AgeCheq THE COPPA COMPLIANCE ECOSYSTEM

API DOCUMENTATION

v3.05 11.12.2015

AgeCheq is a proud member of the Application Developers Alliance

www.agecheq.com

Document Version

Version	Date	Description
0.1	12/31/2013	Initial Version
0.2	01/05/2014	Initial Version Complete
0.3	01/18/2014	New Look documentation
0.4	02/16/2014	New app-based technology
0.5	03/13/2014	Active Registration Update
1.0	03/24/2014	Full Version
1.01	04/03/2014	Title Change
1.02	04/21/2014	Added associateData command
1.03	04/30/2014	Added agegateDevice command
1.04	07/16/2014	Updated http to https; updated screenshots
1.05	07/31/2014	JSON Object
1.06	08/08/2014	Developer Key
2.00	08/21/2014	Updated to API v 2.0
2.10	08/30/2014	Added trials parameter in check command
2.11	09/18/2014	Added new data.checktype
2.12	09/25/2014	Rephrased for clarity
2.13	01/07/2015	Updated to reflect change in Developer Dashboard
3.00	03/18/2015	Updated to API v 3.0
3.01	03/25/2015	Added associate command
3.02	03/31/2015	Added new VPC method
3.03	04/01/2015	Invalid AgeCheq PIN return message
3.04	10/28/2015	Added new Register API command
3.05	11/10/2015	Edits / Typos

*Other names and brands may be claimed as the property of others.

1.0 Purpose

The purpose of this documentation is to show developers how to use the AgeCheq API. It outlines what the API does and how a developer might integrate it into their online service. AgeCheq maintains a few Software Development Kits (SDKs) for popular platforms to make integrating the API that much easier. Find this document and all SDKs on the AgeCheq Documentation Site at: http://documentation.agecheq.com.

2.0 AgeCheq API

The AgeCheq API consists of a series of RESTful GET Web API requests. These requests are used to determine the status of a child's relationship with a particular online service. It requires an encoded authentication header, and returns a block of JSON data.

2.1 Using Basic Authentication

The AgeCheq API requires an Authorization header be supplied. Generally, AgeCheq SDKs let you specify your Developer Key and will handle proper encoding of the HTTP Basic Authentication header for you. If you prefer to use the AgeCheq API directly, you'll need to set the Authorization header yourself. The header value should be constructed by adding a colon (":") and a space (" ") to your Developer Key, then base64 encoding that string.

```
xhr.setRequestHeader("Authorization", "Basic " + window.btoa(devkey + ": "));
```

Note that the ":" and trailing space are required as it is actually the separator and password. The request header would then look something like this:

```
Authorization "Basic ZWRjMjZhMDctNThhMS00MTgxLTkxYTctYTM3NWYxYTI0YTRjOmE="
```

See this page for more details on Basic Authentication: http://en.wikipedia.org/wiki/Basic_access_authentication.

The endpoint for all AgeCheq API calls is: https://api.agecheq.com

2.2 Registering the Session

Before using the AgeCheq API, the system must first be primed using the *register* command. The register command is used to keep track of the Monthly Active Users (MAU) of a game or app, and records analytic data for use later. According to AgeCheq's Terms of Service, the register API command

must be called before any age gate that might be created for a general audience game or app that targets children under 13 as well as a more mature audience.

The URL for the call should be structured like this:

```
https://api.agecheq.com/applications/[Application ID]/register/[Unique Identifier]
```

- **Application ID**: This is the unique identifier for the particular application when it was set up on the AgeCheq site. It can be found in the developer dashboard under the App Info listed as App ID.
- **Unique Identifier**: This optional parameter is a unique identifier used to track an app start. The Unique Identifier should be passed if a previous *register* command was executed for a particular user. If no Unique Identifier is passed, a new one unique to the system will be generated and returned from this command.

When using the API directly, this call should include a header to pass authentication information as described in section 2.1. The developer key is a unique GUID available from the Developer Dashboard. Find it in the *Account* section across the top of the page at http://developers.agecheq.com. The header key should be the header value should be constructed by adding a ': ' to your Developer Key, then base64 encoding that string. See this page for more details on Basic Authentication: http://en.wikipedia.org/wiki/Basic_access_authentication. If you prefer not to have to deal with authentication, consider using one of the AgeCheq SDKs. They will handle proper encoding of the HTTP Basic Authentication header for you.

This API call will return a JSON object with the following properties:

- **rtn**: This property will return either "ok" or "fail" depending on whether the request was well formed and whether there was an error on the AgeCheq server or not.
- **rtnmsg**: This property will return a message describing why a check failed. If the check was successful, this property will return an empty string except for the *register* command.
- **data**: This property is a JSON object with some properties of its own:
  - **apiversion**: The version of the AgeCheq API used to fulfill requests.
  - **uid**: The unique identifier passed to the API, or a brand new unique identifier if no identifier was passed.

