

RESTful Doodle

At doodle.com, Doodle AG offers a hugely popular service called Doodle®. In addition to the user interfaces that Doodle AG provides in order to access the Doodle service, Doodle AG also provides a RESTful application programming interface (API). The RESTful Doodle API is the topic of this document.

In what follows, the base URL for accessing the RESTful Doodle API is <https://doodle.com/api1>. The base URL for the XML schemas is <http://doodle.com/xsd1>. Accessing the RESTful Doodle API on the production server requires authorization (cf. [AAforRESTfulDoodle.pdf](#)). For testing purposes, the RESTful Doodle API can be accessed without authorization on the test server at <http://doodle-test.com/api1WithoutAccessControl>.¹

Prerequisites

We assume that the reader is familiar with RESTful Web services in general. An excellent text book on this topic is "RESTful Web Services" by Leonard Richardson and Sam Ruby (<http://shop.oreilly.com/product/9780596529260.do>). A text book on this topic that is available in German only is "REST und HTTP" by Stefan Tilkov (<http://dpunkt.de/buecher/3574/REST%20und%20HTTP.html>).

Acknowledgments

We would like to thank Prof. Dr. Dominik Gruntz, Prof. Dr. Jürg Luthiger, Leonard Richardson, Luzian Scherrer, Mark Schmid and Beat Walti for their valuable feedback on various aspects of the RESTful Doodle API and its documentation.

Contents

- [Prerequisites](#)
- [Acknowledgments](#)
- [Contents](#)
- [Poll](#)
- [GET](#)
 - [Request](#)
 - [Resource](#)
 - [Header \(optional\)](#)
 - [Response](#)
 - [Body](#)
 - [Errors](#)
- [POST \(i.e.. "save as"\)](#)
 - [Request](#)
 - [Content type](#)
 - [Resource](#)
 - [Body](#)

¹ In addition to your browser and cURL (<http://curl.haxx.se/>), you can also use our test client (<http://doodle.com/xsd1/TestRESTinPeace.jar>) to experiment with the API; it requires Java SE 6 can be started like any other executable JAR: java -jar TestRESTinPeace.jar Doodle's terms of service (<http://doodle.com/about/tos.html>) also apply to the test client.

[Response](#)
 [Header](#)
 [Status Code](#)
[PUT \(i.e., "save"\)](#)
 [Request](#)
 [Content type](#)
 [Resource](#)
 [Body](#)
 [Header](#)
 [Response](#)
 [Status Code](#)
 [Errors](#)
[DELETE](#)
 [Request](#)
 [Resource](#)
 [Header](#)
 [Response](#)
 [Status Code](#)
 [Errors](#)
[Participant](#)
 [POST \(i.e., "save as"\)](#)
 [Request](#)
 [Content type](#)
 [Resource](#)
 [Body](#)
 [Response](#)
 [Header](#)
 [Status Code](#)
 [Errors](#)
 [PUT \(i.e., "save"\)](#)
 [Request](#)
 [Content type](#)
 [Resource](#)
 [Body](#)
 [Header \(optional\)](#)
 [Response](#)
 [Status Code](#)
 [Errors](#)
 [DELETE](#)
 [Request](#)
 [Resource](#)
 [Header \(optional\)](#)
 [Response](#)
 [Status Code](#)
 [Errors](#)
[Comment](#)
 [POST \(i.e., "save as"\)](#)
 [Request](#)
 [Content type](#)
 [Resource](#)
 [Body](#)
 [Response](#)
 [Status Code](#)
 [DELETE](#)
 [Request](#)
 [Resource](#)

[Header](#)
[Response](#)
[Status Code](#)
[Errors](#)

[User](#)
[Event](#)

Poll

GET

Request

Resource

/polls/**ID**

ID denotes the poll's 16-character identifier.

Header (optional)

X-DoodleKey:**KEY**

KEY denotes the poll's 8-character key. Including the right key makes a difference when the poll is a hidden poll.

Response

Body

The poll encoded in XML according to schema poll.xsd.

Example:

```
<poll xmlns="http://doodle.com/xsd1">
<latestChange>2008-07-23T16:40:50+02:00</latestChange>
<id>rgnrsqvsirr5s22s</id>
<type>DATE</type>
<extensions rowConstraint="1" columnConstraint="5"/>
<hidden>>false</hidden>
<writeOnce>>false</writeOnce>
<requireAddress>>false</requireAddress>
<requireEMail>>false</requireEMail>
<requirePhone>>false</requirePhone>
<byInvitationOnly>>false</byInvitationOnly>
<levels>2</levels>
<state>OPEN</state>
<title>Date Options and Extensions</title>
<description></description>
<initiator>
<name>Paul</name>
<userId></userId>
</initiator>
<options>
<option date="2008-08-26"/>
<option date="2008-08-27">noon</option>
<option startDateTime="2008-08-27T15:00:00" endDateTime="2008-08-27T17:00:00"/>
<option date="2008-08-28">Austin</option>
<option date="2008-08-28">Zurich</option>
</options>
```

```
<participants>
<participant>
<id>1</id>
<name>Myke</name>
<userId>rgnrsvsirr5s22srgnrsvsirr5s22s</userId>
<preferences>
<option></option>
<option>0</option>
<option>0</option>
<option>0</option>
<option>0</option>
<option>1</option>
</preferences>
</participant>
</participants>
<comments>
<comment>
<who>Reto</who>
<userId></userId>
<what>yada, yada, yada</what>
<when>2008-07-23T16:47:41+02:00</when>
</comment >
</comments>
</poll>
```

Errors

- 404: Poll does not exist
- 410: Poll has been deleted

POST (i.e., "save as")

Request

Content type

application/xml

Resource

/polls

Body

The poll encoded in XML according to schema poll.xsd.

