since the last call of "listen".If no events are available the call is kept alive until events arrive or the configurable timeout period is reached (Default: 30000ms). On a timeout an empty array is returned.</p> <p><b>SUBSCRIPTION_ID</b>  id that identified the subscription the call should be applied to</p>

**Note:** When reconnecting while "listen" is running, keep the old encryption token to decrypt the last listen fired with the **old device encryption token. To correctly implement the events API, keep your old device encryption token in case of a reconnect!**

## STOP SUBSCRIPTION

Call	Request/Response	Description
<b>/events/unsubscribe</b>	<b>Params (POST) :</b> { Long: <b>SUBSCRIPTION_ID</b> }  <b>Response</b> @see SubscriptionResponse	End the subscription  <b>SUBSCRIPTION_ID</b> id that identified the subscription the call should be applied to

## GENERAL CALLS

Call	Request/Response	Description
<b>/events/getsubscription</b>	<b>Params (POST) :</b> { Long: <b>SUBSCRIPTION_ID</b> }  <b>Response</b> @see SubscriptionResponse	Get SubscriptionResponse for subscription.  <b>SUBSCRIPTION_ID</b> id that identified the subscription the call should be applied to.

<p><b>/events/getsubscriptionStatus</b></p>	<p><b>Params (POST):</b>  <pre>{   Long: SUBSCRIPTION_ID }</pre></p> <p><b>Response</b>  @see SubscriptionStatusResponse</p>	<p>Get SubscriptionStatusResponse for subscription</p> <p><b>SUBSCRIPTION_ID</b>  id that identified the subscription the call should be applied to</p>
<p><b>/events/changesubscriptiontimeouts</b></p>	<p><b>Params (POST):</b>  <pre>{   Long: SUBSCRIPTION_ID,   Long: POLL_TIME_OUT,   Long: MAX_KEEP_ALIVE }</pre></p> <p><b>Response</b>  @see SubscriptionResponse</p>	<p><b>SUBSCRIPTION_ID</b>  id that identified the subscription the call should be applied to</p> <p><b>POLL_TIME_OUT</b>  in how many ms should a "listen" call time out</p> <p><b>MAX_KEEP_ALIVE</b>  in how many ms should the whole subscription time out</p>
<p><b>/events/listpublisher</b></p>	<p><b>Params (POST):</b>  -</p> <p><b>Response</b>  @see PublisherResponse[]</p>	<p>Get a list of all available Event Publishers</p>

### Currently Available Event Publisher Names:

“captchas”  
“downloads”  
“linkcollector”  
“downloadwatchdog”

Use “\*EVENT\_PUBLISHER\_NAME” to get all events from that publisher for example

### EVENT API RESPONSE OBJECTS:

#### SubscriptionResponse:

// Meta Data of the Subscription

```
{
  "subscriptionid": Number,
  "maxkeepalive": Number,
  "maxpolltimeout": Number,
  "subscriptions": String[],
  "exclusions": String[]
}
```

// Representation of the event

#### EventObject

```
{
  "eventid": String, //e.g. "STRUCTURE_CHANGED"
  "publisher": String //e.g. "downloads"
  "eventdata": Object, //can be used by event publisher to send additional data
}
```

#### SubscriptionStatusResponse

// Status of the subscription

```
{
  "subscriptionid": Number,
  "queuesize": Number //number of events waiting in queue
}
```

//can be used by event publisher to send additional data

#### PublisherResponse

```
{
  "publisher": String //can be used by event publisher to send additional data
  "eventids": String[], //all events the publisher can throw
}
```

# JDOWNLOADER-API

## /config

Call	Params (POST unencrypted)	Description
/config/get	{ String:INTERFACE_NAME, String:STORAGE, String:KEY }	<b>INTERFACE_NAME:</b> Name of the interface containing the desired value, including package path. <b>STORAGE:</b> In the case of the interface being represented by multiple storage objects, set this variable, Can be left "null" by default. <b>KEY:</b> The key of the config entry
/config/set	{ String:INTERFACE_NAME, String:STORAGE, String:KEY, Object:VALUE }	<b>INTERFACE_NAME:</b> Name of the interface containing the desired value, including package path. <b>STORAGE:</b> In the case of the interface being represented by multiple storage objects, set this variable, Can be left "null" by default. <b>KEY:</b> The key of the config entry <b>VALUE:</b> The new value the config entry should have, depends on the datatype of the config entry
/config/list	{ String:PATTERN, Boolean:RETURN_DESCRIPTION, Boolean:RETURN_VALUES, Boolean:RETURN_DEFAULT_VALUES }	<b>PATTERN:</b> Regex String to search for all the interface names, that should be queried <b>RETURN_DESCRIPTION</b> Should a description be returned? <b>RETURN_VALUES</b> Should the current values be returned? <b>RETURN_DEFAULT_VALUES</b> Should the default values be returned?