Example Call:

```
https://api.agecheq.com/applications/5bba264c-2adc-4cce-a657-d53d0d1d32f4/register/64c02071-83b2-4410-8448-25f151b7dbad
```

Example Results:

On a request that the server can't understand or rejects for some reason, the results may look like this.

```
{ rtn: "fail", rtnmsg: "invalid command" }
```

Once the system registers successfully, the API will acknowledge that it has received the data successfully and return a new unique identifier, or reflect the passed unique identifier.

```
{ rtn: "ok", rtnmsg: "", data: {
   apiversion: 3, uid: "64c02071-83b2-4410-8448-25f151b7dbad"
   }
}
```

2.3 Checking a Child's Status

The AgeCheq *check* API command determines the child's relationship to a particular online service at any certain time. It also provides age data about the child as reported by an adult from the AgeCheq Parent Dashboard. Please note that a child identified by a parent as being 13 years of age or older is not subject to the COPPA rule allowing developers to collect Personally Identifiable Information (PII) regardless of whether the app is authorized to collect PII from a parent or not. The URL for the call should be structured like this:

```
https://api.agecheq.com/applications/[Application ID]/acpin/[AgeCheq
PIN]/check
```

- **Application ID**: This is the unique identifier for the particular application when it was set up on the AgeCheq site. It can be found in the developer dashboard under the App Info listed as App ID.
- **AgeCheq PIN**: This parameter is the unique id assigned to a child by an adult from the AgeCheq Parent Dashboard (http://parent.agecheq.com). This information is unique for each and every child that uses the service. This AgeCheq PIN should be conserved by the developer for the purposes of reporting captured PII back to parents, or revoking parental approval for an online service.

When using the API directly, this call should include a header to pass authentication information as described in section 2.1. The developer key is a unique GUID available from the Developer Dashboard. Find it in the *Account* section across the top of the page at http://developers.agecheq.com. The header key should be the header value should be constructed by adding a ': ' to your Developer Key, then base64 encoding that string. See this page for more details on Basic Authentication:

http://en.wikipedia.org/wiki/Basic_access_authentication. If you prefer not to have to deal with authentication, consider using one of the AgeCheq SDKs. They will handle proper encoding of the HTTP Basic Authentication header for you.

This API call will return a JSON object with the following properties:

- **rtn**: This property will return either "ok" or "fail" depending on whether the request was well formed and whether there was an error on the AgeCheq server or not.
- **rtnmsg**: This property will return a message describing why a check failed. If the check was successful, this property will return an empty string except for the *register* command.
- **data**: This property is a JSON object with some properties of its own:
    - **apiversion**: The version of the AgeCheq API used to fulfill requests
    - **checktype**: How the check response was calculated based on the settings of the "Runtime Tuning" throttle found in the Developer Dashboard under "runtime"
        - **0** = Normal Check Call
        - **1** = Normal Check as a result of a throttled call
        - **2** = All data forced to 'allow' as a result of a throttled call
        - **3** = All data forced to 'allow' as a result of a trial call
    - **appid**: The unique id for the application within the AgeCheq system
    - **acpin**: The unique AgeCheq PIN sent with the call
    - **agecheq**: an object containing all AgeCheq-related data
        - **appauthorized**: This property will be set to true if the application is currently authorized by a parent, and false if the application is not authorized.
        - **appblocked**: This property indicates whether the application is currently marked as being blocked by the parent or not. It will return true if the application is not to be loaded because it is currently blocked. It will return false if it is not blocked by a parent. The
        - **parentverified**: The current verification status of the parent account tied to the device.
            - **0**: Not verified
            - **1**: Fully verified via payment or form
            - **2**: Partially verified via EmailPlus, SMS, or VOIP
            - **3**: Verified through an online payment implementation
        - **under13:** This field will be *true* if a child's birthdate assigned by an adult inside the parent dashboard makes them currently under the age of thirteen. Otherwise, this field will be *false*.
        - **under18:** This field will be *true* if a child's birthdate assigned by an adult inside the parent dashboard makes them currently under the age of eighteen. Otherwise, this field will be *false*.

- **underdevage:** This field will be *true* if a child's birthdate assigned by an adult inside the parent dashboard makes them currently under the age set by the developer in the AgeCheq Developer Dashboard. If this value is not set, or if the child's birthdate makes them older than the set date, this field will be *false*.
- **trials:** This is the number of app starts a user gets before they will be asked to verify through creating an account. This is optional and can be set in the developer dashboard under "runtime".

