

# developer garden

Voice Record Lab API Documentation - Version Labs-1.0

## Table of Contents

1. Overview .....	1
1.1. About this document .....	1
1.2. Change history .....	1
1.3. Feedback .....	1
1.4. Copyright .....	1
2. Introduction .....	2
2.1. Voice Call Data Objects .....	2
3. Voice Record Lab API .....	4
3.1. Overview .....	4
3.2. Voice Record Java SDK .....	4
3.3. VoiceRecord Interface .....	4
3.3.1. Method <code>createVoiceRecord</code> .....	4
3.3.2. Method <code>getVoiceRecordStatus</code> .....	6
3.3.3. Method <code>getVoiceRecord</code> .....	7
3.3.4. Method <code>deleteVoiceRecord</code> .....	8
3.4. Interaction schema .....	9
3.5. VoiceRecord REST Interface .....	9
3.5.1. REST Method: Initiate recording of a voice data record .....	10
3.5.2. REST Method: Return status and meta-information of a voice record .....	12
3.5.3. REST Method: Delete voice record .....	15
3.6. Voice Record Mock Environment .....	16
Bibliography .....	18

# Chapter 1. Overview

## 1.1. About this document

This document describes the Voice Record Lab API which allows to record voice call data objects persistently.

## 1.2. Change history

Table 1.1. Change history of the Voice Record Lab API documentation

Version	Changes to the previous version
Labs-1.0	Release of Voice Record Lab API

## 1.3. Feedback

We welcome your suggestions, comments, and questions. The Developer Garden (at [www.developergarden.com](http://www.developergarden.com) [http://www.developergarden.com]) is available for exchanges between service users.

## 1.4. Copyright

This user documentation is copyright protected. All rights reserved.

For more detailed legal notices and information about copyright, see the [Developer Garden](http://www.developergarden.com) [http://www.developergarden.com].

## Chapter 2. Introduction

One thing that stands out immediately is that voice is still missing as a data object in Web 2.0, except of podcasts. Even if the Web aims to integrate all media contents, voice content is still scarcely contained in Web pages. The main content types are text, images and video. Voice integration is mainly limited to click-to-call and Podcast.

Voice is still not considered a mainstream web media, such as images or even video. It is not really "mashable", it is missing tools and most web developers are unfamiliar with it. However, voice is a core asset of telecommunication providers such as Deutsche Telekom AG, due their core business is communication and more specific voice call based communication!

### Web 2.0: Data is the next Intel inside

Tim O'Reilly's famous definition of "What is Web 2.0" [O'Reilly Web 2.0] illustrated several aspects of what is characteristic of Web 2.0. Among other things, he stated that every significant Web 2.0 application to date has been backed by specialized data. Just to name a few among the top US Web companies, images are the data asset of Flickr, YouTube deals with videos, Amazon handles enriched product data, Google's data assets are indexed web pages and targeting information and so on.

Voice calls/conversations stored as persistent Web data objects are a valuable data assets too! They should be applicable to typical Web 2.0 alike mechanisms. They should be "mashable", due they should be stored in Web storages and should be accessible from Web applications via permalinks and HTML mashup code just as YouTube videos or Flickr images are.

### 2.1. Voice Call Data Objects

The opportunities of voice call data objects are manifold – ranging from voice messaging to voice conversations. Voice call data objects allow to record one-way voice messages and publish them in the web (e.g. voice comments).

But voice messages can never substitute text messages, as text can be much faster picked up. Voice can not be "fast forwarded" and today there is no mainstream mechanism to automatically detect key words and search phrases. Still voice has advantages in certain areas:

- It transports emotions, which are certainly important in social networks.
- It can be used for fun messages.
- It can be used for personalization (a voice is as important as a picture).
- It can be quite useful in mobile scenarios (with optional Speech-to-Text features).

The heart of voice call data objects is that voice calls result in data objects storing telephone calls or conversations persistently. Voice call data objects can be linked and mashed up within web and enterprise applications comparable to the linkage of YouTube videos or Flickr images in today's web applications. Voice call data objects enable web (and enterprise) applications to integrate voice messaging and conversation functions.

They are "mashable", due they are be stored in Web storages and are accessible from Web applications via permalinks and HTML mashup code just as YouTube videos or Flickr images are. The publishing and mashup of voice data objects within mashups is supported by a voice data audio player which is started when accessing voice data object via the corresponding permalink.

