mod_quickjs / builtin classes

(version: 1.0_16012022)

Revisions history:

- 11082021 - initial version
- 16012022 - added ODBC, XML
  changes: CURL, Session

Table of content:

Codec

Properties:

Property	Type	Access	Description
isReady	Boolean	RO	indicate the codec state
canEncode	Boolean	RO	codec supports encode
canDecode	Boolean	RO	codec supports decode
name	String	RO	codec name
ptime	Iteger	RO	packetization time (in ms)
channels	Iteger	RO	number of channels
samplerate	Iteger	RO	samplerate (bits/sec)

Methods:

- **Codec(session, name, samplerate, channels, ptime)**
  name - codec name
  samplerate - samplerate
  channels - number of channels
  ptime - packetisation time (ms)

- **encode(srcBuff, srcSmpRate, srcLen, dsrBuff, dstSmpRate)**
  return: encoded data length (bytes)
  srcBuff - ArrayBuffer
  dstBuff - ArrayBuffer
  scrLen - src data length

- **decode(srcBuff, srcSmpRate, srcLen, dsrBuff, dstSmpRate)**
  return: decoded data length (byte)
  srcBuff - ArrayBuffer
  dstBuff - ArrayBuffer
  scrLen - src data length

Manuals : https://docs.freeswitch.org/group__codecs.html

Examples: examples/session_frames.js

CoreDB

Properties:

Property	Type	Access	Description
path	String	RO	database location path

Methods:

- **CoreDB(dbName)**
  dbName - database name (without extension)

- **exec(sqlQuery, [fetchCallback])**
  return: rows count for update queries
  Execute update or select query
  sqlQuery - sql query
  callback - function(rowData) { ...
  rowData - object with properties equals row columns names

- **prepare(sqlQuery)**
  return: true/false
  Processing the query and prepare statement for next manipulation

- **next()**
  return: true/false
  Move to the next record in a result (use it after **prepare** method)

- **step()**
  Move to the next record in a result (use it after **prepare** method)
  return: true/false

- **fetch()**
  return: object or undefiend
  Get row data as associative array (keys will be equals columns names)

- **bindText()**
  return: true/false
  Bind parameter in the query as text (use it after **prepare** method)

- **bindInt()**
  return: true/false
  Bind parameter in the query as int (use it after **prepare** method)

- **tableExists(tableName)**
  return: true/false
  Check the table for existence

Examples: examples/coredb_frames.js

CURL

Properties:

Property	Type	Access	Description
isReady	Boolean	RO	true if the clieant ready to make queries
url	String	RW	url
method	String	RW	HTTP methos: GET/POST (by default: GET)
timeout	Integer	RW	the request maximum time in seconds
credentials	String	RW	name:password (credentials for auth basic method)
contentType	String	RW	the request content type (default: application/json)
sslVerfyPeer	Boolean	RW	verify the peer's SSL certificate
sslVerfyHost	Boolean	RW	verify the certificate's name against host
sslCAcert	String	RW	path to CA certificate (with CA file name)
proxy	String	RW	proxy url [http(s)://ip:port/]
proxyCredentials	String	RW	proxy credentials [username:password]
proxyCAcert	String	RW	is as in sslCAcert property

Methods:

- **CURL(url, [method, timeout, credentials, contentType])**
  url   - [http|https]://....
  method - [POST | GET] (by default GET)

- **doRequest()**
  ```
  return: { error         : null or error message,
            content       : response data
            contentLength : response data length }
  ```
  Execute the request.

- **uploadFile(path)**
  return: 0 or http error code
  Upload a file to the server

Examples: examples/curl_test.js

DTMF

Properties:

Property	Type	Access	Description
digit	String	RO	digits that received
duration	Integer	RO	not used

Methods:

- **DTMF([duration])**

Examples: examples/session_input.js

Event

Properties:

Property	Type	Access	Description
isReady	Boolean	RO	true if the event is ready to use

Methods:

- **Event(eventName [, subClassName])**
  eventName   - an event name
  subClassName - use it with custom events

- **addHeader(name, value)**
  return: true/false

- **getHeader(name)**
  return: string or undefined

- **addBody(bodyData)**
  return: true/false

- **getBody()**
  return: string or undefined

- **getType()**
  return: event type as string

- **serialize(format)**
  return: string
  Text represetantion of the event
  format - ['xml' | 'json'] or leave it blank for plain text

