

# Partner Integration Plan

Using the ProofHQ API

## Contents

<b>Executive summary</b>	<b>3</b>
<b>Partner Integration Plan</b>	<b>4</b>
<i>Overview of the steps</i>	4
<b>Step 1: Selecting the right account model</b>	<b>5</b>
<i>Examples</i>	6
1. TrackWork Inc - Hub & Satellite accounts	6
2. Acme Design Agency Ltd - Single account	7
<b>Step 2: License management</b>	<b>8</b>
<i>Single account</i>	8
<i>Multiple accounts (Hub &amp; Satellites)</i>	8
1. Creating your Hub account	8
2. Creating client satellite accounts	8
<b>Step 3: User management</b>	<b>10</b>
<i>Named users vs. Generic users</i>	10
<i>Managing passwords</i>	10
<i>Adding users</i>	10
<i>Updating users</i>	11
<i>Removing users</i>	11
<b>Step 4: Creating proofs</b>	<b>12</b>
<b>Step 5: Capturing proof events</b>	<b>13</b>
<i>Account-level proof callbacks</i>	13
<i>Proof-level callbacks</i>	13
<i>Example of using Callbacks: Custom “holding” page</i>	13
<b>Step 6: Embedding the Proof Viewer</b>	<b>15</b>
<i>Embedded Miniproof Viewer</i>	16
1. <code>getProofEmbedCodes()</code>	17
2. <code>getProofReviewerEmbedCode()</code>	17
<i>Standard Proof Viewer</i>	17
<b>Step 7: Managing versions</b>	<b>19</b>

## Executive summary

ProofHQ's comprehensive Web services API significantly reduces the effort required to integrate your application with ProofHQ and provide online proofing solutions to your users or customers. Integration with ProofHQ is fast, simple, secure and low risk, enabling you to achieve a fast return on your investment.

Our 7 step Partner Integration Plan [[link to detailed plan](#)] will give you everything you need to complete and launch ProofHQ's online proofing solution within your own. The Partner Integration Plan may also be used for adding online proofing functionality to 3rd party applications like Adobe CS [[link](#)].

The ProofHQ API supports all major development environments and tools including .NET, Java and PHP. Versioning of the API will also ensure that your integration never breaks, while allowing you to start using enhancements when you are ready to do so.

The ProofHQ API is hosted on the same scalable and secure architecture as ProofHQ's high availability web application. We have additionally developed a "Hub & Satellite" account model that allows partners and customers to manage multiple ProofHQ accounts. Please see our [Security overview](#) for more information.

## Partner Integration Plan

This plan is for anyone integrating ProofHQ's online proofing solution with their own applications. This plan takes you through 7 easy-to-complete steps and delivers an integrated solution allowing you to realize a return on your effort as quickly as possible.

Please make sure to read the [overview](#) page of the ProofHQ API documentation before continuing with this guide

### Overview of the steps

1. Selecting the right account models - during this step we will explain the two different account models and provide advice on which one to use
2. License management - this step will cover the creation of satellite accounts and managing their plans (licenses)
3. User management - the creation, updating, activation and deactivation of users
4. Creating proofs - uploading of files, creating of proofs, management of the reviewers
5. Capturing events - using callbacks to capture when the proof is ready and other reviewing events
6. Embedding the proof viewer - embed the Miniproof or full proof viewer inside your application
7. Managing versions - uploading new versions of a proof.

## Step 1: Selecting the right account model

Before starting with the integration it is important to understand your options for setting up and managing ProofHQ accounts and users and how these relate to your application.

At the very core of ProofHQ is an account (representing a real-world organization) and within an account are users, proofs, files and folders. When considering the account model you need to consider the following two options:

- **Single ProofHQ account** for all your proofs and clients where your clients could simply be guest reviewers
- **Multiple ProofHQ accounts** with a central hub account allowing you to create and manage satellite accounts on behalf of your clients (Hub & Satellite model) - for more information on satellite accounts please see our [help](#) pages.