Recent trends in social networking show, that there is an accelerating demand for voice messaging within social networks such as Facebook. Voice messages are much more personal and emotionally than text mes-

sages. This may fit well into "social" networks. Leaving voice comments on web pages may be attractive and comfortable to users as well and allows the users to leave emotional comments in blogs, product descriptions of merchants or in online media.

## Chapter 3. Voice Record Lab API

### 3.1. Overview

The Voice Record Lab API allows to record voice calls. It initiates calls to a given phone number and prompts the user for voice input. The Voice Record Lab API stores the resulting voice call data object and returns a URL linking to the voice call data object and a URL to a corresponding built-in player provided by the service. The voice call data object can be played with the build-in player or any other player (e.g. preferred external player of the user) as well.

The Voice Record Service offers SOAP and REST interfaces. [Section 3.2, “Voice Record Java SDK”](#) describes a Java Voice Record SDK supplied in order to conveniently interact with the Voice Record Interface described in [Section 3.3, “VoiceRecord Interface”](#). [Section 3.5, “VoiceRecord REST Interface”](#) introduces the REST interface.

### 3.2. Voice Record Java SDK

In order to use the Voice Record Lab API conveniently a Voice Record Java SDK is supplied. The SDK is available under [Apache License version 2.0](http://www.apache.org/licenses/LICENSE-2.0) [http://www.apache.org/licenses/LICENSE-2.0].

### System Requirements

- Developer Garden Java SDK version 2.5.4 or higher
- Sun JDK 1.5 or higher

The method signatures and parameters of the Voice Record SDK are aligned with the `VoiceRecord` Interface described in the following section. For more details please refer to the JavaDoc of Voice Record SDK included in the SDK.

### 3.3. VoiceRecord Interface

The Voice Record Lab API is realized by the component `VoiceRecord`, which provides methods to create a voice call data object (`createVoiceRecord`), to query status information about the voice call data object (`getVoiceRecordStatus`), to retrieve the voice call data object Meta information (`getVoiceRecord`) and to delete a voice call object (`deleteVoiceRecord`). These methods will be described in the following sections.

#### 3.3.1. Method `createVoiceRecord`

The method `createVoiceRecord` initiates the recording of a voice data object. First, it establishes a phone call to the given phone number and prompts the called party for voice input by means of playing a welcome message. Afterwards, the voice input of the callee is recorded and a goodbye message is played. The resulting voice call data object is stored persistently.

Welcome and goodbye message can be supplied as optional parameter by means of URLs pointing to existing voice call data objects. These voice call data objects have to be recorded beforehand. Future versions of the Voice Record Lab API will provide the opportunity to supply URLs to any other audio file used as welcome or goodbye message. The audio format WAV (PCM A-law uncompressed, 8 Bit, mono, 8kHz) is used for all voice call data objects, i.e. recorded voice call data object and welcome / goodbye message.

The method `createVoiceRecord` utilizes an asynchronous behavior. It returns immediately checking the validity of input parameters, because otherwise the method call would block until the callee finishes the recording and hangs up.

The method `createVoiceRecord` returns a reference to the voice call data object to be recorded which is required to request status information about the call establishment and voice recording. In order to retrieve this status information, the [getVoiceRecordStatus](#) method has to be invoked.

Table 3.1. Parameter list of method `createVoiceRecord`

Name	Description
<code>recordingPhoneNo</code>	The underlying voice application calls this phone number and prompts the callee for the voice message to be stored.
<code>startTime</code> (optional)	The call establishment will be started at the given date and time. The <code>startTime</code> needs to be between now and 24 hours in the future.  If no <code>startTime</code> is specified, the call will be established immediately.
<code>welcomeMsgMediaObjRef</code> (optional)	URL reference to a voice call data object that stores an individual welcome message prompt.  If no welcome message prompt is specified, the default prompt is used.
<code>goodbyeMsgMediaObjRef</code> (optional)	URL reference to a voice call data object that stores an individual goodbye message prompt which will be played at the end of the recording.  If no goodbye message prompt is specified, the default goodbye message prompt is used.
<code>cannotHearYouMsgMediaObjRef</code> (optional)	URL reference to a voice call data object that stores an individual message prompt which will be played when the callee speaks not loud enough.  If no message prompt is specified, the default message prompt is used.
<code>maxRecordingTime</code> (optional)	An application specific maximum recording time for a voice call data object in seconds. This timeout can be used to override the system specific timeout of 30 minutes (1800 seconds), if it is too large. Maximum recording times larger than 30 minutes are forbidden.

