

Profile REST

Description

This object represents the current state of the user's profile.

Use the [accounts.setSchema](#) command to require data in profile fields and to set client side access conditions.

The profile object is a JSON object holding the user's profile fields and is stored in the [User account](#) .

Gigya encrypts all [PII fields](#) by default, however, you can encrypt additional fields within a site's individual schema. Encrypted fields are stored and transmitted encrypted, Gigya manages the decryption of these fields transparently.

Individual fields can be encrypted via the [Screen-set UI Builder](#) or using the [accounts.setSchema](#) API.

Certain profile fields undergo *HTML sanitization*: city, country, firstName, lastName, nickname, photoURL, profileURL, thumbnailURL, zip, and more.

The following rules are applied to these fields:

1. Embedded Style Sheets are not allowed.
2. HTML comments are not allowed.
3. The following HTML tags are removed: script, noscript, iframe, frameset, frame, noframes, style (b, i, u, strong, br, and p tags are allowed). Div, em, blockquote, tt, quote, ecode, ul, ol and li tags are removed, but their child text is allowed.
4. Unknown HTML tags are removed.
5. Ampersand is encoded to an HTML entity.
6. Common HTML attributes are checked for a valid value: align, href, title, lang, nohref, rel, and nofollow, otherwise they are stripped.
7. In profile fields containing a URL: PhotoURL, ProfileURL and Thumbnail URL, unsafe characters " ' < > are URL encoded to %22 %27 %3C and %3E.

Data Availability

The following table represents the profile data returned within Gigya's User object for each social network provider. Note that not all fields are populated by social network data; some include data generated by Gigya. The availability of data from a specific network does not guarantee that the user has entered any information for that field or that the data can be retrieved without additional permissions granted by the social network. Program logic should be able to handle any missing data.

Notes:

- **Facebook** provides the "Current city" data via the "Location" field of the [Facebook user object](#), from which the state or country can be inferred.
- **PayPal** in the table refers to the OAuth implementation of Paypal. The OpenID implementation of PayPal has been deprecated.
- **LinkedIn** recently made changes to their permissions structure. As a result of those changes, much of the Extended Permissions data that apps were once able to receive now require becoming a LinkedIn Partner. The Gigya app utilized above does not have this 'Partner' status, so a number of fields in the above example will show up as *This data is provided by the social network but is not returned for this specific user profile*. Please view the [LinkedIn Developer Program Transition Guide](#) for more information.
- Slight differences exist between the User object returned by the client and the User object returned by the server. When retrieved from the **client**, missing properties will appear as empty strings and the identities and capabilities fields are returned as JSON objects. When retrieved from the **server**, only fields with data are returned, the identities field returns as an array and the capabilities field as a comma separated string.

Data Members

Field Name	Type	Description
firstName	string	The user's first name.
lastName	string	The user 's last name.
nickname	string	The user's nickname, this may be either the nickname provided by the social network provider or a c
address	string	The user's address.
age	integer	The age of the user. Note: This is a computed field that is updated by the Gigya service when are performed (such as accounts.setAccountInfo). This means that for son read only.
bio	string	The user's bio or a description of the person's professional profile (depending upon network).
birthDay	integer	The day in the month in which the user was born.
birthMonth	integer	The month in which the user was born.
birthYear	integer	The year in which the person was born.
capabilities	object	The capabilities of the user. The capabilities of the user are the unification of the capabilities of all th capabilities is indicating which operations are now available for the current user. The capabilities obj n, notifications, actions, friends, status, contacts and photos . Usage example: <pre>if (user.capabilities.friends) { //Do Something }</pre> Note: When retrieved from the server, capabilities are sent as a comma-s capabilities: " capabilities: Login, Friends, Notifications, Actions ".
certifications	Array of objects	Each object in the array includes the fields below: <ul style="list-style-type: none">• certifications.name - The user's certification name.• certifications.authority - The user's certification authority.• certifications.number - The user's certification number.• certifications.startDate - The user's certification start date.• certifications.endDate - The user's certification end date.
city	string	The user's city.
country	string	The user's country.
education	Array of objects	Each object in the array includes the fields below: <ul style="list-style-type: none">• education.school - The user's school name.• education.schoolType - The user's school type.• education.fieldOfStudy - The user's field of study.• education.degree - The user's degree.• education.startYear - The user's education start year.• education.endYear - The user's education end year.
educationLevel	string	The education level. Can be "Graduate school", "College", or "High-school".