====- help -====

Description: This Call

Call: /help

====- list -====

Description: list all available config entries

Call: /list

====- list -====

Description: list entries based on the pattern regex

Parameter: 1 - String-pattern

Parameter: 2 - boolean-returnDescription

Parameter: 3 - boolean-returnValues

Parameter: 4 - boolean-returnDefaultValues

Parameter: 5 - boolean-returnEnumInfo

Call: /list?String1&boolean1&boolean2&boolean3&boolean4

====- list -====

Description: DEPRECATED! list entries based on the pattern regex

Parameter: 1 - String-pattern

Parameter: 2 - boolean-returnDescription

Parameter: 3 - boolean-returnValues

Parameter: 4 - boolean-returnDefaultValues

Call: /list?String1&boolean1&boolean2&boolean3

====- set -====

Description: set value to interface by key

Parameter: 1 - String-interfaceName

Parameter: 2 - String-storage

Parameter: 3 - String-key

Parameter: 4 - Object-value

Call: /set?String1&String2&String3&Object1

====- getDefault -====

Description: get default value from interface by key

Parameter: 1 - String-interfaceName

Parameter: 2 - String-storage

Parameter: 3 - String-key

Call: /getDefault?String1&String2&String3

====- reset -====

Description: reset interface by key to its default value

Parameter: 1 - String-interfaceName

Parameter: 2 - String-storage

Parameter: 3 - String-key

Call: /reset?String1&String2&String3

====- listEnum -====

Description: list all possible enum values

Parameter: 1 - String-type

Call: /listEnum?String1

====- get -====

Description: get value from interface by key

Parameter: 1 - String-interfaceName

Parameter: 2 - String-storage

Parameter: 3 - String-key

Call: /get?String1&String2&String3

====- query -====

Parameter: 1 - AdvancedConfigQueryStorable-query  
Call: /query?AdvancedConfigQueryStorable1

## **/linkgrabberv2**

**org.jdownloader.api.linkcollector.v2.LinkCollectorAPIV2**

====- clearList -====

Call: /clearList

====- moveToDownloadlist -====

Parameter: 1 - long[]-linkIds  
Parameter: 2 - long[]-packageIds  
Call: /moveToDownloadlist?long[]1&long[]2

====- queryLinks -====

Parameter: 1 - CrawledLinkQueryStorable-queryParams  
Call: /queryLinks?CrawledLinkQueryStorable1

====- cleanup -====

Parameter: 1 - long[]-linkIds  
Parameter: 2 - long[]-packageIds  
Parameter: 3 - Action-action  
Parameter: 4 - Mode-mode  
Parameter: 5 - SelectionType-selectionType  
Call: /cleanup?long[]1&long[]2&Action1&Mode1&SelectionType1

====- addContainer -====

Parameter: 1 - String-type  
Parameter: 2 - String-content  
Call: /addContainer?String1&String2

====- getDownloadUrls -====

Parameter: 1 - long[]-linkIds  
Parameter: 2 - long[]-packageIds  
Parameter: 3 - UriDisplayTypeStorable[]-urlDisplayTypes  
Call: /getDownloadUrls?long[]1&long[]2&UriDisplayTypeStorable[]1

====- setPriority -====

Parameter: 1 - PriorityStorable-priority  
Parameter: 2 - long[]-linkIds  
Parameter: 3 - long[]-packageIds  
Call: /setPriority?PriorityStorable1&long[]1&long[]2

====- renameLink -====

Parameter: 1 - long-linkId  
Parameter: 2 - String-newName  
Call: /renameLink?long1&String1

