

PG Dating Pro. Mobile API description.

General information

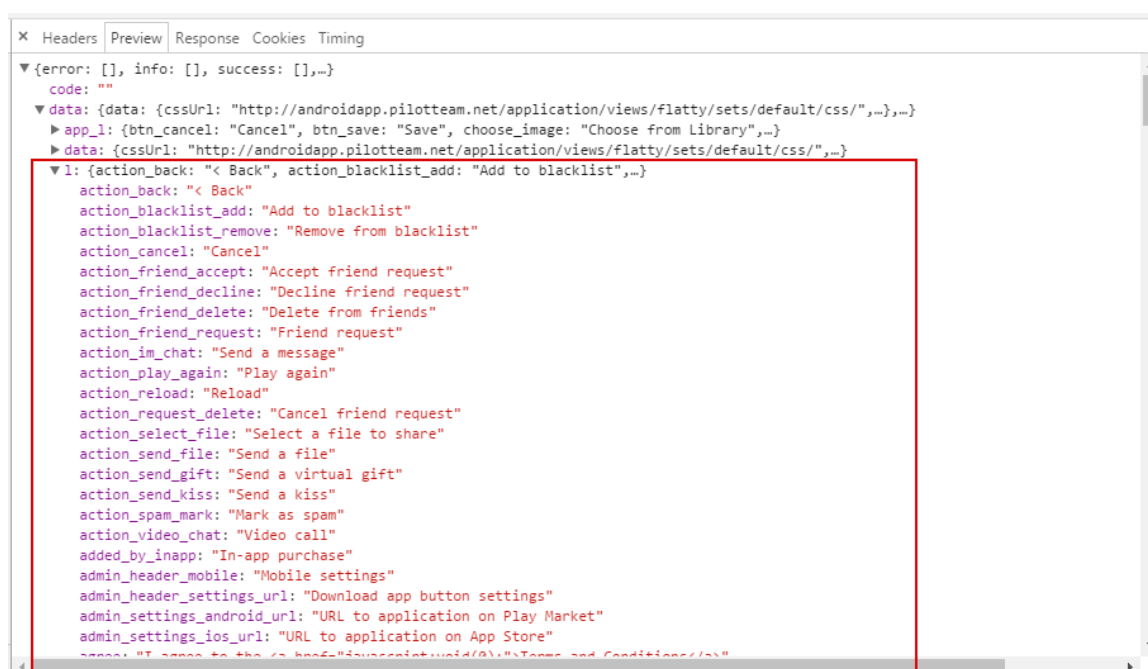
Communication with the site API occurs through asynchronous queries similar to <http://androidapp.pilotteam.net/api/users/search> with POST data or without them. Where 'androidapp.pilotteam.net' is the address of our dating site, 'api' means that an API method is called, 'users' is the name of the module, and 'search' is the API method that is called. Depending on the method, we send some POST data and receive the response in JSON format.

Connection

For the purpose of connection between the app and the site, it is necessary to request a token using the http://androidapp.pilotteam.net/api/get_token method. If the token is already in place, initialize using the <http://androidapp.pilotteam.net/api/mobile/init/1> method that includes the post fields:

token and **is_app: true**.

'1' after init/ is the language id that will be returned after the init query. This is the language that we will receive messages in as well as the texts of the app. This is an example of how it looks like in JSON (in "key": "text" format):



Authorization

For an authorized user we need to get a new token, also with the method http://androidapp.pilotteam.net/api/get_token email=ivan@mail.com&password=123456, where email and password stand for the email and password of the user in question (POST data). In response, we get the token and the access details of our user.

```
X Headers Preview Response Cookies Timing
▼ {error: [], info: [], success: [],...}
  code: ""
  data: {token: "72a059c0129e84fa83c31bb0d84e7329",...}
    token: "72a059c0129e84fa83c31bb0d84e7329"
    ▶ user_data: {account: "190", activated_end_date: "2018-01-14 08:40:01", activity: "1", address: "", age: "26",...}
  error: []
  errors: ""
  info: []
  js_events: false
  messages: ""
  success: []
  ▶ user_session_data: {session_id: "a06c3fb02eb03c843329b4dc84d9bfa3", ip_address: 1540553950,...}
```

Then follows initialization - <http://androidapp.pilotteam.net/api/mobile/init/1>

Every site query should contain POST data with the token on top of other info.

