



Numerology Application API

This API document provides you with a comprehensive breakdown of the fields and parameters required to obtain information from the server for the execution of a numerology request.

Should you experience any errors with the API, or if think there are bugs in the response, then do your best to work around these and document your issues in a “api-errors.txt” file that is included within your submission.

The communications target for all API calls is:

<http://numerology.kamkash.com/comms.php>

All queries can be executed through GET or POST to the target URL as name, value pairs as regular form posted parameters. For example:

<http://numerology.kamkash.com/comms.php?action=AppOpen&AppID=0>

All query responses will be in JSON and always contain a variable titled “status” if the server has at minimum received the query. If “status” is not returned, it should be assumed that the query was not properly received by the server. For example, for the above query, the server would respond in the format:

```
{"status": "OK", "retID": 75}
```

The “status” variable indicates the success of the query execution at the highest level. If the purpose of the query is achieved on the server side, then “status” will be set to:

OK

else the “status” variable will be set to:

FAIL

the response may contain additional information, those parameters are defined below.

Note that the names within the Input/Output columns are the case-sensitive names of the variables that are either expected by the server or returned by the server. All fields are mandatory unless specified otherwise.

Method: APPLICATION_OPEN

Purpose: This method informs the server that an application has been launched on a mobile. If the application is being launched for the first time on a device, then the method provides a unique identifier for the application to store and to submit for all future transactions. If the application has been opened before, it expects to receive an existing unique ID that has been stored by the device.

Inputs

action	This should equal "AppOpen"
AppID	If the application is being launched for the first time, or no AppID can be found in the local datastore, then this should be "0" (zero). Else, the existing AppID should be submitted
AppVersion (<i>optional</i>)	Submit the application version code, only required to be used when launched for the first time
ScreenWidth (<i>optional</i>)	Detect and return the device width in px, only required to be used when launched for the first time
ScreenHeight (<i>optional</i>)	Detect and return the device height in px, only required to be used when launched for the first time
AppOS (<i>optional</i>)	Detect and submit the application OS and version, only required to be used when launched for the first time

Outputs

status	If "OK" then the open request was successfully processed. Otherwise, the AppID may not exist, or there was a database error
retID	If "0" (zero) was sent then a unique ID that is to be stored on the device is returned, else the AppID submitted as part of the inputs is returned

Method: CALCULATION REQUEST

Purpose: This method request the server for a preliminary numerology report based on user inputs. Details are submitted and the outputs from this method are expected to be displayed directly to the user.

Inputs

action	This should equal "NumerologyCalculate"
AppID	This should be the stored AppID as returned from the APPLICATION_OPEN method
personName	The person's full name as a string
personDay	The day of the month the person was born as an integer between 1 and 31
personMonth	The month of the year the person was born as an integer between 1 and 12
personYear	The year the person was born as an integer above 1900

Outputs

status	If "OK" then the calculation request was successfully processed and there is data to use below. Otherwise, the AppID may not exist, there may be an irregularity in the date of birth format, or there was a database error
lifeNumber	An integer between 1 and 9
dayNumber	An integer between 1 and 9
nameSuitable	A boolean to indicate if the current name is suitable for the user given their date of birth
feedbackText	<p>Instant high level assessment and feedback of the date birth and name. Presented as a dynamic length array of JSON objects. If everything was processed correctly, the server will respond with the multiple JSON objects in an array:</p> <p>SectionTitle=> The title of the section SectionText=> The text associated with the title</p> <p>The display should clearly distinguish sections with titles displayed differently to body text.</p> <p>If processing failed, then a single string will be provided as feedbackText to provide insight into the cause of the error</p>

Method: GET GURUS

Purpose: This method allows for the application to query for a list of the gurus within our network. The returned result should consist of a full list of gurus, with names, descriptions and photo URLs. No real input is necessary for this command.