====- setVariant -====

Parameter: 1 - long-linkid

Parameter: 2 - String-variantID

Call: /setVariant?long1&String1

====- getPackageCount -====

Call: /getPackageCount

====- renamePackage -====

Parameter: 1 - long-packageld

Parameter: 2 - String-newName

Call: /renamePackage?long1&String1

====- movePackages -====

Parameter: 1 - long[]-packagelds

Parameter: 2 - long-afterDestPackageld

Call: /movePackages?long[]1&long1

====- setEnabled -====

Parameter: 1 - boolean-enabled

Parameter: 2 - long[]-linkIds

Parameter: 3 - long[]-packagelds

Call: /setEnabled?boolean1&long[]1&long[]2

====- getVariants -====

Parameter: 1 - long-linkid

Call: /getVariants?long1

====- addLinks -====

Parameter: 1 - AddLinksQueryStorable-query

Call: /addLinks?AddLinksQueryStorable1

====- getChildrenChanged -====

Parameter: 1 - long-structureWatermark

Call: /getChildrenChanged?long1

====- movetoNewPackage -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-pkgIds

Parameter: 3 - String-newPkgName

Parameter: 4 - String-downloadPath

Call: /movetoNewPackage?long[]1&long[]2&String1&String2

====- removeLinks -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packagelds

Call: /removeLinks?long[]1&long[]2

====- getDownloadFolderHistorySelectionBase -====

Call: /getDownloadFolderHistorySelectionBase

====- splitPackageByHoster -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-pkgIds

Call: /splitPackageByHoster?long[]1&long[]2

====- help -====

Description: This Call

Call: /help

====- setDownloadDirectory -====

Parameter: 1 - String-directory

Parameter: 2 - long[]-packageIds

Call: /setDownloadDirectory?String1&long[]1

====- startOnlineStatusCheck -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Call: /startOnlineStatusCheck?long[]1&long[]2

====- moveLinks -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long-afterLinkID

Parameter: 3 - long-destPackageID

Call: /moveLinks?long[]1&long1&long2

====- setDownloadPassword -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Parameter: 3 - String-pass

Call: /setDownloadPassword?long[]1&long[]2&String1

====- queryPackages -====

Parameter: 1 - CrawledPackageQueryStorable-queryParams

Call: /queryPackages?CrawledPackageQueryStorable1

====- addVariantCopy -====

Parameter: 1 - long-linkid

Parameter: 2 - long-destinationAfterLinkID

Parameter: 3 - long-destinationPackageID

Parameter: 4 - String-variantID

Call: /addVariantCopy?long1&long2&long3&String1

====- *LinkQueryStorable* - ====

LinkQueryStorable=

```
{  
  "bytesTotal" : false,
```

```
"comment" : false,
"status" : false,
"enabled" : false,
"maxResults" : -1,
"startAt" : 0,
"packageUUIDs" : null,
"host" : false,
"url" : false,
"bytesLoaded" : false,
"speed" : false,
"eta" : false,
"finished" : false,
"priority" : false,
"running" : false,
"skipped" : false,
"extractionStatus" : false
}
```

====- *AddLinksQueryStorable* - =====

AddLinksQueryStorable=

```
{
  "autostart": false,
  "deepDecrypt": false,
  "autoExtract": false,
  "overwritePackagizerRules": false,
  "links": null,
  "dataURLs": [],
  "packageName": null,
  "extractPassword": null,
  "sourceUrl": null,
  "downloadPassword": null,
  "destinationFolder": null,
  //true: store a Unique JOB id in all links, that can be used later to adress links resulting from the addLinks call
  "assignJobID": true|false,
}
```

====- *Enums* -=====

### **Mode**

-[REMOVE\_LINKS\_AND\_DELETE\_FILES, REMOVE\_LINKS\_AND\_RECYCLE\_FILES,  
REMOVE\_LINKS\_ONLY]

### **SelectionType**

- [SELECTED, UNSELECTED, ALL, NONE]

### **Action**

- [DELETE\_ALL, DELETE\_DISABLED, DELETE\_FAILED, DELETE\_FINISHED, DELETE\_OFFLINE, DELETE\_DUPE, DELETE\_MODE]

### **PriorityStorable**

- [HIGHEST, HIGHER, HIGH, DEFAULT, LOWER]