When considering the most appropriate model to use, the following factors can be considered:

- A user may only belong to one account#, a very important factor to consider if you have users that need to create proofs in multiple accounts
- Satellite accounts must be on a paid plan
- Proofs are deducted from the account that 'owns' the proof (the proof owner is a user who belongs to a specific account)
- Satellite accounts for each client provide an additional layer of security. Users in one account will not be able to see any information from another account unless that information is explicitly shared with them.
- The Hub & Satellite account model allows a partner to manage separate licenses for each client more easily. It is easy to set up dedicated accounts for each client and easy to manage the billing for each separate account.
- Branding is unique to each account and the Multiple ProofHQ accounts model will allow you to brand each one separately.

## Examples

### 1. TrackWork Inc. - Hub & Satellite accounts

#### *Background*

TrackWork Inc. develops and hosts a multi-tenanted project tracking and management application. The data model is similar to ProofHQ's where an organization may contain multiple users and users only belong to a single organization.

Permission to view files and content is managed through Projects. Users are given access to participate on projects by the owner of the project. Projects will contain multiple files and all users participating in the project have access to all the project's files.

#### *Account model*

TrackWork uses the Hub & Satellite account model where each satellite account represents a client account in TrackWork. TrackWork chose this model for the following reasons:

- Ability to charge each client for a proofing license add-on
- Ability to handle all billing i.e. TrackWork charges clients and pays ProofHQ
- TrackWork's clients can choose the appropriate proofing plan while TrackWork can manage the upgrades and downgrades using the ProofHQ API
- TrackWork adds users to ProofHQ with Administrator rights allowing them to see all proofs in the account while knowing they cannot access another TrackWork client's proofs
- TrackWork was able to brand the Proof Viewer loading logo specific to each client.

## 2. Acme Design Agency Ltd - Single account

### *Background*

The Acme Design Agency provides general design and advertising services for a wide range of clients. They produce and manage feedback on 1000s of designs and videos every month for multiple clients. Managing a large volume of files means Acme Design Agency had to develop their own project management software internally to manage the process.

The reviewing process is currently handled outside of Acme's software and they urgently need to provide an integrated proofing solution to reduce the existing overhead.

### *Account model*

Acme selected to use a single ProofHQ account for all their proofing requirements:

- Acme is always the proof creator and manages all proofs
- Acme's clients are only reviewers and therefore do not require a ProofHQ user license, in turn meaning they could only access the proofs shared with them
- Acme uses their branding for the Proof Viewer loading logo

## Step 2: License management

### Single account

Generally if you are using a single ProofHQ account the plan and payment method are decided by your organization and managed by logging in to your ProofHQ account and going to the Billing section.

If you would like to manage the plan using the API you can do so by upgrading and downgrading your plan using the [updateBillingPlan\(\)](#) method (leaving the accountID empty).

### Multiple accounts (Hub & Satellites)

Once you have decided to use the Hub & Satellite account model you must then decide how you would like to create and manage the satellite accounts.

#### 1. Creating your Hub account

Before you do this, ensure that you have your Hub account ready:

1. Sign up for a trial account - this account will be your **Hub** account i.e. your organization's account
2. Email our Partner account management team at [partners@proofhq.com](mailto:partners@proofhq.com) to begin the process of getting your relevant promo codes that must be used when creating satellite accounts.

While step 2 is in progress you can continue with the integration.

#### 2. Creating client satellite accounts

You can opt to create and manage your client satellite accounts manually through the web interface or automate this using the ProofHQ API.

Many partners start with the manual process of managing satellite accounts as this is not a critical step to automate via the API.

### *Web interface*

Our help section contains the information you require to [Create](#) and [Manage](#) your satellite accounts. This includes [upgrading](#) and downgrading of your client's satellite account.

You will then need to obtain the new satellite account's ID for future API calls - do this by calling the [getSatelliteAccounts\(\)](#) method.

### *API*

The API allows you to programmatically manage client satellite accounts e.g. you can provision new satellite accounts, upgrade and downgrade their proofing plans using the ProofHQ API. This essentially gives your users the capability of selecting the online proofing plan of their choice however you decide to package it.