email	string	<p>The user's email address.</p> <p>If additional/multiple email addresses are added (edited) to a user's account using <code>accounts.setAccountInfo</code> addresses will also be added to the <code>loginIDs.emails</code> array. To remove an email from the <code>loginIDs</code> array:</p> <pre>gigya.accounts.setAccountInfo({ "removeLoginEmails": "email" });</pre>
favorites	Array of objects	<p>Each object in the array includes the fields below *:</p> <ul style="list-style-type: none"> favorites.interests. <ul style="list-style-type: none"> favorites.interests.id - The ID of the user's favorite interest. favorites.interests.name - The name of the user's favorite interest. favorites.interests.category - The category of the user's favorite interest. favorites.activities. <ul style="list-style-type: none"> favorites.activities.id - The ID of the user's favorite activity. favorites.activities.name - The name of the user's favorite activity. favorites.activities.category - The category of the user's favorite activity. favorites.books.* <ul style="list-style-type: none"> favorites.books.id - The ID of the user's favorite book. favorites.books.name - The name of the user's favorite book. favorites.books.category - The category of the user's favorite book. favorites.music.* <ul style="list-style-type: none"> favorites.music.id - The ID of the user's favorite music. favorites.music.name - The name of the user's favorite music. favorites.music.category - The category of the user's favorite music. favorites.movies.* <ul style="list-style-type: none"> favorites.movies.id - The ID of the user's favorite movie. favorites.movies.name - The name of the user's favorite movie. favorites.movies.category - The category of the user's favorite movie. favorites.television.* <ul style="list-style-type: none"> favorites.television.id - The ID of the user's favorite television show. favorites.television.name - The name of the user's favorite television show. favorites.television.category - The category of the user's favorite television show.
followersCount	integer	The number of users following this user. This is the sum of all the users following this user, from all providers. If a user has no followers data in Twitter, this field will show "no data".
followingCount	integer	The number of users this user is following. This is the sum of all the users who this user is following supported. If a user has no followers data in Twitter, this field will show "no data".
gender	string	The gender of the user. Can be 'm', 'f', or 'u', for male, female, or unspecified.
hometown	string	The user's hometown.
honors	string	The list of honors of the user.
identities	dictionary object	<p>A dictionary object where the keys are the names of the providers to which the user is connected (e.g., 'facebook', 'twitter'). The value for each key is a dictionary object where the keys are the names of the providers to which the user is connected (e.g., 'facebook', 'twitter'). The value for each key is the ID of the user in that provider.</p> <p>For example: If the user is connected to Facebook then <code>user.identities['facebook']</code> is the identity object and <code>user.identities['facebook'].providerUID</code> is the ID of the user in Facebook.</p> <div style="border: 1px solid #add8e6; padding: 5px;"> <p>Notes:</p> <ul style="list-style-type: none"> When retrieved from the server this field will be an array and not a dictionary object. When used in <code>getFriendsInfo</code> this field contains only identities from the providers that both User A and User B are friends on Facebook but not on Twitter, then only User A's identities are returned. </div>
industry	string	The user's professional industry.
interestedIn	string	The preferred gender of the user's romantic interests.
interests	string	The user's interests.
isConnected	Boolean	Indicates whether the user is connected to any available provider. This is true only if the user isSiteLinked to any provider.