This is a sample response to different queries:

```
X Headers Preview Response Cookies Timing
▼ {error: [], info: [], success: [],...}
  code: ""
  data: {data: {cssUrl: "http://androidapp.pilotteam.net/application/views/flatty/sets/default/css/",...},...}
    error: []
    errors: ""
    info: []
    js_events: false
    messages: ""
    success: []
  ▶ user_session_data: {session_id: "e359ead8643f1ee5e044f2c341e18aa3", user_data: "a:1:{s:8:'token_id';s:3:'238';}",...}
```

Here, in the 'data' field we see the data that we expect from the method. **messages**, **error**, **errors**, **success**, **info** are various messages that the query can return (in the case of success or failure). **user_session_data** is also present in each response and contains details about our authorized user.

```
X Headers Preview Response Cookies Timing
▼ {error: [], info: [], success: [],...}
  code: ""
  data: {indicators: {newMessages: "0", newFriends: 2}}
  error: []
  errors: ""
  info: []
  js_events: false
  messages: ""
  success: []
  ▶ user_session_data: {session_id: "d32e84aafa2956376073e2180f4d4de0", user_data: "a:1:{s:8:'token_id';s:3:'238';}",...}
    activity: "1"
    age_max: "49"
    age_min: "22"
    approved: "1"
    auth_type: "user"
    email: "will@mail.com"
    fname: "William"
    lang_id: "1"
    last_activity: 1490278653
    logo: "http://androidapp.pilotteam.net/uploads/user-logo/0/0/0/16/small-ee5136d1ce.jpg"
    name: "William Elk"
    nickname: "William"
    online_status: "1"
    output_name: "William Elk"
    session_id: "d32e84aafa2956376073e2180f4d4de0"
    show_adult: "0"
    site_status: "0"
    sname: "Elk"
    user_data: "a:1:{s:8:'token_id';s:3:'238';}"
    user_id: "16"
    user_type: "male"
```

This is what the complete response to an init-query looks like: [\[init_log.php\]](#)

array [l] is the list of application texts depending on chosen language.
[language] stands for the chosen language, [languages] – the list of active languages.
[userData] – current user's details.
[properties] – various settings ([userTypes] – available types of users, [age] => [min], [max] – minimum and maximum age, [currency] – current site currency).

Backend query

For an authorized user, it is necessary to send a backend query every few seconds in order to get various data (from new friendship requests to incoming messages, etc.):

<http://androidapp.pilotteam.net/api/start/backend>

POST-data for the query:

```
token=4894fec4fc1357fbfaf3985386c6fd5c&data[indicators][indicators][0]=new_messages&data[indicators][indicators][1]=new_friends&data[indicators][indicators][2]=new_kisses&data[indicators][indicators][3]=new_gifts&data[indicators][indicators][4]=new_guests&data[indicators][indicators][5]=new_files&data[indicators][indicators][6]=new_chats&data[indicators][method]=get_indicators&data[indicators][model]=mobile_model&data[indicators][module]=mobile&data[user][method]=get_account&data[user][model]=users_model&data[user][module]=users&data[user][user][0]=get_account
```

If there are any new events they are sent back to us in the following form:



In the image above we see that the user has 2 friendship requests. Accordingly, we display the indicator in the menu, next to 'Friends'.

This is the complete response that we receive from the backend: [\[backend_response.php\]](#)
[account] is the number of money on user's account.