Creating a satellite account using the ProofHQ API:

1. Log in to your Hub account using the [doLogin\(\)](#) method to obtain an API session ID - now you're logged in to ProofHQ as this user
2. Create the satellite account by calling the [createSatelliteAccount\(\)](#) method using the SessionID obtained in Step 1 and providing the client's account name, time zone and the promo code provided by ProofHQ (if you do not have a promo code please contact our Partners account management team at [partners@proofhq.com](mailto:partners@proofhq.com))
3. The [createSatelliteAccount\(\)](#) method will return an object which will contain the account ID for the new satellite account. Make sure to store this for future use.
4. The satellite account can be upgraded / downgraded to the relevant plan by calling the [updateBillingPlan\(\)](#) method using the SessionID obtained in Step 1 and providing the AccountID obtained from the [SOAPSatelliteAccountObject](#).

## Step 3: User management

Performing actions using the API is the same as a user logging in to the web application and performing those same actions. Furthermore, only people who need to create and manage proofs need to be registered users in ProofHQ. People who only review proofs do not have to be registered as ProofHQ users.

Users in ProofHQ are identified by their email address.

### Named users vs. Generic users

We highly recommend not using a generic *API* user. This generic user will be reflected on all proofs during the review process and often creates confusion. Best practice is to create a user for each real world proof creator and have them create and/or own the proofs in ProofHQ.

### Managing passwords

ProofHQ gives you the ability to restrict user accounts to logging in via the API only. All user accounts require a unique email address and password, but since your clients will not be logging in to ProofHQ you can create the user accounts with a password that you can store in your application and is not known to your clients i.e. there should be no need for you to create user accounts with the user's original password.

There are two exceptions:

- Secure proofing: where a user is required to log in to ProofHQ to review a proof
- Electronic signatures: where electronic signatures are enabled and decisions can only be made by a user entering their username and password.

If you would like more information on how to handle these cases please contact us.

### Adding users

You can add a user to an account by calling the [addUser\(\)](#) method and passing in the SessionID obtained and satellite Account ID along with the person's unique email (this email must not be used as the username in any other ProofHQ account) along with their credentials and account permissions profile.

We recommend suppressing the activation email, which will automatically activate the user account and also not send them any welcome emails.

The APIOnly parameter should also be set to true as this user will most likely never log in to the ProofHQ web interface.

Each activated user will take up 1 user license.

## Updating users

There are various methods that can be used to update users' information, email address etc. - please see our API documentation for a full list of methods.

## Removing users

Users can either be **deleted** from the account or simply **deactivated**, with both methods freeing up a user license.

### *Deleting users*

If a user has left the organization i.e. is deleted from your application, you can delete the user and transfer ownership of all proofs to another user using the [deleteUser\(\)](#) method.

### *Deactivating users*

Sometimes your application will only require a subset of users to create proofs. You can provide your client's administrators with the option of changing who these proof creation users are.

This means your clients only need a proofing plan that fits their needs. They can [deactivate\(\)](#) users and [reactivate\(\)](#) them at a later stage to achieve this flexibility.

## Step 4: Creating proofs

We suggest that proofs are created by the same person who would normally have e-mailed or distributed the proof. You may achieve this by logging in to the API as this user and then creating the proof. If you opted to use a generic API user to create the proof, be aware that this user will appear on the reviewer's list.

Creating a proof is a simple 4 step process:

1. Upload the file to ProofHQ's servers. On successful uploading of the file ProofHQ will return a hash code for the file
2. Create the proof using the API and passing in the relevant fields including the hash code for the file
3. Register the proof for callbacks (event notification)
4. Add one or more reviewers to the proof

This section will cover steps 1,2 and 4. Step 3 is covered in the next section.