- **fire()**
  return: true/false
  Emit the event and free used resources

EventHandler

Properties:

Property	Type	Access	Description
isReady	Boolean	RO	true if the handler is ready to use

Methods:

- **Eventhandler()**

- **subscribe(eventName [,eventName1,EventName2,...])**
- **subscribe(CustomEvent, subClassName [,...])**
  return: true/false

- **unsubscribe(eventName [,eventName1,EventName2,...])**
- **unsubscribe(CustomEvent, subClassName [,...])**
  return: true/false

- **addFilter(headerName, headerValue)**
  return: true/false

- **delFilter(headerName)**
  return: true/false

- **getEvent([timeout])**
  return: Event or undefiend
  timeout - timeout in ms

- **sendEvent(event [, sessionUUID])**
- **sendEvent(event [, session])**
  return: true/false
  Send the event as global or for session

FileHandle

Properties:

Property	Type	Access	Description
speed	Inreger	RW	playback speed (use [+1|-1] for steps)
volume	Integer	RW	playback volume (use [+1|-1] for steps)
channels	Integer	RO	channels number
samplerate	Integer	RO	media samplerate

Methods:

- **FileHandle(path, session, flags)**
  path - absolute path or only name (will be merged with the default media path)
  flags - read|write|short|int|double|float|raw

- **pause()**
  return: true/false
  Turn on/off pause on playback/record

- **truncate()**
  return: true/false
  Truncate the file

- **restart()**
  return: true/false
  Move to the start of the file and set speed to default

- **seek(position)**
  return: true/false
  position - absolute or offset: +NNN | -NNNN (in samples)

- **read(buffer, samples)**
  return: read samples
  Read NNN samples into buffer.
  buffer - ArrrayBuffer
  samples - sample to read

- **write(buffer, samples)**
  return: wrote samples
  Write NNN samples from buffer
  buffer - ArrrayBuffer
  samples - sample to write

- **close()**
  return: true/false
  Close the handle and free used resources

- **stop()**
  return: true/false
  Set flag: SWITCH_FILE_DONE

Manuals : https://docs.freeswitch.org/group__Media.html

File

Properties:

Property	Type	Access	Description
canRead	Boolen	RO	possible to read
canWrite	Boolen	RO	possible to write
isOpen	Boolen	RO	true if File or Directory opened
isFile	Boolen	RO	true if this is a file object
isDirectory	Boolen	RO	true if this is a directory
path	String	RO	absolute path
name	String	RO	file name
size	Integer	RO	file size or count of elements in a directory
position	Integer	RO	file position
creationTime	Intger	RO	not used (always 0)
lastModified	Intger	Ro	not used (always 0)

Methods:

- **File(path)**
  path - absolute path or mask (myfile.XXXXX) for a temporary file

- **exists()**
  return: true/false
  Check file or directory for existence

- **mktemp()**
  return: true/false
  Create a temporary file by mask, this file will be automatically removed

- **open(mask)**
  return: true/false
  mask - [r|w|c|a|t|b] (r-read, w-write, c-create, a-append, t-truncate, b-binary)
  Open file or directory

- **close()**
  return: true/false
  Close file or directory and free used resources

- **read(buffer, lenght)**
  return: bytes read
  buffer - ArrayBuffer
  length - bytes to read

- **write(buffer, lenght)**
  return: bytes wrote
  buffer - ArrayBuffer
  length - bytes to write

- **writeString(str)**
  return: bytes wrote

- **readString(lenght)**
  return: string
  lenght - bytes to read

- **seek(position)**
  return: true/false
  position - absolute value (0 - Nxxx)

- **remove()**
  return: true/false
  Remove file or directory

- **rename(newName)**
  return: true/false
  Rename a file

- **copy(newPath)**
  return: true/false
  Copy a file

- **mkdir()**
  return: true/false
  Create directory that defined in constructor (recursive)

- **list(callback)**
  return: true/false
  Explore a directory (open it before use this function)
  callback - function(path, name) { ... }

  ```
            path - absolute path
            name - short name
  ```