iRank	decimal	<p>Influencer rank. The iRank is a number between 0-99, which denotes the percentile location of the user. If a user's iRank equals 60, this means that 60% of the site users influence less than this user, or in other words, 60% of the site users have a lower iRank than this user. The iRank is calculated based on the amount of exposure this user provides the site. The calculator uses the following factors to determine the iRank:</p> <ul style="list-style-type: none"> • The number of friends this user has in all the networks to which he is connected through this site. • The number of times this user shared something in this site (per month). • The number of click backs that were made as a result of this user's shares. <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p>Note:</p> <ul style="list-style-type: none"> • The iRank is not available in the User object by default. To retrieve the <i>extraFields</i> parameter. Include the iRank in the list of extra fields to retrieve. • Influencer ranks may also be retrieved through our Influencers Report. </div>
isSiteUID	Boolean	<p>This field indicates if the UID is a site UID (provided by your site) or an ID generated by Gigya. Your site may provide the site UID using the <code>socialize.setUID</code>, <code>socialize.notifyRegistration</code> or <code>socialize.setSiteUID</code> methods.</p> <p>The value true indicates the UID has been provided by your site, <i>not</i> generated by Gigya.</p>
isSiteUser	Boolean	Indicates whether the current user is a user of the site. This is useful for checking which friends of the user are also users of the site.
languages	string	A comma-separated list of languages that the person speaks.
likes	Array of objects	<p>Each object in the array includes the fields below: *</p> <ul style="list-style-type: none"> • likes.name - The name of the user's like. • likes.category - The category of the user's like. • likes.id - the identifier of the user's like. • like.time - the time the like was created in UTC standard <i>ISO 8601</i> format (i.e., yyyy.mm.dd.Th). • like.timestamp - the time the like was created in Unix time format (i.e., the number of seconds since January 1, 1970).
locale	string	The user's locale, as received from the social network or as it was entered during an account import.
loginProvider		<p>The name of the provider that the user used in order to log in. If the user logged in using your site but not logged in, then the value of this parameter would be an empty string.</p> <p>If the value of this parameter is not an empty string then <code>user.identities[user.loginProvider]</code> is the identity object for the provider.</p>
loginProviderUID	string	If the user is logged in, this field will hold the user's ID from the login provider. For example, if the user is logged in through Facebook, this field will hold the user's Facebook ID.
oldestDataAge	integer	The difference in seconds between the oldestDataUpdatedTimestamp and the current time.
oldestDataUpdatedTimestamp	integer	The time when the oldest data of the object was refreshed, in Unix time format, i.e., seconds since January 1, 1970.
patents	Array of objects	<p>Each object in the array includes the fields below:</p> <ul style="list-style-type: none"> • patents.title - The user's patent title. • patents.summary - The user's patent summary. • patents.number - The user's patent number. • patents.office - The user's patent office. • patents.status - The user's patent status. • patents.date - The user's patent date. • patents.url - The user's patent URL.
phones	Array of objects	<p>Each object in the array includes the fields below:</p> <ul style="list-style-type: none"> • phones.type - The user's phone type (string). • phones.number - The user's phone number (string).
photoURL	string	The URL of the user's full-size photo.
politicalView	string	The user's political view.
professionalHeadline	string	The user's professional headline.
profileURL	string	The URL of the user's profile.
providers	array of strings	Contains the names of the providers to which the user is connected/logged in.