Please see these links for recommended reading before continuing:

- [FAQs](#)
- [What's the difference between a proof and a version?](#)

The section "[How do I create my first proof?](#)" covers steps 1 and 2 above. Once your proof has been created you should register the "Callbacks" and then you can add one or more reviewers by using the following methods:

- [addProofReviewer\(\)](#) will add a single reviewer to the proof and allows you to suppress the email notification from ProofHQ
- [addProofReviewers\(\)](#) allows you to add multiple reviewers to the proof and will send an email notification from ProofHQ.

## Step 5: Capturing proof events

Callbacks allow you to register one or more URLs on proofs that ProofHQ will then use to post proof updates to. This alleviates the need for you to intermittently poll ProofHQ for updates on your proofs.

Callbacks can either be registered on an account or on a proof.

Please read this help page before continuing: <http://api.proofhq.com/home/callbacks>

### Account-level proof callbacks

Account-level proof callbacks automatically register the URLs to be called on every subsequent proof created in the account whether through the API or through the Web UI.

Please see the following methods for managing Account-level callbacks:

- [setAccountCallback\(\)](#)
- [getAccountCallbacks\(\)](#)

### Proof-level callbacks

Proof-level callbacks are registered once the proof is created and generally before adding reviewers. Proof-level callbacks allow registering unique URLs for every proof for example if you would like the proof ID appended to the URL.

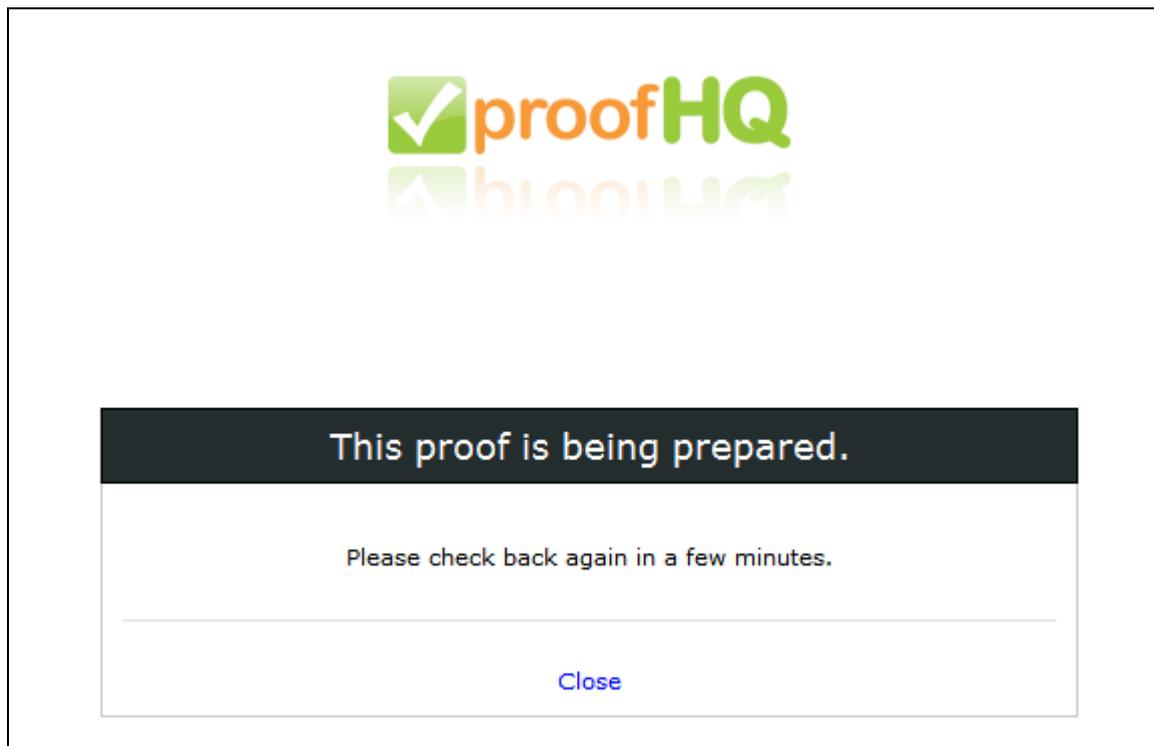
Please see the following methods for managing Proof-level callbacks:

- [setProofCallback\(\)](#)
- [getProofCallbacks\(\)](#)

### Example of using Callbacks: Custom “holding” page

Every proof that is uploaded to ProofHQ goes through an encoding process whereby it is converted for online web display or delivery. This has numerous benefits, the most noteworthy being that reviewers are not required to download the original file but only a series of images much smaller in size than the original file.