Users list

This is the query to receive the list of users: <http://androidapp.pilotteam.net/api/users/search>

POST-data for the query:
items_on_page=12&page=1&token=7a96dde061b00984a6ea6dacd0e2ccd3

or a more detailed query according to criteria:

items_on_page=12&id_country=US&id_city=5&user_type=female&online_status=1&id_region=5&age_min=18&page=1&age_max=30&token=7a96dde061b00984a6ea6dacd0e2ccd3

We receive this array in response: [\[users_search_list.php\]](#)

[users] in the array includes the list of users with information about each user.

The total number of displayed users can be collected from the field `[page_data] => [total_rows]`.

These are the links to user's profile image: `[media] => [user_logo] => [thumbs] =>` different sizes from small to grand. This is where the profile picture (avatar) comes from: `[big]` (size 200*200).

Username is stored in the `[output_name]` field.

Mind the user's `[id]` to be able to go to the user's profile.

User profile

This is the query to receive a user's profile: <http://androidapp.pilotteam.net/api/users/view>

POST-data: id=15&token=7a96dde061b00984a6ea6dacd0e2ccd3

This is the response: [\[view_profile_log.php\]](#)

Here the **[user]** array contains all the user's details.

In the **[sections]** array we find additional information (field editor).

In `[sections]` we are interested in the following fields: `[name]` and `[value]` or `[value_str]` (if applicable).

This is the main information about user that we can collect from the `[user]` array: `[output_name]`, `[age]`, `[looking_user_type_str]`, `[age_min]`, `[age_max]`, `[location]`, and the user logo (avatar).

Friends list status

Along with the user's profile info, when we view someone's profile, we receive statuses that determine the possibility of various actions (e.g. send a friendship request, reject an incoming friendship request, confirm an incoming request, revoke an outgoing friendship request, and remove from the friends list).

This is the query: http://androidapp.pilotteam.net/api/friendlist/get_statuses

id_dest_user=56&token=c69128b08157ee66e4485a1807d587eb

where **id_dest_user** is the id of user's profile we are viewing.

This is the response: [\[friend_statuses_log.php\]](#)

Pay attention at the **[allowed_btns]** array. If `[allow]` field is set to 1, it means that this action can be performed.

My profile

The workings are the same as with another user's profile, see [User profile](#). The only difference is that you should put the current user's id into the id field. Also mind the profile editing option.

Editing the additional profile fields

Receiving details: <http://androidapp.pilotteam.net/api/users/profile>

lang_id=1§ion=about-me&id=15&token=c69128b08157ee66e4485a1807d587eb
where **lang_id=1** is the code of the language that the information will be received in, and
section=about-me is the gid of a field editor section

This is the response: [[profile_section_log.php](#)]

Here the [**fields_data**] array contains different fields that can be changed in the control panel of site (such as weight, height, hobbies, etc.). The fields types can be: select, multiselect, text, checkbox. The field type is designated in [field_type]. The list of available answer options is available in the [options] array of the element.

Saving profile info after editing (additional info)

<http://androidapp.pilotteam.net/api/users/save>

section=about-

me&id=15&token=c69128b08157ee66e4485a1807d587eb&fe_about_me=&fe_body=6&fe_hair=4&fe_height=4&fe_style[0]=1&fe_style[1]=2&fe_style[2]=3&fe_style[3]=4&fe_style[4]=8&fe_weight=3

Where does **fe_body** come from? **fe_** means that the field to be saved comes from the field editor, and **body** is the gid of the field.

Saving profile info after editing (main info)

<http://androidapp.pilotteam.net/api/users/save>

POST-data:

looking_user_type=male&fname=Anna&id_region=2&age_min=23§ion=personal&token=dea9aef1cb879b16c705682bc3640c6f&id_country=DE&id_city=2&sname=Crawford&nickname=Annie&id=15&age_max=50

section=personal means that we are editing the main profile information.

Retrieving location data

Location data is required to search for users and to edit the profile of the current user.