Example Call:

```
https://api.agecheq.com/applications/5bba264c-2adc-4cce-a657-
d53d0d1d32f4/acpin/olive126/check
```

Example Results:

On a request that the server can't understand or rejects for some reason, the results may look like this.

```
{ rtn: "fail", rtnmsg: "invalid command" }
```

If the request references a child id that doesn't exist, or has been removed from the system the results may look like this.

```
{ rtn: "fail", rtnmsg: "invalid child AgeCheq PIN" }
```

This is a sample response for an application that has not been authorized yet.

```
{ rtn: "ok", rtnmsg: "", data: {
    apiversion:3,
    checktype:0,
    appid: "5bba264c-2adc-4cce-a657-d53d0d1d32f4",
    acpin: "bobcat",
    appauthorized: false,
    appblocked: false,
    parentverified: 0,
    under13: false,
    under18: false,
    underdevage: false,
    trials: 0
   }
}
```

This is a sample response for an application that has been completely authorized to run and age gate information has been recorded.

```
{ rtn: "ok", rtnmsg: "", data: {
    apiversion:3,
    checktype:0,
    appid: "5bba264c-2adc-4cce-a657-d53d0d1d32f4",
    acpin: "bobcat",
    appauthorized: true,
    appblocked: false,
    parentverified: 1,
    under13: true,
    under18: true,
    underdevage: true,
    trials: 0
    }
}
```

2.4 Associating Data with a User

The COPPA law requires that developers remove any Personally Identifiable Information (PII) about a child under 13 if a parent revokes their permission for an app to collect it. Additionally, COPPA requires that developers share any PII taken from a child at the parent's request.

A parental permission revocation is a notice that is sent to a developer when a parent revokes their permission for a previously authorized app to collect PII. A PII request is a notice sent to a developer when a parent requests all the PII a developer has on their child. These revocation notices may be transmitted back to you by email, by a Web call, or both. Both of these notices are required by COPPA and both may be automated through the AgeCheq ecosystem.

The *associate* command gives you the ability to specify a single string of data that AgeCheq will return back to you in either a parental permission revocation or a PII request. That way, you can determine what records the parent is targeting and take appropriate actions to fulfill the parent's request.

The URL for the call should be structured like this:

```
https://api.agecheq.com/applications/[Application ID]/acpin/[AgeCheq
```

```
PIN]/associate/[Associated Data]
```

- **Application ID**: This is the unique identifier for the particular application when it was set up on the AgeCheq site. It can be found in the developer dashboard under the App Info listed as App ID.
- **AgeCheq PIN**: This parameter is the unique id assigned to a user by an adult from the AgeCheq Parent Dashboard (http://parent.agecheq.com). This information should be unique for each and every child that uses the service. This AgeCheq PIN should be conserved by the developer for the purposes of reporting captured PII back to parents, or revoking parental approval for an online service.
- **Associated Data**: The string to associate with the application and AgeCheq PIN. It will be returned as part of a parental PII request or a parental permission revocation.

When using the API directly, this call should include a header to pass authentication information as described in section 2.1. The developer key is a unique GUID available from the Developer Dashboard. Find it in the *Account* section across the top of the page at http://developers.agecheq.com. The header key should be the header value should be constructed by adding a ': ' to your Developer Key, then base64 encoding that string. See this page for more details on Basic Authentication: http://en.wikipedia.org/wiki/Basic_access_authentication. If you prefer not to have to deal with authentication, consider using one of the AgeCheq SDKs. They will handle proper encoding of the HTTP Basic Authentication header for you.

This API call will return a JSON object with the following properties:

- **rtn**: This property will return either "ok" or "fail" depending on whether the request was well formed and whether there was an error on the AgeCheq server or not.
- **rtnmsg**: This property will return a message describing why a check failed. If the check was successful, this property will return an empty string except for the *register* command.
- **data**: This property is a JSON object with some properties of its own:
    - **apiversion**: The version of the AgeCheq API used to fulfill requests

Example Call:

```
https://api.agecheq.com/applications/5bba264c-2adc-4cce-a657-
d53d0d1d32f4/acpin/olive126/associate/0014237872
```

Example Results:

On a request that the server can't understand or rejects for some reason, the results may look like this.

```
{ rtn: "fail", rtnmsg: "invalid command" }
```

If the request references a child id that doesn't exist, or has been removed from the system the results may look like this.

```
{ rtn: "fail", rtnmsg: "invalid child AgeCheq PIN" }
```

Once the data is associated successfully, the system will acknowledge that it has received the data successfully with results like this.

```
{ rtn: "ok", rtnmsg: "", data: {
   apiversion: 3
   }
}
```