Table 3.2. Response of method `createVoiceRecord`

Name	Description
<code>mediaObjRef</code>	Unique references to voice call data object to be recorded. The reference is required in order to get either the status or the voice call data object itself.

Table 3.3. Exceptions of method **createVoiceRecord**

Name	Description
<code>invalidPhoneNumber</code>	The <code>recordingPhoneNo</code> number is invalid.
<code>welcomeMsgNotExists</code>	The <code>welcomeMsgMediaObjRef</code> does not exist.
<code>goodbyeMsgNotExists</code>	The <code>goodbyeMsgMediaObjRef</code> does not exist.
<code>cannotHearYouMsgNotExists</code>	The <code>cannotHearYouMsgMediaObjRef</code> does not exist.
<code>invalidStartTime</code>	The <code>startTime</code> lies in the past already or lies more than 24 hours in the future.
<code>internalError</code>	Internal system error.

### 3.3.2. Method `getVoiceRecordStatus`

The method `getVoiceRecordStatus` returns the status of the voice call data object and the call establishment process. Due to the asynchronous nature of the `createVoiceRecord` method, the `getVoiceRecordStatus` method has to be used in order to check if the call could be established successfully or not. The method returns whether the call is queued for execution, whether the phone is ringing, whether the callee did not answer, whether the recording of the voice call data object is in progress, was finished successfully or was aborted by the callee. This method is called best within a loop prior to accessing the voice call data object with the `getVoiceRecord` method.

 Table 3.4. Parameter list of method **getVoiceRecordStatus**

Name	Description
<code>mediaObjRef</code>	Unique reference to a voice call data object.

Table 3.5. Response of method **getVoiceRecordStatus**

Name	Description
mediaObjStatus	<p>Status of the voice call data object:</p> <ul style="list-style-type: none"> <li>• <b>Queued</b> Outbound call is queued for execution.</li> <li>• <b>Recording</b> Recording of the voice call data object is in progress.</li> <li>• <b>Finished</b> Recording of the voice call data object has been finished normally.</li> <li>• <b>Aborted</b> Recording of the voice call data object has been aborted by the callee.</li> <li>• <b>Error</b> An error occurred during recording of the voice call data object.</li> </ul>

Table 3.6. Exceptions of method **getVoiceRecordStatus**

Name	Description
invalidMediaObjRef	The mediaObjRef is invalid and does not point to a valid voice call data object.
internalError	Internal system error.

### 3.3.3. Method **getVoiceRecord**

The method `getVoiceRecord` returns meta information required to access a voice call data object. The meta information comprises a media object URL linking to a voice call data object, a media object player URL to a corresponding built-in player provided by the service and a string containing HTML code fragment which can be used to embed the voice call data object into a Web page.

Table 3.7. Parameter list of method **getVoiceRecord**

Name	Description
mediaObjRef	Unique reference to a voice call data object.

Table 3.8. Response of method **getVoiceRecord**

Name	Description
mediaObjectInfo	<p>Media object information holder that contains the following information:</p> <ul style="list-style-type: none"> <li>• <code>mediaObjectURL</code> A URL reference that links to the voice call data object directly.</li> <li>• <code>mediaObjectPlayerURL</code> A URL reference that links to the built-in audio player for that voice call data object.</li> <li>• <code>mediaObjectPlayerHtmlCode</code> An HTML code fragment which can be used to embed the voice call data object into a Web page.</li> </ul>

Table 3.9. Exceptions of method **getVoiceRecord**

Name	Description
invalidMediaObjRef	The <code>mediaObjRef</code> is invalid and does not point to a valid voice call data object.
internalError	Internal system error.

### 3.3.4. Method `deleteVoiceRecord`

The method `deleteVoiceRecord` deletes a voice call data object referenced by a unique identifier.

Table 3.10. Parameter list of method **deleteVoiceRecord**

Name	Description
mediaObjRef	Unique reference to a voice call data object.

Table 3.11. Response of method **deleteVoiceRecord**

Name	Description
deletedContentSize	Size of the voice call data object that has been deleted successfully in bytes.