List of countries: http://androidapp.pilotteam.net/api/countries/get_countries
token=dea9aef1cb879b16c705682bc3640c6f

List of regions in a country: http://androidapp.pilotteam.net/api/countries/get_regions?country_code=DE&token=dea9aef1cb879b16c705682bc3640c6f

List of cities in a region: http://androidapp.pilotteam.net/api/countries/get_cities?region_id=2&token=dea9aef1cb879b16c705682bc3640c6f

These are the responses: [[countries_log.php](#)], [[regions_log.php](#)], [[cities_log.php](#)]

Friends list

Query: <http://androidapp.pilotteam.net/api/friendlist/index>

POST-data: page=1&token=7a96dde061b00984a6ea6dacd0e2ccd3&formatted=true

Response: [[friendlist_log.php](#)]

And this is the query used to retrieve the incoming friendship requests:

http://androidapp.pilotteam.net/api/friendlist/friends_requests

POST-data: page=1&token=dea9aef1cb879b16c705682bc3640c6f&formatted=true

Response: [[friends_requests_log.php](#)]

The total amount of friends and friendship requests are returned with the following queries:

http://androidapp.pilotteam.net/api/friendlist/friendlist_count

http://androidapp.pilotteam.net/api/friendlist/friends_requests_count

Sending a friendship request:

<http://androidapp.pilotteam.net/api/friendlist/request> id_dest_user=56&comment=testing comment&token=dea9aef1cb879b16c705682bc3640c6f

Canceling one's own friendship request:

<http://androidapp.pilotteam.net/api/friendlist/remove>

id_dest_user=56&token=dea9aef1cb879b16c705682bc3640c6f

Accepting friendship request

<http://androidapp.pilotteam.net/api/friendlist/accept>

id_dest_user=8&token=dea9aef1cb879b16c705682bc3640c6f

Declining friendship request

<http://androidapp.pilotteam.net/api/friendlist/decline>

id_dest_user=1&token=dea9aef1cb879b16c705682bc3640c6f

Removing someone from friends list:

<http://androidapp.pilotteam.net/api/friendlist/remove>

id_dest_user=8&token=dea9aef1cb879b16c705682bc3640c6f

My guests

Query: http://androidapp.pilotteam.net/api/users/my_guests

page=1&token=73cdfaefd2ed1ef21388b275b4d43d5f

Response: [[my_guests_log.php](#)]

[**users**] is the array of users, [view_dates] is the last time your profile was visited by user id; [page_data] => [total_rows] is the number of all guests.

IM chat

IM chat messages

Retrieving the contacts list: http://androidapp.pilotteam.net/api/im/get_contact_list

POST-data: token=73cdfaefd2ed1ef21388b275b4d43d5f&with_messages=true

Response: [[contact_list_log.php](#)]

[**list**] stands for the list of contacts; [site_status] => 1 means that the user is online; [count_new] is the number of unread messages; [contact_user] is the contact's details.

[im_messages] is the latest messages from the contacts where the array key is **id_contact**.

IM chat window (dialogue)

Retrieving messages: http://androidapp.pilotteam.net/api/im/get_messages

count=10&id_contact=1&token=73cdfaefd2ed1ef21388b275b4d43d5f

Response: [[get_messages_log.php](#)]

If the chat window is open we should add additional fields into our backend query:

data[im][contact_id]=1&data[im][from_id]=41&data[im][method]=getNewMessages&data[im][model]=im_messages_model&data[im][module]=im

data[im][from_id]=41 is the current user's id

data[im][contact_id]=1 is the contact's id (id of the user with whom the current user is chatting).