Examples: examples/file_test.js

Socket

Properties:

Property	Type	Access	Description
type	String	RO	socket type
nonblock	Boolean	RW	nonblock flag (default: false)
timeout	Integer	RW	connection timeout in ms (default: 0)
ttl	Integer	RW	TTL value for multicast sockets (default: 1)

Methods:

- **Socket('tcp' [,timeout])**
- **Socket('udp', localAddress)**

- **connect(host, port)**
  return: true/false
  For TCP/UDP, you should define target host and port

- **connect(mcastGroup, port)**
  return: true/false
  For MULTICAST, you should define group and port

- **close()**
  return: true/false
  Close the socket and free used resources

- **write(buffer, len)**
  return: byte wrote
  buffer - ArraayBuffer
  len - bytes to write

- **read(buffer, len)**
  return: bytes read
  buffer - ArraayBuffer
  len - bytes to read

- **writeString(str)**
  return: byte wrote

- **readString([delimeter])**
  return: string or undefined
  Read data from the socket and stop on delimiter (by default \n)

Examples: examples/soc_tcp_test.js
examples/soc_udp_test.js

Session

Properties:

Property	Type	Access	Description
isReady	Boolean	RO	true if the session and channel are ready
isAnswered	Boolean	RO	true if channel answered
isMediaReady	Boolean	RO	true if channel media is ready
name	String	RO	session name (sofia path)
uuid	String	RO	session UUID
state	String	RO	channel state
cause	String	RO	hangup cause
causecode	Integer	RO	hangup cause code
dialplan	String	RO	dialplan name
destination	String	RO	destination number
callerIdName	String	RO	
callerIdNumber	String	RO	
readCodecName	String	RO	
writeCodecName	String	RO	
samplerate	Integer	RO	
channels	Integer	RO	
ptime	Integer	RO	session packetisation time (ms)

Methods:

- **Session(uuid)**
  Lookup session by UUID

- **Session(oldSession)**
  Bridge a new session

- **setHangupHook(callback)**
  return: true/false
  callback - function(causeCode) { ... }

- **setAutoHangup()**
  return: true/false

- **speak(ttsEngine, ttsVoice, ttsText [, callback [,cbArg]])**
  return: true/false
  ttsEngine - engine name (for example: flite)
  ttsVoice - if null will be getting from variable: tts_voice
  ttsText - text to speach
  callback - function(session, dtmf, type, cbArg) { ... }
  type - ['dtmf' | 'event']
  cbArg - custom data

- **sayPhrase(macroName, data, language, [callback])**
  return: true/false
  Interface to 'switch_ivr_phrase_macro_event'

- **streamFile(path [, callback [, cbArg]])**
  return: true/false
  path - string or File object
  callback - function(filehandle, dtmf, type, cbArg) { ... }
  type - ['dtmf' | 'event']
  cbArg - custom data

- **recordFile(path [, callback [, cbArg, limit, thresold, silenceHits]])**
  return: true/false
  path - string or File object
  limit - record limit (seconds)
  silTresh - silence threshold
  silHits - silence hits
  callback - function(filehandle, dtmf, type, cbArg) { ... }
  type - ['dtmf' | 'event']
  cbArg - custom data

- **collectInput(cllback, cbArg [, absTimeout])**
- **collectInput(cllback, cbArg [, digiTimeout, absTimeout])**
  return: true/false
  absTimeout - summary time to wait for input
  digiTimeout - time for a single digit
  callback - function(filehandle, dtmf, type, cbArg) { ... }
  type - ['dtmf' | 'event']
  cbArg - custom data

- **flushEvents()**
  return: true/false
  Clear session events queue

- **flushDigits()**
  return: true/false
  Clear session DTMF input

- **setVariable(name, value)**
  return: true/false
  Set session variable

- **getVariable(name)**
  return: variable or undefined
  Get session variable

- **getDigits(digits [, terminator, timeout, digitTimeout, absTimeout])**
  return: string with digits or undefined
  See: switch_ivr_collect_digits_count

- **answer()**
  return: true/fasle
  Answer on a call (switch_channel_answer)

- **preAnswer()**
  return: true/fasle
  PreAnswer on a call (switch_channel_pre_answer)

- **generateXmlCdr()**
  return: CDR as xml string or undefined
  See: switch_ivr_generate_xml_cdr()

- **getEvent()**
  return: Event object or undefiend

- **sendEvent(event)**
  return: true/fasle

- **hangup(causeCode)**
- **hangup(causeName)**
  return: true/fasle
  Hangup the call

- **execute(appName[, appArg])**
  return: true/false
  Call another module.