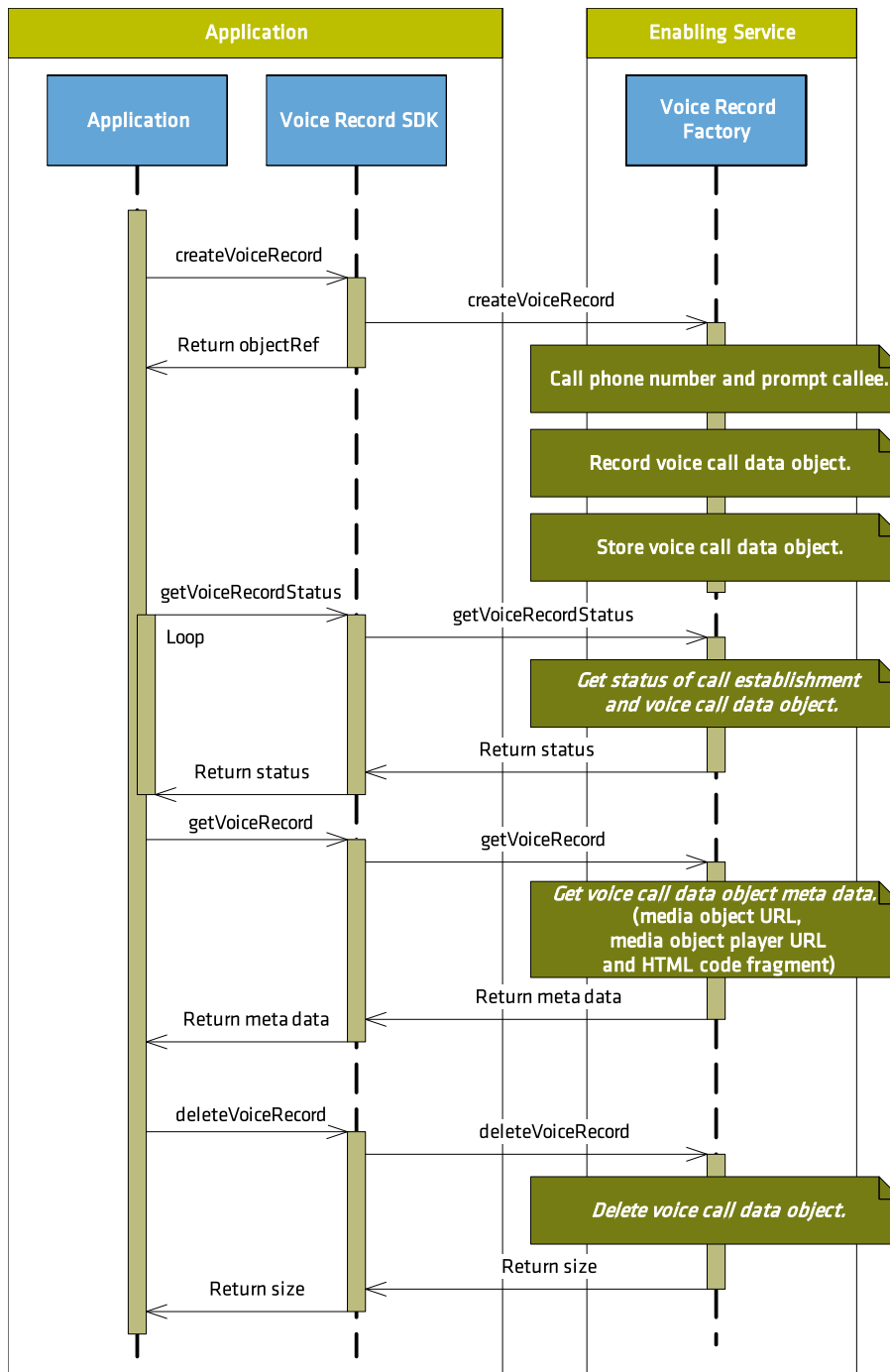
Table 3.12. Exceptions of method **deleteVoiceRecord**

Name	Description
invalidMediaObjRef	The <code>mediaObjRef</code> is invalid and does not point to a valid voice call data object.
internalError	Internal system error.