## **/update**

org.jdownloader.api.update.UpdateAPI

====- help -====

Description: This Call

Call: /help

====- restartAndUpdate -====

Call: /restartAndUpdate

====- runUpdateCheck -====

Call: /runUpdateCheck

====- isUpdateAvailable -====

Call: /isUpdateAvailable

## /accountsV2

org.jdownloader.api.accounts.v2.AccountAPIV2

====- help -====

Description: This Call

Call: /help

====- removeAccounts -====

Parameter: 1 - long[]-ids

Call: /removeAccounts?long[]1

====- disableAccounts -====

Parameter: 1 - long[]-ids

Call: /disableAccounts?long[]1

====- refreshAccounts -====

Parameter: 1 - long[]-ids

Call: /refreshAccounts?long[]1

====- listPremiumHoster -====

Call: /listPremiumHoster

====- getPremiumHosterUrl -====

Parameter: 1 - String-hoster

Call: /getPremiumHosterUrl?String1

====- addAccount -====

Parameter: 1 - String-premiumHoster

Parameter: 2 - String-username

Parameter: 3 - String-password

Call: /addAccount?String1&String2&String3

====- listPremiumHosterUrls -====

Call: /listPremiumHosterUrls

====- enableAccounts -====

Parameter: 1 - long[]-ids

Call: /enableAccounts?long[]1

====- setUserNameAndPassword -====

Parameter: 1 - long-accountId

Parameter: 2 - String-username

Parameter: 3 - String-password

Call: /setUserNameAndPassword?long1&String1&String2

====- listAccounts -====

Parameter: 1 - AccountQuery-query

Call: /listAccounts?AccountQuery1

## /contentV2

org.jdownloader.api.content.v2.ContentAPIV2

====- getIcon -====

Parameter: 1 - String-key

Parameter: 2 - int-size

Call: /getIcon&String1&int1

====- getIconDescription -====

Parameter: 1 - String-key

Call: /getIconDescription?String1

====- help -====

Description: This Call

Call: /help

====- getFileIcon -====

Parameter: 1 - String-filename

Call: /getFileIcon&String1

====- getFavIcon -====

Parameter: 1 - String-hostername

Call: /getFavIcon&String1

## /dialogs

org.jdownloader.api.dialog.DialogApiInterface

====- help -====

Description: This Call

Call: /help

====- list -====

Call: /list

====- answer -====

Parameter: 1 - long-id

Parameter: 2 - HashMap-data

Call: /answer?long1&HashMap1

====- getTypeInfo -====

Parameter: 1 - String-dialogType

Call: /getTypeInfo?String1

====- get -====

Parameter: 1 - long-id

Parameter: 2 - boolean-icon

Parameter: 3 - boolean-properties

Call: /get?long1&boolean1&boolean2

## /downloadsV2

org.jdownloader.api.downloads.v2.DownloadsAPIV2

====- setStopMark -====

Parameter: 1 - long-linkId

Parameter: 2 - long-packageId

Call: /setStopMark?long1&long2

====- queryLinks -====

Parameter: 1 - LinkQueryStorable-queryParams

Call: /queryLinks?LinkQueryStorable1

====- cleanup -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Parameter: 3 - Action-action

Parameter: 4 - Mode-mode

Parameter: 5 - SelectionType-selectionType

Call: /cleanup?long[]1&long[]2&Action1&Mode1&SelectionType1

====- getDownloadUrls -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Parameter: 3 - UriDisplayTypeStorable[]-urlDisplayType

Call: /getDownloadUrls?long[]1&long[]2&UriDisplayTypeStorable[]1

====- setPriority -====

Parameter: 1 - PriorityStorable-priority

Parameter: 2 - long[]-linkIds

Parameter: 3 - long[]-packageIds

Call: /setPriority?PriorityStorable1&long[]1&long[]2

====- renameLink -====

Parameter: 1 - Long-linkId

Parameter: 2 - String-newName

Call: /renameLink?Long1&String1

====- getStructureChangeCounter -====

Parameter: 1 - long-oldCounterValue

Call: /getStructureChangeCounter?long1

====- renamePackage -====

Parameter: 1 - Long-packageId

Parameter: 2 - String-newName

Call: /renamePackage?Long1&String1

====- getStopMarkedLink -====

Call: /getStopMarkedLink

====- forceDownload -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Call: /forceDownload?long[]1&long[]2

====- movePackages -====

Parameter: 1 - long[]-packageIds

Parameter: 2 - long-afterDestPackageId

Call: /movePackages?long[]1&long1

====- resumeLinks -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Call: /resumeLinks?long[]1&long[]2

====- setEnabled -====

Parameter: 1 - boolean-enabled

Parameter: 2 - long[]-linkIds

Parameter: 3 - long[]-packageIds

Call: /setEnabled?boolean1&long[]1&long[]2

====- resetLinks -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Call: /resetLinks?long[]1&long[]2

====- packageCount -====

Call: /packageCount

====- movetoNewPackage -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-pkgIds