Let's set the frequency of the backend query to, say, 2 seconds. New messages will come as the response: [[backend_message_log.php](#)]

The query will look something like this: <http://androidapp.pilotteam.net/api/start/backend>

token=73cdfaefd2ed1ef21388b275b4d43d5f&data[im][contact_id]=1&data[im][from_id]=41&data[im][method]=getNewMessages&data[im][model]=im_messages_model&data[im][module]=im&data[indicators][indicators][0]=new_messages&data[indicators][indicators][1]=new_friends&data[in

data[indicators][2]=new_kisses&data[indicators][3]=new_gifts&data[indicators][4]=new_guests&data[indicators][5]=new_files&data[indicators][6]=new_chats&data[indicators][method]=get_indicators&data[indicators][model]=mobile_model&data[indicators][module]=mobile&data[user][method]=get_account&data[user][model]=users_model&data[user][module]=users&data[user][user][0]=get_account

Kisses

Query for retrieving the list of kisses to send (kisses are emoji-like pictures):

<http://androidapp.pilotteam.net/api/kisses/getKisses/3>

token=73cdfaefd2ed1ef21388b275b4d43d5f

Here /3 means the id of the user who we send a kiss to.

Response: [\[get_kisses_log.php\]](#)

The names of the files are listed in the [image] field. Take the full path to the images folder

<http://androidapp.pilotteam.net/uploads/kisses-file/> and insert the file's name here.

After the user selects a kiss to send, s/he taps the Send button and the following query is sent:

<http://androidapp.pilotteam.net/api/kisses/setKisses/>

POST-data: kiss=4&message=test

comment&token=678298af3c2693584e22917571a5a597&object_id=56

Where kiss=4 is the kiss's id,

message is an accompanying comment to the kiss from the sender,

object_id is the id of the recipient.

The response comes in the form of a message that is contained in the message field in the body of the response.

Lists of incoming and outgoing kisses

<http://androidapp.pilotteam.net/api/kisses/index/outbox/1>

<http://androidapp.pilotteam.net/api/kisses/index/inbox/1>

Here **inbox** or **outbox** stand for the incoming or outgoing kisses, and /1 is a page number.

Response: [\[kisses_list_log.php\]](#)

[list] is the array with the kisses list,

[items_on_page] is the number of kisses per page,

[count] – the total number of kisses (received or sent).

Virtual gifts

Receiving a list of gifts to send: http://androidapp.pilotteam.net/api/virtual_gifts/getGifts/56

token=678298af3c2693584e22917571a5a597

Here /56 is the id of the user who we send a gift to.

Response: [get_gifts_log.php]

[price] => 1.0000 and [fk_currency_gid] => USD are the currency and the cost of the gift that will be deducted from the sender's account. A price is displayed along with each gift.

Sending a gift:

http://androidapp.pilotteam.net/api/virtual_gifts/sendGiftPayment/56/2

is_private=1&comment=test comment&token=678298af3c2693584e22917571a5a597

/56 is the recipient's id,

/2 the gift id,

is_private=1 means that gift will only be visible to the recipient. **is_private=0** will mean that everyone will be able to see the gift. The feature can be implemented as a checkbox in the process of selecting a virtual gift.

comment is an accompanying comment by the sender.

The response comes in the form of a text message in the message field (success or failure).

List of received gifts

http://androidapp.pilotteam.net/api/virtual_gifts/getUserGifts/1

token=c69128b08157ee66e4485a1807d587eb

/1 is the number of a gift page

Response: [gifts_list_log.php]

[is_new]=>1 means that the gift is new and not yet accepted or rejected by the recipient. The latter can either accept or reject the gift with the following query:

http://androidapp.pilotteam.net/api/virtual_gifts/userGiftStatus/7/approved

token=c69128b08157ee66e4485a1807d587eb

Here /7 is the gift id,

approved (or **decline**) is the available action: accept or reject.

The response comes in the form of a text message in the message field.

Services

Getting the list of available paid services:

http://androidapp.pilotteam.net/api/services/buy_list

token=c69128b08157ee66e4485a1807d587eb

Response: [[services_buy_list_log.php](#)]

[is_purchased] => 1 means that the service is purchased by the user, (0 means that it's not).

In order to display the list of paid services on the screen, we need the following fields: [name], [description] and [price]. The [gid] field is necessary to proceed to the Buy service screen.