### 3.4. Interaction schema

The following message sequence chart shows a typical interaction schema between applications using the Voice Record SDK and the Voice Record Lab API:

Figure 3.1. Typical interaction schema



### 3.5. VoiceRecord REST Interface

The VoiceRecord REST Interface is very similar to the Developer Garden REST Interface. Therefore, to invoke a REST method, the acquisition of a security token from the Security Token Server is required.

The only difference is that the authentication token for the Voice Record Lab API has to be obtained from a different path at the Security Token Server:

Path for Developer Garden Services	https://sts.idm.telekom.com/rest-v1/tokens/odg
Path for Voice Record Lab API	https://sts.idm.telekom.com/rest-v1/tokens/ <b>voicerecording</b>

For further information on this procedure, please see the Appendix of the [Developer Garden Documentation](http://www.developergarden.com/openapi/dokumentation) [http://www.developergarden.com/openapi/dokumentation].

### 3.5.1. REST Method: Initiate recording of a voice data record

This resource realizes the method `createVoiceRecord`

Table 3.13. REST resource information for REST Method: Initiate recording of a voice data record

HTTP Method	POST
REST URL	https://esg.i3alab.net/es-odg/voicerecord-service/rest/<environment>/voicerecord

Table 3.14. REST resource path for REST Method: Initiate recording of a voice data record

Path element	Meaning
/esg.i3alab.net/es-odg/voicerecord-service/rest	URL of the REST interface
/ <b>&lt;environment&gt;</b>	The environment in which the service should be used. Possible values are: <ul style="list-style-type: none"> <li>• production</li> <li>• sandbox</li> <li>• mock</li> </ul>
/voicerecord	

Table 3.15. Request Parameters for REST Method: Initiate recording of a voice data record

Name	Description
recordingPhoneNo	The underlying voice application calls this phone number and prompts the callee for the voice message to be stored.
startTime (optional)	The call establishment will be started at the given date and time. The <code>startTime</code> needs to be between now and 24 hours in the future.  If no <code>startTime</code> is specified, the call will be established immediately.  The format follows ISO8601 specification.

Name	Description
welcomeMsgMediaObjRef (optional)	URL reference to a voice call data object that stores an individual welcome message prompt.  If no welcome message prompt is specified, the default prompt is used.
goodbyeMsgMediaObjRef (optional)	URL reference to a voice call data object that stores an individual goodbye message prompt which will be played at the end of the recording.  If no goodbye message prompt is specified, the default goodbye message prompt is used.
cannotHearYouMsgMediaObjRef (optional)	URL reference to a voice call data object that stores an individual message prompt which will be played when the callee speaks not loud enough.  If no message prompt is specified, the default message prompt is used.
maxRecordingTime (optional)	An application specific maximum recording time for a voice call data object in seconds. This timeout can be used to override the system specific timeout of 30 minutes (1800 seconds), if it is too large. Maximum recording times larger than 30 minutes are forbidden.

## REST invocation (example)

Example 3.1. REST request for Method: Initiate recording of a voice data record

```
POST /es-odg/voicerecord-service/rest/production/voicerecord/ HTTP/1.1
Host: esg.i3alab.net
Authorization: TAuth realm="https://odg.t-online.de", tauth_token="<token>"
Accept: application/json
Content-type: application/x-www-form-urlencoded
Content-Length: 59

recordingPhoneNo=016097xxxxxx&startTime=2010-06-01T12:30:00
```

Example 3.2. REST response for Method: Initiate recording of a voice data record

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Date: Tue, 01 Jun 2010 12:30:01 GMT
Content-Type: application/json

{"mediaObjectRef": "moSJwSGsCH3MQNDY5gNmVYDHR7-xxxxxxxxxx-1234567890000-1234567890000" }
```

Table 3.16. Return values for REST Method: Initiate recording of a voice data record

Field	Meaning
mediaObjRef	Unique references to voice call data object to be recorded. The reference is required in order to get either the status or the voice call data object itself.

Table 3.17. Return codes for REST Method: Initiate recording of a voice data record

REST HTTP code	Message (response body in case of error)
200	OK
400	No phone number was given.
400	The number format is invalid.
400	The number is not allowed.
400	Schedule time format is invalid.
400	Schedule time is invalid.
400	The schedule time exceeded: the number of days until the recording will start is higher than the limit.
401	Token is invalid.
403	Required permissions are missing.
403	The number is not allowed.
404	The environment is invalid.
404	Invalid welcome message URL.
404	Invalid goodbye message URL.
404	Invalid cannot hear you message URL.
406	The accept header is unknown.
500	An internal error occurred.

In case of an error, the detail message is returned as response body:

Example 3.3. Example REST error response