Parameter: 3 - String-newPkgName

Parameter: 4 - String-downloadPath

Call: /movetoNewPackage?long[]1&long[]2&String1&String2

====- removeLinks -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Call: /removeLinks?long[]1&long[]2

====- splitPackageByHoster -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-pkgIds

Call: /splitPackageByHoster?long[]1&long[]2

====- help -====

Description: This Call

Call: /help

====- setDownloadDirectory -====

Parameter: 1 - String-directory

Parameter: 2 - long[]-packageIds

Call: /setDownloadDirectory?String1&long[]1

====- startOnlineStatusCheck -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Call: /startOnlineStatusCheck?long[]1&long[]2

====- moveLinks -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long-afterLinkID

Parameter: 3 - long-destPackageID

Call: /moveLinks?long[]1&long1&long2

====- setDownloadPassword -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Parameter: 3 - String-pass

Call: /setDownloadPassword?long[]1&long[]2&String1

====- getStopMark -====

Call: /getStopMark

====- queryPackages -====

Parameter: 1 - PackageQueryStorable-queryParams

Call: /queryPackages?PackageQueryStorable1

====- removeStopMark -====

Call: /removeStopMark

## **/downloadcontroller**

org.jdownloader.api.downloads.v2.DownloadWatchdogAPI

====- help -====

Description: This Call

Call: /help

====- start -====

Call: /start

====- stop -====

```
Call: /stop

====- pause -====
Parameter: 1 - boolean-value
Call: /pause?boolean1

====- getSpeedInBps -====
Call: /getSpeedInBps

====- forceDownload -====
Parameter: 1 - long[]-linkIds
Parameter: 2 - long[]-packageIds
Call: /forceDownload?long[]1&long[]2

====- getCurrentState -====
Call: /getCurrentState
```

## /events

org.apwork.remotepi.events.EventsAPIInterface

```
====- changesubscriptiontimeouts -====
  Parameter: 1 - long1
  Parameter: 2 - long2
  Parameter: 3 - long3
    Call: /changesubscriptiontimeouts?long1&long2&long3

====- help -====
  Description: This Call
    Call: /help

====- unsubscribe -====
  Parameter: 1 - long1
    Call: /unsubscribe?long1

====- getsubscriptionstatus -====
  Parameter: 1 - long1
    Call: /getsubscriptionstatus?long1

====- listen -====
  Parameter: 1 - long1
    Call: /listen&long1

====- subscribe -====
  Parameter: 1 - String[]1
  Parameter: 2 - String[]2
    Call: /subscribe?String[]1&String[]2

====- addsubscription -====
  Parameter: 1 - long1
  Parameter: 2 - String[]1
  Parameter: 3 - String[]2
    Call: /addsubscription?long1&String[]1&String[]2

====- listpublisher -====
    Call: /listpublisher

====- removesubscription -====
  Parameter: 1 - long1
  Parameter: 2 - String[]1
  Parameter: 3 - String[]2
    Call: /removesubscription?long1&String[]1&String[]2

====- getsubscription -====
  Parameter: 1 - long1
    Call: /getsubscription?long1
```

====- setsubscription -====

Parameter: 1 - long1

Parameter: 2 - String[]1

Parameter: 3 - String[]2

Call: /setsubscription?long1&String[]1&String[]2

## /extraction

org.jdownloader.api.extraction.ExtractionAPI

====- help -====

Description: This Call

Call: /help

====- addArchivePassword -====

Parameter: 1 - String-password

Call: /addArchivePassword?String1

====- getArchiveInfo -====

Returns: ArchiveStatusStorable[]

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Call: /getArchiveInfo?long[]1&long[]2

====- getArchiveSettings -====

Parameter: 1 - String[]-archiveIds

Call: /getArchiveSettings?String[]1

====- setArchiveSettings -====

Parameter: 1 - String-archiveId

Parameter: 2 - ArchiveSettingsAPIStorable-archiveSettings

Call: /setArchiveSettings?String1&ArchiveSettingsAPIStorable1

====- startExtractionNow -====

Parameter: 1 - long[]-linkIds

Parameter: 2 - long[]-packageIds

Call: /startExtractionNow?long[]1&long[]2

====- getQueue -====

Call: /getQueue

====- cancelExtraction -====

Parameter: 1 - long-controllerID

Call: /cancelExtraction?long1

### ControllerStatus Enum

- RUNNING // Extraction is currently running
- QUEUED //Archive is queued for extraction and will run as soon as possible
- NA //No controller assigned.