Inputs

action	This should equal "GetGurus"
AppID	This should be the stored AppID as returned from the APPLICATION_OPEN method

Outputs

status	If "OK" then the database query completed successfully and the full list of gurus and details have been supplied in the guruList. If false, then there was a database error and the command should be tried again.
guruList	This will contain an array that will be dynamic in size depending on the number of gurus that join the network. The data will be returned as an array of JSON objects described as follows: guruID => The database entry ID for the guru (key field) guruName => Name of guru guruPhoto => Online URL to a JPG of the Guru guruDescription => A brief description (dynamic length) of the guru's speciality and history guruCity => City name as a string (where the guru is located) guruCountry => Country name as a string

Method: PURCHASE COMPLETED

Purpose: This method is to be called only once the market/app-store purchase process has been completed. It is used to inform our servers of the successful purchase and trigger the workflow to involve the guru.

Inputs

action	This should equal "PurchaseCompleted"
AppID	This should be the stored AppID as returned from the APPLICATION_OPEN method
personName	The person's full name as a string
personDay	The day of the month the person was born as an integer between 1 and 31
personMonth	The month of the year the person was born as an integer between 1 and 12
personYear	The year the person was born as an integer above 1900
purchaseRef	The receipt number as provided by the market/app-store on transaction completion
emailAdd	The email address captured where the report is intended to be sent to
guruID	The ID number of the guru that should receive the request
osPurchase	If on Android this should equal "ANDROID", else if on iOS this should equal "IOS". Other values will throw an error. This is important as some markets require callback functionality to confirm online purchases.

Outputs

status	If "OK" then the purchase completed request was successfully processed and the workflow has been kicked off. AppID will not stumble the purchase request as payment has been submitted. Otherwise, a "FAIL" response may be provided if the transaction payload does not contain a proper receipt number, or the email address is not valid, or there was a database error
--------	--



Method: EMAIL REPORT

Purpose: This method allows the user to email the free report to an address of their choice

Inputs

action	This should equal "EmailReport"
AppID	This should be the stored AppID as returned from the APPLICATION_OPEN method
personName	The person's full name as a string
personDay	The day of the month the person was born as an integer between 1 and 31
personMonth	The month of the year the person was born as an integer between 1 and 12
personYear	The year the person was born as an integer above 1900
emailAdd	The email address as supplied by the user to dispatch the free report to

Outputs

status	If "OK" then the exit screen was successfully processed. Else there was a database error or the AppID does not exist.
--------	---

**Method: GET_DAILY_TEXT**

Purpose: This method is called whenever the application refreshes the home screen. It is used to provide the user with significant numerology news/facts.

Inputs

action	This should equal "GetDailyText"
AppID	This should be the stored AppID as returned from the APPLICATION_OPEN method

Outputs

status	If "OK" then the request was successfully processed. Else there was a database error or the AppID does not exist.
displayText	The text to display to the user, note that this will contain mild HTML formatting such as and
displayImage	This will contain a URL to an image that should be displayed in the adjacent image area – it will likely be a PNG with transparency

**Method: APPLICATION_CLOSE**

Purpose: This method is called whenever the application is exited on the user handheld. It is used to inform us as to where the customer leaves the application.

Inputs

action	This should equal "UserExit"
AppID	This should be the stored AppID as returned from the APPLICATION_OPEN method
ExitLocation	A predefined string to describe the screen where the user exited the application

Outputs

status	If "OK" then the exit screen was successfully processed. Else there was a database error or the AppID does not exist.
--------	---



Method: GURU WEBFRAME

Purpose: This facility allows a user to review a more detailed profile of a particular guru.

How to use: The guru_webframe is designed to be rendered in a webframe on a mobile device, as there may be additional links and graphics that the user will want to review about the guru. When this webframe needs to be called, the application should render the URL:

http://numerology.kamkash.com/guruProfile.php?id=<GURU_ID>

Where the <GURU_ID> parameter is found from the guru database that has been downloaded to the device.