```
HTTP/1.1 404 Not Found
Transfer-Encoding: chunked
Date: Tue, 01 Jun 2010 12:50:01 GMT
Content-Length: 38
Content-Type: text/plain
Vary: Accept-Encoding

The media object reference is invalid.
```

### 3.5.2. REST Method: Return status and meta-information of a voice record

This resource combines the methods `getVoiceRecordStatus` and `getVoiceRecord`. It returns both the status of the voice call data object and the call establishment process as well as meta information required to access and play a voice call data object.

Table 3.18. REST resource information for REST Method: Return status and meta-information of a voice record

HTTP Method	GET
-------------	-----

REST URL	<code>https://esg.i3alab.net/es-odg/voicerecord-service/rest/&lt;environment&gt;/voicerecord/&lt;mediaobjref&gt;</code>
----------	---

Table 3.19. REST resource path for REST Method: Return status and meta-information of a voice record

Path element	Meaning
<code>https://esg.i3alab.net/es-odg/voicerecord-service/rest</code>	URL of the REST interface
<code>/&lt;environment&gt;</code>	The environment in which the service should be used. Possible values are: <ul style="list-style-type: none"> <li>• production</li> <li>• sandbox</li> <li>• mock</li> </ul>
<code>/voicerecord</code>	
<code>/&lt;mediaobjref&gt;</code>	Unique reference to a voice call data object.

This method has no parameters.

## REST invocation (example)

Example 3.4. REST request for Method: Return status and meta-information of a voice record

```
GET /es-odg/voicerecord-service/rest/production/voicerecord/moSJwSGsCH3MQNDY5gNmVYDHR7-
xxxxxxxx-1275657380075-1275561000000 HTTP/1.1
Host: esg.i3alab.net
Authorization: TAuth realm="https://odg.t-online.de",tauth_token="<token>"
Accept: application/json
```

Example 3.5. REST response for Method: Return status and meta-information of a voice record

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Date: Tue, 01 Jun 2010 12:31:00 GMT
Content-Type: application/json

{
  "mediaObjectStatus": "Finished",
  "mediaObjectPlayerHtmlCode": "<object codebase=http://esg.i3alab.net/web_voicerecord/
type=application/x-shockwave-flash"
data=VCDOPlayer.swf width=140 height=20 hspace=0 vspace=0"
name=flashfile id=flashfile"
<param name=movie value=VCDOPlayer.swf"/>
<param name=FlashVars value=vcdoFileURL=http%3A%2F%2Fesg.i3alab.net%2F
web_mediaobjectstoreadapter%2FMediaContentLoaderServlet%3FmediaObjectRef%3D
moSJwSGsCH3MQNDY5gNmVYDHR7-xxxxxxxx-1275657380075-1275561000000%26asMP3%3Dtrue"/>
<param name=AllowScriptAccess value=always"/>
</object> ",
  "mediaObjectPlayerURL": "http://esg.i3alab.net/web_voicerecord/player.jsp?mediaObjectRef=
moSJwSGsCH3MQNDY5gNmVYDHR7-xxxxxxxx-1275657380075-1275561000000",
  "mediaObjectURL": "http://esg.i3alab.net/web_mediaobjectstoreadapter/
MediaContentLoaderServlet?mediaObjectRef=
moSJwSGsCH3MQNDY5gNmVYDHR7-xxxxxxxx-1275657380075-1275561000000"
}
```

Table 3.20. Return values for REST Method: Return status and meta-information of a voice record

Field	Meaning
<code>mediaObjStatus</code>	<p>Status of the voice call data object:</p> <ul style="list-style-type: none"> <li>• <b>Queued</b> Outbound call is queued for execution.</li> <li>• <b>Recording</b> Recording of the voice call data object is in progress.</li> <li>• <b>Finished</b> Recording of the voice call data object has been finished normally.</li> <li>• <b>Aborted</b> Recording of the voice call data object has been aborted by the callee.</li> <li>• <b>Error</b> An error occurred during recording of the voice call data object.</li> </ul>
<code>mediaObjectURL</code>	A URL reference that links to the voice call data object directly.
<code>mediaObjectPlayerURL</code>	A URL reference that links to the built-in audio player for that voice call data object.
<code>mediaObjectPlayerHtmlCode</code>	An HTML code fragment which can be used to embed the voice call data object into a Web page.

