LOGO ME BEAUTIFUL

# Developer Guide for KillSwitch API

# KillSwitch International, Inc.
**Document Version 01**

**Draft Notice**

# Notices

# Contents

# Preface

## Purpose of this Document

This document is a guide for developers to successfully implement the KillSwitch API within their systems.

## Audience

The *primary audience* of this document is developers and other people implementing KillSwitch features and functionalities within their product.

The *secondary audience* of this document is developers, technical leads, architects, and project managers working with the primary audience

## Revision History

Table 1.      **Revision History of This Book**

| Version | Author | Description |
|---------|--------|-------------|
| 01 | Pragati Chaplot Jain | Initial draft |
| 02 | Pragati Chaplot Jain | Revised draft with feature description and relevant use cases |

## Customer Documentation

The most up-to-date versions of all KillSwitch technical publications are available on the KillSwitch technical publications site, available directly from the Client Self Services Express site: http://KillSwitchservice.lstatic.com/helpdesk

All documents are available in Adobe™ Acrobat™ (PDF) format, and require Acrobat Reader version 8.0 or later, available from the Adobe site at: http://www.adobe.com

## Contact Information

The following table lists the ways in which to contact KillSwitch International about support and service, technical problems, suggestions, or questions.

Table 2.      **How to Contact KillSwitch**

| For help with... | Contact... | At... |
|------------------|-----------|-------|
| Technical problems or questions such as System failures or hardware RMAs | Technical Support Services | *Email Address (mailto:KillSwitch.support@lstatic.com)*<br>*Web Site http://www.lstatic.com/en-US/Service/Default.aspx* |
| Submitting incident requests, or checking the status of existing requests or RMAs | Client Self Services Express. Requires your client ID and KillSwitch-supplied password. | *Web Site http://KillSwitchservice.lstatic.com/helpdesk* |

# Chapter 1: Introduction

## KillSwitch Overview

KillSwitch shares and un shares photos—content you don't want forever on the Internet. Users can share photos on the Internet while controlling the viewer profile and for how long the image remains available on the web.

A user selects an image, uploads it to the KillSwitch server and identifies for how long the image must be active on the Internet. As soon as an image is uploaded into the KillSwitch cloud servers, a KillSwitch sharable link is generated for sharing the encrypted images. Since each image is encrypted, viewers must supply a password for viewing them. When the timer expires or the user "recalls" the image by manually flicking the KillSwitch, the password and image are removed from the KillSwitch servers. While the KillSwitch links can be reshared, the KillSwitch images cannot be downloaded.

KillSwitch application can be accessed as a web-based application on desktops and laptops or as an iOS mobile application on your mobile devices. While both modes facilitate easy sharing and managing of images across different social media platforms, KillSwitch iOS application allows users to delete a shared image and also tracks and displays more information about the image—"Like" notifications, image views, number of thumbs received, and number of smiles received— and notifies the image owners when someone likes their photo.

KillSwitch images are JPEG photos and are compatible with any established JPEG infrastructure. The following diagram describes the Privacy-Preserving Photo Sharing (P3) encryption technology implemented in the KillSwitch application for generating secured images:
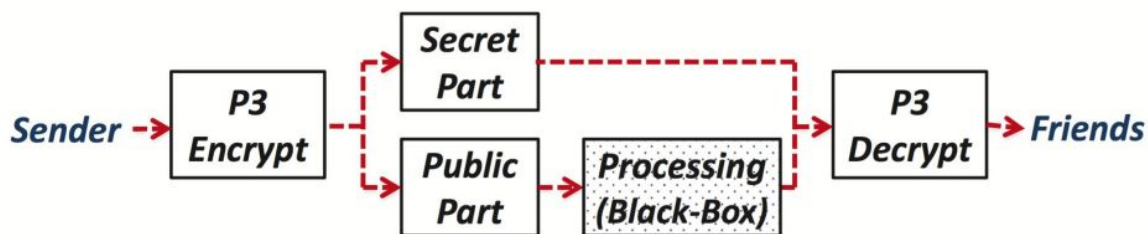


*Figure 1: P3 encryption technology for generating secured images*

While consumers can access the KillSwitch application as a web-based application or an iOS application, businesses can embed the KillSwitch features and functionality into their product by implementing the KillSwitch APIs.

# KillSwitch Features

## Upload an Image

Upload an image into the KillSwitch application and generate links for sharing images across different social media channels.

## Control an Image

Control a viewer access to a KillSwitch image by defining different image parameters such as viewer profile information, viewer location or image views.

## Recall an Image

Recall a KillSwitch image any time by flipping the KillSwitch or by setting different timer and image parameters such as the active link time period, the viewer profile information or the allowed number of views.

# Chapter 2: Use Cases

This section lists all relevant use cases for KillSwitch application. Corresponding API calls for each use case can be found in "Chapter 4: API References".

## Upload an Image

A user wishes to share pics of his new car with his friends and family on social media. However, he wants to avoid any unwanted attention from strangers on social media.

If this example has access to the KillSwitch application, the user can create a KillSwitch link and set an expiration day for 10 days. He can then distribute the KillSwitch link with his friends and family on social media. Irrespective of the social media platform the image is being shared on, the image will be automatically recalled after 10 days.

**Function Tree**
1. User uploads an image into the KillSwitch application.
2. User applies a timer to the image to create a protected shareable link.
3. Corresponding KillSwitch (protected sharable link) is returned to the user.

**API Usage**: Please see "Upload an Image" in the next chapter, "API References".

## Control an Image

A group of teens recently took part in a robotics competition and won the best design award for their robot. Since the team plans to use the same robot design for upcoming competitions, they are looking for a discreet way to share the images with their school team (peer groups, mentors and other educators) across different social media platforms.