publications	Array of objects	<p>Each object in the array includes the fields below:</p> <ul style="list-style-type: none"> • publications.title - The user's publication title. • publications.summary - The user's publication summary. • publications.publisher - The user's publication publisher. • publications.date - The user's publication date. • publications.url - The user's publication URL.
relationshipStatus	string	The user's relationship status.
religion	string	The user's religion.
samlData	object	A Custom SAML data object. When mapping IdP attributes to identity fields, we allow mapping to a field mapped to this custom identity field by mapping it to a name like "samlData.myField".
signatureTimestamp	integer	If the global configuration object 's "signIDs" field is set to 'true', Gigya "signs" the User object (when fraud. This field will hold the timestamp that should be used for verification * . The timestamp is the time since Jan. 1st 1970).
skills	Array of objects	<p>Each object in the array includes the fields below:</p> <ul style="list-style-type: none"> • skills.skill - The user's skill. • skills.level - The level of the user's skill. • skills.years - The years of the user's skill.
specialities	string	The user's professional specialities.
state	string	The user's state.
timezone	string	The user's timezone.
thumbnailURL	string	The URL of the user's thumbnail image (when available). The following unsafe characters in the UR
UID	string	<p>A unique ID for the person represented by this object. The UID is returned by login or notifyLogin and is consistent over time for that person (note that if notifyLogin was used, the UID contains the same methods that require you to specify a user.</p> <p>Gigya uses digital signatures to transmit the UID. See Validate the UID Signature in the Social Log</p> <p>The UID is case-sensitive, so that, for example, two UIDs that differ only by one character being up integration depends on using case-insensitive UIDs, please discuss the issue with your Implementa</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Notes:</p> <ul style="list-style-type: none"> • UIDs are not globally unique and may be set to a custom unique string method. I.e., every site can set its own UIDs and different sites may ch • The UID value may be long, and if you plan to store it in a database, th • The UID string must be encoded using the encodeURIComponent() fu </div>
UIDSignature	string	If the global Configuration object 's "signIDs" field is set to 'true', Gigya "signs" the User object (when fraud. This field will hold the signature , that should be used for verification * .
username	string	The user's username.
isVerified	boolean	If the user's Gigya account is considered verified.
verified	string/boolean	<p>When returned as part of the Profile object, it is the date and time the account was verified in ISO 8601 format. If the user's email address has been verified by the social network (boolean).</p> <p>▼ Response Example</p> <pre style="border: 1px dashed #ccc; padding: 10px; margin-top: 10px;"> { "UID": "_guid_UJM...Hi0=", "UIDSignature": "T7x70...T2I=", "signatureTimestamp": "1479802948", "loginProvider": "facebook", "isRegistered": true, "isActive": true, "isLockedOut": false, </pre>

```
"isVerified": true, // whether the user's Gigya a
"socialProviders": "amazon,facebook,googleplus,li
"identities": [
  {
    "provider": "facebook",
    "providerUID": "1443...82",
    "isLoginIdentity": true,
    "photoURL": "https://graph.facebook.com/v2.5/
    "thumbnailURL": "https://graph.facebook.com/v
    "firstName": "...",
    "lastName": "...",
    "gender": "m",
    "email": "...@....com",
    "profileURL": "https://www.facebook.com/app_s
    "proxiedEmail": "",
    "allowsLogin": true,
    "isExpiredSession": false,
    "verified": "true", // Whether the user's ema
    "lastUpdated": "2016-11-21T14:21:46.850Z",
    "lastUpdatedTimestamp": 1479738106850,
    "oldestDataUpdated": "2016-11-21T14:21:46.272
    "oldestDataUpdatedTimestamp": 1479738106272
  }
],
"data": {
  "subscribe": false
},
"password": {},
"created": "2015-08-25T14:08:35.481Z",
"createdTimestamp": 1440511715481,
"lastLogin": "2016-11-21T14:21:46.303Z",
"lastLoginTimestamp": 1479738106303,
"lastUpdated": "2016-11-21T14:21:46.850Z",
"lastUpdatedTimestamp": 1479738106850,
"oldestDataUpdated": "2016-04-05T15:11:38.014Z",
"oldestDataUpdatedTimestamp": 1459869098014,
"registered": "2015-08-25T14:08:35.544Z",
"registeredTimestamp": 1440511715544,
"verified": "2015-08-25T14:08:35.497Z", // Time t
"verifiedTimestamp": 1440511715497, // Unix times
"statusCode": 200,
"errorCode": 0,
"statusReason": "OK",
```

```

    "callId": "d1b...78a",
    "time": "2016-11-22T08:22:28.227Z"
  }

```

verifiedTimestamp	integer	The time the user's Gigy account was verified in Unix format (i.e., seconds since January 1st, 1970).
work	Array of objects	Each object in the array includes the fields below: <ul style="list-style-type: none"> • work.company - The company where the user works/worked. • work.companyID - The company's ID. • work.title - The user's title in the company. • work.companySize - The company's size. • work.startDate - The date the user started working at the company. • work.endDate - The date the user stopped working at the company. • work.industry - The industry of the company. • work.isCurrent - Indicates whether the user is still working at the company.
zip	string	The user's zip code.

* To verify the authenticity of the User Object, follow the instructions in the section [Signature Validation Process](#), using the **UID**, **UID Signature** and **signatureTimestamp** fields.

- The profile information is returned only for providers who support it. If a field doesn't exist for a chosen provider, the response will be returned without this field. For information on making direct API calls to a specific provider, please see [Direct API Calls to Providers](#).
- * We limit the response for this parameter to only retrieve the first 500 objects.
- Not all available fields are returned by default. Please see the applicable API documentation for any optional parameters required for returning extra data fields such as a user's **likes** or complete **Identities** data, for example, [accounts.getAccountInfo](#) or [socialize.getUserInfo](#).