During this proof creation process ProofHQ will display a “holding” page letting the reviewer know the proof is not ready for review.



You may want to display your own “holding” page while the proof is being prepared. This can be achieved by registering for the “Proof processing” callback. Once ProofHQ has created the proof we will post a message to this URL to notify you that the proof is ready for review and in the meantime you can display your own “holding” page.

## Step 6: Embedding the Proof Viewer

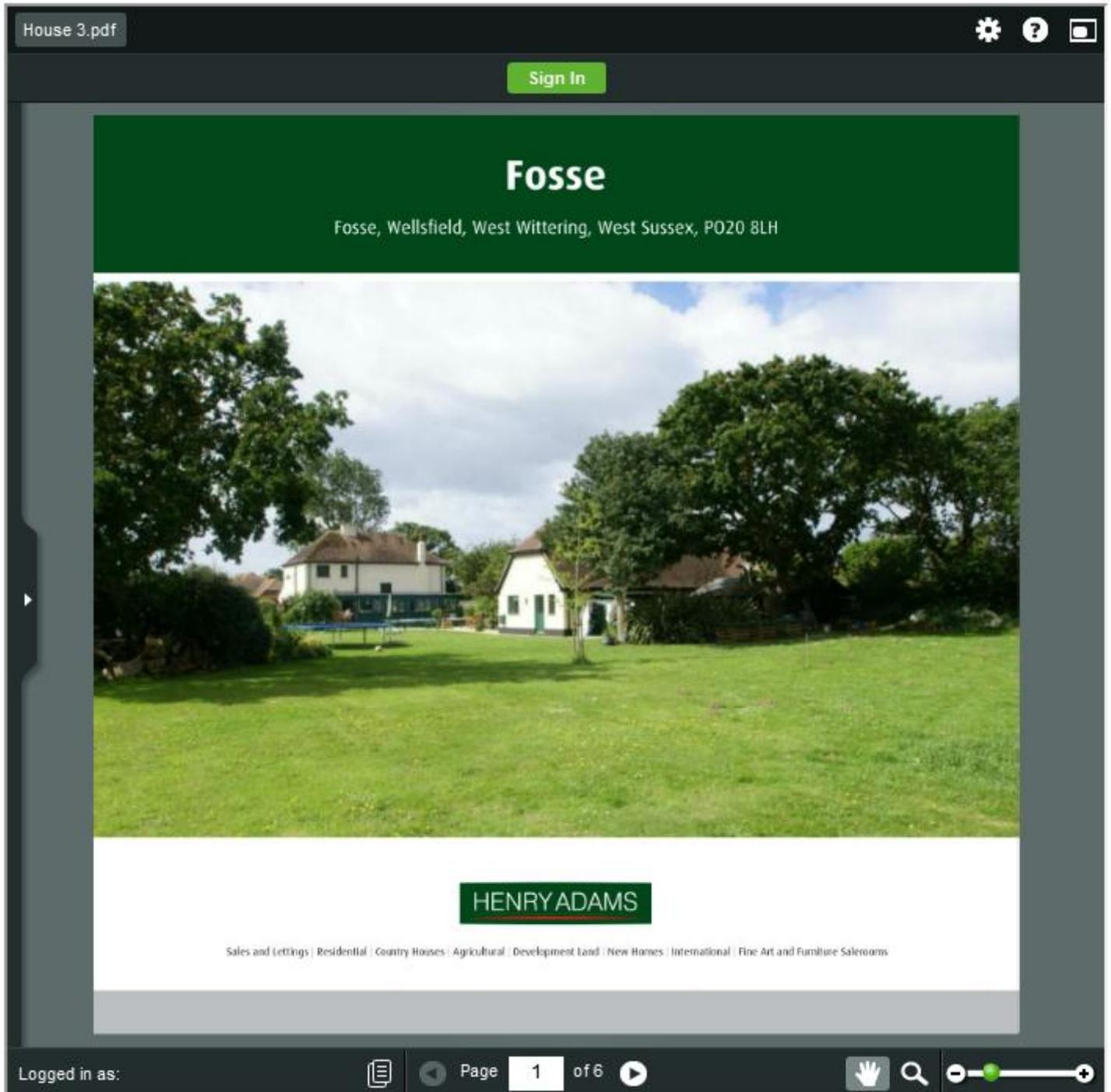
ProofHQ offers two options:

- Embedded Miniproof viewer: this proof viewer provides the basic functions for review and excludes the compare mode, which can be accessed by switching to the full screen standard proof viewer from within the Miniproof viewer
- Standard proof viewer: this is the viewer you would use when clicking on a standard proof link in ProofHQ

Each reviewer has a unique URL (code) which identifies the proof and the reviewer on the particular proof. When embedding either Proof Viewer it is important to use the personal URL for that user and proof combination: <http://api.proofhq.com/faqs#TOC-What-is-a-proof-personal-URL->

The following sections explain how to embed each Proof Viewer:

## Embedded Miniproof Viewer



House 3.pdf

Sign In

# Fosse

Fosse, Wellsfield, West Wittering, West Sussex, PO20 8LH



**HENRY ADAMS**

Sales and Lettings | Residential | Country Houses | Agricultural | Development Land | New Homes | International | Fine Art and Furniture Salerooms

Logged in as: [document icon] Page 1 of 6 [hand icon] [search icon] [zoom slider]

The Miniproof viewer contains all the reviewing functions and allows the reviewer to access the standard Proof Viewer by clicking the icon in the top right-hand corner. The Miniproof viewer does not support the Compare function.

The URL format to display the Miniproof viewer is as follows:

<https://www.proofhq.com/viewer/embedID/personalCode>

The API has methods that will supply both the iFrame source code you can use to embed the Miniproof viewer in your application and optionally the URL to the Miniproof viewer if you want to use a separate window or tab to display the Miniproof viewer:

### 1. **getProofEmbedCodes()**

Returns for all reviewers on the proof their iFrame HTML code to embed the Miniproof viewer in your application including the HTTP URL to the Miniproof viewer:

*iFrame HTML code*

```
<iframe src="https://www.proofhq.com/viewer/embedID/personalCode" width="700" height="700" scrolling="no"></iframe>
```

*HTTP URL*

The URL in the *src* part of the iFrame code:

<https://www.proofhq.com/viewer/embedID/personalCode>

### 2. **getProofReviewerEmbedCode()**

Returns the Miniproof viewer iFrame HTML code for a specific reviewer.

## **Standard Proof Viewer**

The standard Proof Viewer includes all the available functions depending on the reviewer's role. The URL format for the standard Proof Viewer is:

<https://www.proofhq.com/proof/personalCode>

You can choose to open the standard Proof Viewer in a separate window or tab or optionally display it using an iFrame or light-box.



All you require is a reviewer's personal proofing code which you can obtain by calling [getProofReviewers\(\)](#). On the SOAPRecipientObject the proof\_url parameter will provide you the full URL to the standard Proof Viewer for a specific reviewer.

## Step 7: Managing versions

Please see our API documentation for the difference between a proof and a version: <http://api.proofhq.com/faqs#TOC-What-s-the-difference-between-a-proof-and-a-version->

It is important to note that each version has a separate Proof ID and they are merely linked by their IDs into a group to represent a single proof in the Web UI.

The same 4 step process should be used to create a new version of a proof except that the method [createProofVersion\(\)](#) should be used that takes the prior version's proof ID. This ensures that reviewers are able to navigate and compare between versions of the same proof.