### ArchiveStatusStorable Enum

```
{
// -1 or the controller ID if any controller is active. Used in
extraction/cancelExtraction?<ControllerID>
    "controllerId": -1,
    "archiveName": "The name of the archive",
// The status of the controller
    "controllerStatus": "<ControllerStatus>",

// Type of the archive. e.g. "GZIP_SINGLE", "RAR_MULTI", "RAR_SINGLE", ....
    "type": null,

// ID to address the archive. Used for example for
extraction/getArchiveSettings? [<ARCHIVE_ID_1>, <ARCHIVE_ID_2>, ...]
    "archiveId": null,

// File State Map
    "states": {"filename1": "<ArchiveFileStatus>"....}
}
```

## **/system**

org.jdownloader.api.system.SystemAPI

====- exitJD -====

Call: /exitJD

====- help -====

Description: This Cäall

Call: /help-.

====- restartJD -====

Call: /restartJD

====- hibernateOS -====

Call: /hibernateOS

====- shutdownOS -====

Parameter: 1 - boolean-force

Call: /shutdownOS?boolean1

====- standbyOS -====

Call: /standbyOS

## /captcha

org.jdownloader.api.captcha.CaptchaAPI

====- help -====

Description: This Call

Call: /help

====- list -====

Description: returns a list of all available captcha jobs

Call: /list

====- skip -====

Parameter: 1 - long-id

Call: /skip?long1

====- skip -====

Parameter: 1 - long-id

Parameter: 2 - SkipRequest-type

Call: /skip?long1&SkipRequest1

====- getCaptchaJob -====

Description: Returns CaptchaJob Object for the given id

Parameter: 1 - long-id

Call: /getCaptchaJob?long1

====- get -====

Description: Returns Captcha Image as Base64 encoded data url

Parameter: 1 - long-id

Call: /get&long1

====- solve -====

Parameter: 1 - long-id

Parameter: 2 - String-result

Call: /solve?long1&String1

====- Enums -====

Enum: SkipRequest - [SINGLE, BLOCK\_HOSTER, BLOCK\_ALL\_CAPTCHAS, BLOCK\_PACKAGE, REFRESH, STOP\_CURRENT\_ACTION, TIMEOUT]

**/downloadevents**

org.jdownloader.api.downloads.DownloadControllerEventPublisherInterface

====- help -====

Description: This Call

Call: /help

====- queryLinks -====

Parameter: 1 - LinkQueryStorable-queryParams

Parameter: 2 - int-diffID

Call: /queryLinks?LinkQueryStorable1&int1

====- setStatusEventInterval -====

Parameter: 1 - long-channelID

Parameter: 2 - long-interval

Call: /setStatusEventInterval?long1&long2

## **/extensions**

org.jdownloader.api.extensions.ExtensionsAPI

====- list -====

Parameter: 1 - ExtensionQueryStorable-query  
Call: /list?ExtensionQueryStorable1

====- install -====

Parameter: 1 - String-id  
Call: /install?String1

====- help -====

Description: This Call  
Call: /help

====- isEnabled -====

Parameter: 1 - String-classname  
Call: /isEnabled?String1

====- setEnabled -====

Parameter: 1 - String-classname  
Parameter: 2 - boolean-b  
Call: /setEnabled?String1&boolean1

====- isInstalled -====

Parameter: 1 - String-id  
Call: /isInstalled?String1

**/jd**

org.jdownloader.api.jd.JDAPI

====- doSomethingCool -====  
    Call: /doSomethingCool

====- help -====  
    Description: This Call  
    Call: /help

====- sum -====  
    Parameter: 1 - int-a  
    Parameter: 2 - int-b  
    Call: /sum?int1&int2

====- uptime -====  
    Call: /uptime

====- refreshPlugins -====  
    Call: /refreshPlugins

====- version -====  
    Call: /version

====- getCoreRevision -====  
    Call: /getCoreRevision

**/linkcrawler**

org.jdownloader.api.linkcrawler.LinkCrawlerAPI

====- help -====

Description: This Call

Call: /help

====- isCrawling -====

Call: /isCrawling

**/polling**

org.jdownloader.api.polling.PollingAPI

====- help -====

Description: This Call

Call: /help

====- poll -====

Parameter: 1 - APIQuery-queryParams

Call: /poll?APIQuery1