- **sleep(timeout [, callback [, cbArg]])**
  return: true/false
  timeout - in milliseconds
  callback - function(session, dtmf, type, cbArg) { ... }
             type - ['dtmf' | 'event']
             cbArg - custom data

- **genTones(tgmlScript, [callback, cbArg])**
  return: true/false
  Generate tones to the channel
  tgmlScript - see: https://freeswitch.org/confluence/display/FREESWITCH/TGML
  callback   - function(session, dtmf, type, cbArg) { ... }
               type - ['dtmf' | 'event']
               cbArg - custom data

- **getReadCodec()**
  return: Codec object or undefined
  Get session read codec.

- **getWriteCodec()**
  return: Codec object or undefined
  Get session write codec.

- **frameRead(buffer)**
  return: samples read
  Read session frame and copy data into buffer.
  buffer - ArrayBuffer

- **frameWrite(buffer, len, [writeCodec])**
  return: samples wrote
  Write session frame.
  buffer   - ArrayBuffer
  len   - samples to write
  writeCoec - if defined will be used to encode

- **playAndGetDigits(min_digits, max_digits, max_tries ,timeout, terminators, audio_file, bad_audio_file, [digits_regex, var_name, digit_timeout, transfer_on_failure])**
  return: entered symbols as string (or empty string)

  Play a sound and gather digits with the number of retries specified if the user doesn't give digits in the set time (equivalent of switch_play_and_get_digits())

  min_digits   - the fewest digits allowed for the response to be valid
  max_digits   - the max number of digits to accept
  max_tries   - number of times to replay the sound and capture digits
  timeout   - time to wait for input
  terminators   - for input that can include # or *
  audio_file   - file to play
  bad_audio_file   - file to play if the input from the user was invalid

Manuals : https://docs.freeswitch.org/switch__core__session_8c.html

Examples: examples/session_rec_play.js
          examples/session_frames.js
          examples/session_input.js

ODBC

Properties:

Property	Type	Access	Description
isConnected	Boolean	RO	true when connection is established
dsn	String	RO	DSN
username	String	RO	null or your db username
password	String	RO	null or your db password
numRows	Integer	RO	0 or available rows number after a query was executed
numCols	Integer	RO	0 or available rows number after a query was executed

Methods:

- **connect()**
  return: true/false
  Connect to the DB

- **disconnect()**
  return: true/false
  Disconnect from the DB and clean resources

- **query(sqlQuery)**
- **exec(sqlQuery)**
  return: true/false
  Execute the query and prepare the statement for get results.

- **execute(sqlQuery)**
  return: true/false
  Execute query without results.

- **nextRow()**
  return: true/false
  Move to next record.

- **getData()**
  return: map of columns or undefined
  Get row data.

Examples: examples/odbc_test.js

XML

Properties:

Property	Type	Access	Description
name	String	RO	tag name
data	String	RW	tag text
error	String	RO	last error

Methods:

- **addChild(name)**
  return: xml object
  Adds a child tag.
  name - a tag name

- **getChild(name [,attrName, arrtValue])**
  return: xml object or undefined
  Find a child tag in a node called 'name' with an attribute 'attrName' which equals 'attrValue'.
  name - tag name
  attrName - attribute name
  arrtValue - attribute value

- **getAttribute(name)**
  return: attr text value or undefined
  Returns the value of the requested tag attribute, or undefined if not found.
  name - attribute name

- **setAttribute(name, value)**
  return: true/false
  name - attribute name
  value - new value

- **remove()**
  return: true/false
  Removes a tag along with all its subtags

- **copy()**
  return: xml object
  Duplicates an exists tag

- **next()**
  return: xml object or undefined
  Returns the next tag of the same name in the same section and depth or NULL.

- **serialize()**
  return: text
  Serialize XML object into text representation.

Examples: examples/xml_test.js