Buy service screen

<http://androidapp.pilotteam.net/api/services/get>

gid=users_featured&token=c69128b08157ee66e4485a1807d587eb

Insert the service gid here that we obtained from the list earlier.

Response: [[buy_service_screen_log.php](#)]

Purchasing a service

<http://androidapp.pilotteam.net/api/services/form>

payment_type=account&service_gid=users_featured&token=c69128b08157ee66e4485a1807d587eb&activate_immediately=true

payment_type=account is a mandatory field. It means that the money will be written off from the user's account.

activate_immediately=true means that the service will be activated immediately (the user can choose to activate a purchased service later).

List of purchased services

<http://androidapp.pilotteam.net/api/services/my>

token=c69128b08157ee66e4485a1807d587eb

Response: [[my_services_log.php](#)]

Membership groups (Access permissions)

Retrieving the list of groups: http://androidapp.pilotteam.net/api/access_permissions/index

token=c69128b08157ee66e4485a1807d587eb

Response: [[list_groups_log.php](#)]

[groups] is groups list,
[access] is the list of available features (modules) in each group,
[is_purchased] – if a value is specified for this field, it means that the group is purchased.

Buy membership group screen

http://androidapp.pilotteam.net/api/access_permissions/groupPage/silver

token=c69128b08157ee66e4485a1807d587eb

silver is the gid of a group to be purchased.

Response: [[select_group_log.php](#)]

A user has an opportunity to choose the activity period for the selected membership group. Accordingly, the cost of the membership changes depending on the selected period [**periods**].

Purchasing a membership group

http://androidapp.pilotteam.net/api/access_permissions/payment

pay_system_gid=account&group_gid=silver&period_id=1&send=1&token=c69128b08157ee66e4485a1807d587eb

pay_system_gid=account and **send=1** are required parameters for a valid query,
group_gid=silver is the gid of the group being purchased, and
period_id=1 is the id of the group activity period.

The response comes in the form of a text message in either 'success' or 'failure' response field.

This is how the membership groups work. The site owner sets up access to different modules for different groups, from the administration panel of the site. In case a user is a member of a group that has no access to a certain module, an HTTP query will return the following:

```
['info' => ['access_denied' => "Please upgrade your membership in order to get access to this section "]]
```

Based on the **access_denied** key, you can redirect the user to the Buy membership group screen.

Blacklist

Adding a user to a blacklist: <http://androidapp.pilotteam.net/api/blacklist/add/56>

token=c69128b08157ee66e4485a1807d587eb

/56 is the user's id.

The response comes in the form of a text message in a message field.

Retrieving the list of blacklisted users: <http://androidapp.pilotteam.net/api/blacklist/index>

page=1&token=c69128b08157ee66e4485a1807d587eb&formatted=true

Response: [[blacklist_log.php](#)]

Deleting a user (users) from the blacklist: <http://androidapp.pilotteam.net/api/blacklist/remove>

token=c69128b08157ee66e4485a1807d587eb&id_dest_user[0]=40&id_dest_user[1]=56

id_dest_user is an array made up of users' ids.

Report abuse (Mark as spam)

Receiving the list of possible reasons to report someone or something:

http://androidapp.pilotteam.net/api/spam/get_reasons

token=c69128b08157ee66e4485a1807d587eb

Response: "reasons":{"header":"Reason for reporting","option":{"1":"Inappropriate content","2":"Suspicious information","3":"Spam"}}}

Sending a report: http://androidapp.pilotteam.net/api/spam/mark_as_spam

type_gid=users_object&object_id=56&token=c69128b08157ee66e4485a1807d587eb&data[messag e]=spam text message&data[reason_id]=1

type_gid=users_object means that we are reporting a user,

object_id=56 is the user's id who we are reporting,

data[message] is the message from the reporter, attached to the report,

data[reason_id] is the id of the reason that we received by the **get_reasons** method.

We receive a response in the form of a test message in the messages field ("messages": "Report is sent, thank you").