If this example has access to the KillSwitch application, users can create KillSwitch links and passwords for image sharing and apply restrictions for image visibility by viewer profile information. As long as the link is active, the restricted group can access the images from any social media platform.

**Function Tree**
1. User uploads an image into the KillSwitch application.
2. User applies restrictions for image visibility by viewer profile information.
3. User applies a timer to the image to create a protected shareable link.
4. Corresponding KillSwitch (protected sharable link) is returned to the user.
5. User shares the KillSwitch images and the password with the restricted group of individuals.

**API Usage**: Please see "Control and Recall an Image" in the next chapter, "API References".

# Recall an Image

A user is scouting for a new insurance for his boat. He must share photos of the boat and different boat registration documents with different insurance agents to get their quote. Though these images can be quickly shared over a WhatsApp message, text or email, the approach also opens up to privacy and security risks.

If this example has access to the KillSwitch application, the boat owner can protect his privacy by sharing KillSwitch links with different insurance agents. Once he has bought the insurance, he can flip the KillSwitch to remove any trace of the photos online — mitigating his privacy and security concerns.

**<u>Function Tree</u>**
1. User selects the image in the KillSwitch application.
2. User flips KillSwitch to remove the image from the web.

**API Usage**: Please see "Control and Recall an Image" in the next chapter, "API References".

# Chapter 3: API References

## Before you Start

Sign up for a developer account by sending us an email at integrations@killswitchers.com. Once you have a working developer account, you can then start making request against the KillSwitch API at: https://sandbox.killswitch.pics

## Introduction to KillSwitch API

KillSwitch API is organized around REST. It uses standard HTTP verbs like GET and POST, standard HTTP error responses to describe errors, returns JSON responses and all communications are in SSL.

## API Responses

KillSwitch API uses conventional HTTP response codes to indicate the success or failure of an API request. Error responses include details about the problem occurred.

Table 3. **API Response Code and Message Description**

| Response Code | Response Message Description |
|---|---|
| 200 | Success |
| 204 | No content |
| 400 | Bad request |
| 401 | Unauthorized |
| 404 | Not found |
| 500 | Internal server error |

**Errors**

```
{
    "code":40401,
    "message":"No image found",
    "key":"IMAGE_NO_FOUND"
}
```

## API Authorization

All queries to the KillSwitch API must include an API key header followed by the API key to authenticate requests.

# Endpoints

## Upload an Image

Upload an image and returns the corresponding KillSwitch.

POST /v1/images/

Table 4.        Method - *CreateNewImage* Request Parameter Description

| Parameter Name | Data Type | Parameter Description |
|---|---|---|
| image | String | Image file in base 64 |
| duration | Number | Shared duration in seconds (*max* is 1 year=31557600 seconds) |
| embeddedPass | Boolean | Embeds password in the URL to decrypt the image |

*Note: Parameter* embeddedPass *is optional and the default value is 1. If set to 1, the password is embedded in the URL. The viewer does not need a password to decrypt the image. If set to 0, the returned link will ask the viewer for the password to decrypt the image. The password will not be stored in the server and URL.*

**Request**

```
{
    "method":"CreateNewImage",
    "id":50001,
    "jsonrpc":"2.0",
    "params": {
      "image":"myImageName",
      "duration":10800,
      "embeddedPass":0
      }
}
```

Table 5.        Method - *CreateNewImage* Response Parameter Description

| Parameter Name | Data Type | Parameter Description |
|---|---|---|
| link | String | Displays KillSwitch link for image sharing |
| password | String | Shows password for decrypting the image (*Optional*) |
| imageid | String | Identifies the image id for the image |

*Note: If* embeddedPass *in the POST request is set to 0, the returned link will ask the viewer for the password to decrypt the image. The password will not be stored in the server and URL.*

**Response**

```
{
   "id":50001,
   "jsonrpc":"2.0",
   "result": {

      "link":"http://147.15468.24.1254654:8080/vldms/tuner/ocap_locator_f=0x1701e480.
      0xce.m=0x10.m3u8",
      "password": "blueSJSU"
      "imageId":34
      }
}
```

## Control and Recall an Image

Displays different details about an uploaded image. An image will be deleted after the timer expires or by manually flicking the KillSwitch.

GET /v1/images/{imageId

Table 6.        **Method - *GetImage* Response Parameter Description**

| Parameter Name | Data Type | Parameter Description |
|---|---|---|
| views | Number | Displays number of views of the image |
| likes | Number | Displays number of "Likes" received by the image |
| thumbs | Number | Displays number of thumbs received by the image |
| smiles | Number | Displays number of smiles received by the image |
| remaining time | Number | Displays remainder time in seconds till image expires |
| link | String | Shares KillSwitch link for resharing |

*Note: If* `embeddedPass` *in the POST request is set to 0, the returned link will ask the viewer for the password to decrypt the image. The password will not be stored in the server or in the* `link`, *so user has to remember it.*

**Response**

```
{
   "method": "GetImage",
   "id":50001,
   "jsonrpc":"2.0",
   "result": {
     "views":45,
     "likes":34,
     "thumbs":12,
     "smiles":7,
     "remaining time":567s,
     "link":"http://147.15468.24.54:8080/vldms/ocap_locator_f=0x17.0xce.m=0x10.m3u8"
      }
}
```

# Glossary

| | |
|---|---|
| HIPAA | Health Insurance Portability and Accountability Act of 1996 |
| HTTP | HyperText Transfer Protocol |
| JPEG | Joint Photographic Experts Group |
| JSON | JavaScript Object Notation |
| P3 | Privacy-Preserving Photo Sharing |
| REST | Representational State Transfer |