Example:

```
<poll xmlns="http://doodle.com/xsd1">
<type>TEXT</type>
<hidden>>false</hidden>
<levels>2</levels>
<title>PPP</title>
<description>yum!</description>
<initiator>
<name>Paul</name>
</initiator>
<options>
<option>Pasta</option>
<option>Pizza</option>
```

```
<option>Polenta</option>
</options>
</poll>
```

An <initiator>'s <name> is mandatory. Optionally, you can also include her/his <eMailAddress>.

Premium Doodle users can also create polls that require participants to provide their address, their phone number, and/or their e-mail address.

Response

Header

Content-Location:**ID**

ID denotes the poll's 16-character identifier.

X-DoodleKey:**KEY**

KEY denotes the poll's 8-character key.

Status Code

201 (Created)

PUT (i.e., "save")

Request

Content type

application/xml

Resource

/polls/**ID**

ID denotes the poll's 16-character identifier.

Body

The poll encoded in XML according to schema poll.xsd.

Example:

```
<poll xmlns="http://doodle.com/xsd1">
<title>PPP</title>
<description>yum!</description>
<initiator>
<name>Paul</name>
</initiator>
<options>
<option>Pasta</option>
<option>Pizza</option>
<option>Polenta</option>
<option>P4</option>
</options>
</poll>
```

Header

X-DoodleKey:**KEY**

KEY denotes the poll's 8-character key.

When updating a poll, the poll state can be CLOSED or re-OPENed. Closing a poll also allows for picking final options by setting an option's "final" attribute to "true". Furthermore, it's possible to <hide> (true) or un-<hide> (false) the poll and partly modify its <extensions> (columnConstraint or rowConstraint; time-zone support (or lack thereof) cannot be changed).

Response

Status Code

200 (OK)

Errors

- 401: The keys do not match
- 502: Calendar synchronisation failed

DELETE

Request

Resource

/polls/ID

ID denotes the poll's 16-character identifier.

Header

X-DoodleKey:KEY

KEY denotes the poll's 8-character key.

Response

Status Code

204 (No Content)

Errors

- 401: The keys do not match
- 502: Calendar synchronisation failed

Participant

POST (i.e., "save as")

Request

Content type

application/xml

Resource

/polls/ID/participants

ID denotes the poll's 16-character identifier.

Body

The participant encoded in XML according to schema participant.xsd.

Example:

```
<participant xmlns="http://doodle.com/xsd1">
<name>Paul</name>
<preferences>
```

```
<option>1</option>
<option>1</option>
<option>0</option>
</preferences>
</participant>
```

A <participant>'s <name> is mandatory.

Response

Header

Content-Location:**ID**

ID denotes the participant's identifier.

Status Code

201 (Created)

Errors

- 400: Name is empty
- 409: Row constraint (ROWCONSTRAINT) exceeded
- 409: Column constraint (COLUMNCONSTRAINT) exceeded
- 502: Calendar synchronisation failed

PUT (i.e., "save")

Request

Content type

application/xml

Resource

/polls/**ID1**/participants/**ID2**

ID1 denotes the poll's 16-character identifier. **ID2** denotes the participant's identifier.

Body

The participant encoded in XML according to schema participant.xsd.

Example:

```
<participant xmlns="http://doodle.com/xsd1">
<name>Paul</name>
<preferences>
<option>1</option>
<option>1</option>
<option>0</option>
<option>0</option>
</preferences>
</participant>
```

Header (optional)

X-DoodleKey:**KEY**

KEY denotes the poll's 8-character key. Including the right key makes a difference when the poll is a hidden poll and/or a write-once poll.

Response

Status Code

200 (OK)

Errors

- 400: Name is empty
- 403: There is a user behind this participant
- 404: No such participant
- 409: Row constraint (ROWCONSTRAINT) exceeded
- 409: Column constraint (COLUMNCONSTRAINT) exceeded
- 502: Calendar synchronisation failed

DELETE

Request

Resource

/polls/**ID1**/participants/**ID2**

ID1 denotes the poll's 16-character identifier. **ID2** denotes the participant's identifier.

Header (optional)

X-DoodleKey:**KEY**

KEY denotes the poll's 8-character key. Including the right key makes a difference when the poll is a hidden poll and/or a write-once poll.

Response

Status Code

204 (No Content)

Errors

- 403: There is a user behind this participant
- 404: No such participant
- 502: Calendar synchronisation failed

Comment

POST (i.e., "save as")

Request

Content type

application/xml

Resource

/polls/**ID**/comments

ID denotes the poll's 16-character identifier.

Body

The comment encoded in XML according to schema comment.xsd.

Example:

```
<comment xmlns="http://doodle.com/xsd1">
<who>Paul</who>
<what>yada, yada, yada</what>
</comment>
```

Response

Status Code

201 (Created)

DELETE

Request

Resource

/polls/**ID1**/comments/**ID2**

ID1 denotes the poll's 16-character identifier. **ID2** denotes the comment's identifier.

Header

X-DoodleKey:**KEY**

KEY denotes the poll's 8-character key. Including the right key makes a difference when the poll is a hidden poll and/or a write-once poll.

Response

Status Code

204 (No Content)

Errors

- 403: There is a user behind this commentator
- 404: No such comment

User

Accessing users always requires authentication and authorization, even on the test server, and is documented in [AAforRESTfulDoodle.pdf](#).

Event

Accessing events always requires authentication and authorization, even on the test server, and is documented in [AAforRESTfulDoodle.pdf](#).