Table 3.21. Return codes for REST Method: Return status of a voice data record

REST HTTP code	Message (response body in case of error)
200	OK
401	Token is invalid.
403	Required permissions are missing.
404	The environment is invalid.
404	The media object reference is invalid.
406	The accept header is unknown.
500	An internal error occurred.

In case of an error, the detail message is returned as response body:

Example 3.6. Example REST error response

```
HTTP/1.1 404 Not Found
Transfer-Encoding: chunked
Date: Tue, 01 Jun 2010 12:50:01 GMT
Content-Length: 38
Content-Type: text/plain
Vary: Accept-Encoding

The media object reference is invalid.
```

### 3.5.3. REST Method: Delete voice record

This resource realizes the method `deleteVoiceRecord`.

Table 3.22. REST resource information for REST Method: Delete voice record

HTTP Method	DELETE
REST URL	<code>https://esg.i3alab.net/es-odg/voicerecord-service/rest/&lt;environment&gt;/voicerecord/&lt;mediaobjref&gt;</code>

Table 3.23. REST resource path for REST Method: Delete voice record

Path element	Meaning
<code>https://esg.i3alab.net/es-odg/voicerecord-service/rest</code>	URL of the REST interface
<code>/&lt;environment&gt;</code>	The environment in which the service should be used. Possible values are: <ul style="list-style-type: none"> <li>• <code>production</code></li> <li>• <code>sandbox</code></li> <li>• <code>mock</code></li> </ul>
<code>/voicerecord</code>	
<code>/&lt;mediaobjref&gt;</code>	Unique reference to a voice call data object.

This method has no parameters.

### REST invocation (example)

Example 3.7. REST request for Method: Delete voice record

```
DELETE /es-odg/voicerecord-service/rest/production/voicerecord/moSJwSGsCH3MQNDY5gNmVVDHR7 -
xxxxxxxx-1275657380075-1275561000000 HTTP/1.1
Host: esg.i3alab.net
Authorization: TAuth realm="https://odg.t-online.de",tauth_token="<token>"
Accept: application/json
```

Example 3.8. REST response for Method: Delete voice record

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Date: Tue, 01 Jun 2010 12:40:00 GMT
Content-Type: application/json

{"deletedContentSize": "24896"}
```

Table 3.24. Return values for REST Method: Delete voice record

Field	Meaning
deletedContentSize	Size of the voice call data object that has been deleted successfully in Bytes.

Table 3.25. Return codes for REST Method: Delete voice record

REST HTTP code	Message (response body in case of error)
200	OK
401	Token is invalid.
403	Required permissions are missing.
404	The environment is invalid.
404	The media object reference is invalid.
406	The accept header is unknown.
500	An internal error occurred.

In case of an error, the detail message is returned as response body:

Example 3.9. Example REST error response

```
HTTP/1.1 404 Not Found
Transfer-Encoding: chunked
Date: Tue, 01 Jun 2010 12:50:01 GMT
Content-Length: 38
Content-Type: text/plain
Vary: Accept-Encoding

The media object reference is invalid.
```

## 3.6. Voice Record Mock Environment

In the Mock environment only specific virtual phone numbers are allowed. The virtual phone numbers are constructed according to the following rules and lead to specific simulated behavior. If the number indicates successful recording, a default media object will be created:

Table 3.26. Phone number rules for the mock environment

Phone number	Rule
+4932-000-0- <b>p-m</b>	callee picks up after <b>p</b> seconds and leaves a message of <b>m</b> seconds.

Phone number	Rule
+4932-000-1- <b>p</b> - <b>m</b>	callee picks up after <b>p</b> seconds but then aborts after <b>m</b> seconds.
+4932-000-2- <b>p</b> - <b>m</b>	callee picks up after <b>p</b> seconds but after <b>m</b> seconds an error occurs.
+4932-000-3	callee doesn't pick up.
value range for <b>p</b>	0-99
value range for <b>m</b>	0-999

## Bibliography

[O'Reilly Web 2.0] Tim O'Reilly. What is Web 2.0 [<http://www.oreilynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>].