

# ds.store REST

## Description

Stores an object data in Gigya's Data Store (DS).

**Note:** If you plan on integrating the DS, we highly recommend reading the [Data Store Guide](#). The DS is part of the [Customer Identity](#) premium package that requires separate activation. If the DS is not part of your site package please contact your Gigya Customer Engagement Executive or email [support@gigya-inc.com](mailto:support@gigya-inc.com).

## Request URL

Where `<Data_Center>` is:

- `us1.gigya.com` - For the US data center.
- `eul.gigya.com` - For the European data center.
- `aul.gigya.com` - For the Australian data center.
- `ru1.gigya.com` - For the Russian data center.
- `cn1.gigya-api.cn` - For the Chinese data center.

If you are not sure of your site's data center, see [Finding Your Data Center](#).

## Parameters

Required	Name	Type	Description
----------	------	------	-------------

	data JSON object	<p>Specifies the data to store. The data object should be specified as a JSON object complying with the following restrictions:</p> <ul style="list-style-type: none"> <li>• A maximum size of 512K of data.</li> <li>• User defined field names must begin with a character and may contain characters, digits and underscores.</li> <li>• We highly recommend that you give every field a type specification by appending the type to the field name in the following manner: "_". Valid types are: <ul style="list-style-type: none"> <li>• i - an integer number (may contain 64bit long values as well), allows exact matching or range searches.</li> <li>• f - a floating point number, allows exact matching or range searches.</li> <li>• s - a string only allows exact matching.</li> <li>• t - a full text search field, allows for partial searching based on keywords.</li> <li>• b - Boolean.</li> <li>• d - a date field, allows exact matching or range searches. The valid date formats are: <ul style="list-style-type: none"> <li>• Standard ISO time format (like "2011-07-14T11:42:32.123+3", may be specified without a time at all or without a time zone)</li> <li>• "YYYY-MM-DD HH:MM:SS"</li> </ul> </li> </ul> </li> <li>• When the field type is not specified explicitly we will try to auto detect it. Fields that fail auto detection will be considered strings.</li> </ul> <p><b>Note:</b> When defining a 'binary' field type, it is not inferred from the data, and must be set explicitly using the 'setSchema' method.</p> <p>For example:</p> <pre>{   firstName_s : "david",   birthDate_d : "1978-07-22",   about_t : "I love tracking...",   academicDegree : {     university_s : "Berkeley",     degree_s : "MBA",     department_s : "Business",     courses : [{..}, {..},...]   },   hobbies_s : ["running", "music", "movies"] }</pre> <div style="border: 1px solid yellow; padding: 10px; margin-top: 10px;"> <p><b>Important</b> When saving data as a nested field, it is important to save the object in the format below:</p> <pre style="border: 1px dashed blue; padding: 10px; margin: 10px 0;">// Correct {   'a': {     'b': 'value'   } }</pre> <p>and not as:</p> <pre style="border: 1px dashed blue; padding: 10px; margin: 10px 0;">// Incorrect {   a.b: 'value' }</pre> <p>DS always returns an object as it was submitted, this is different than other Gigya schemas, which expand nested objects automatically.</p> </div>
type	string	A string indicating the type of the object, equivalent to a name classifying objects as being of the same type; used in searches and when applying schemas.
oid	string	A unique object identifier. If this parameter is set with the "auto" value, Gigya will generate a unique ID for the object.
UID	string	<b>A unique ID of a user. Setting this parameter</b> associates the data object with the specified user. The identifier of this object is the compound of <b>oid</b> and <b>UID</b> .
updateBehavior	string	This parameter defines how to handle conflicting updates. The parameter may receive one of the following values: <ul style="list-style-type: none"> <li>• "arrayPush" (default) - will cause arrays data provided with this API call to be appended to existing arrays data in the DB.</li> <li>• "arraySet" - will replace the content of existing arrays data in the DB with the values of arrays data provided with this API call.</li> <li>• "replace" - will cause the entire object data in the DB to be replaced with the new data provided with this call.</li> </ul>
format	string	Determines the format of the response. The options are: <ul style="list-style-type: none"> <li>• <i>json</i> (default)</li> <li>• <i>jsonp</i> - if the format is jsonp then you are required to define a <i>callback</i> method (see parameter below).</li> </ul>
callback	string	This parameter is <i>required</i> only when the <i>format</i> parameter is set to <i>jsonp</i> (see above). In such cases this parameter should define the name of the callback method to be called in the response, along with the jsonp response data.
context	string/JSON	This parameter may be used to pass data through the current method and return it, unchanged, within the response.
dontHandleScreenSet	Boolean	This parameter may be used in order to suppress the showing of screen-sets as a result of API calls. Default is <b>false</b> .
httpStatusCodes	Boolean	The default value of this parameter is <i>false</i> , which means that the HTTP status code in Gigya's response is always 200 (OK), even if an error occurs. The error code and message is given within the response data (see below). If this parameter is set to <i>true</i> , the HTTP status code in Gigya's response would reflect an error, if one occurred.

# Authorization Parameters

Each REST API request must contain identification and authorization parameters.

Some REST APIs may function without these authorization parameters, however, when that occurs, these calls are treated as **client-side** calls and all client-side rate limits will apply. In order to not reach client-side IP rate limits that may impact your implementation when using server-to-server REST calls, it is **Recommended Best Practice** to always sign the request or use a secret. A non-exhaustive list of REST APIs that this may apply to are as follows:

- accounts.login
- socialize.login
- accounts.notifyLogin
- socialize.notifyLogin
- accounts.finalizeRegistration
- accounts.linkAccounts

Please refer to the [Authorization Parameters](#) section for details.

## Response Data

Field	Type	Description
errorCode	integer	The result code of the operation. Code '0' indicates success, any other number indicates failure. For a complete list of error codes, see the <a href="#">Error Codes</a> table.
errorMessage	string	A short textual description of an error, associated with the errorCode, for logging purposes. This field will appear in the response only in case of an error.
errorDetails	string	This field will appear in the response only in case of an error and will contain the exception info, if available.
fullEventName	string	The full name of the event that triggered the response. This is an internally used parameter that is not always returned and <b>should not</b> be relied upon by your implementation.
callId	string	Unique identifier of the transaction, for debugging purposes.
time	string	The time of the response represented in ISO 8601 format, i.e., yyyy-mm-dd-Thh:MM:ss.SSSZ or
statusCode	integer	The HTTP response code of the operation. Code '200' indicates success. This property is deprecated and only returned for backward compatibility.
statusReason	string	A brief explanation of the status code. This property is deprecated and only returned for backward compatibility.
oid	string	The object's unique identifier. The value is either the <i>oid</i> parameter that you specified in the API call or a unique string generated by Gigya (if you have set the parameter to be "auto").

## Response Example

```
{
  "oid": "8b8972c3b3114db2a0acdfa2617693bf",
  "statusCode": 200,
  "errorCode": 0,
  "statusReason": "OK",
  "callId": "bcb794e7b8ca41b1b3baf99e7bc1c86a",
  "time": "2015-03-22T11:42:25.943Z"
}
```