# Use Cases for the Authentication and Authorization of Projects Website

## Introduction

The Projects Website data needs to have authorization-of-access applied, by project. To support this, it will also be necessary to sometimes authenticate users and bind them to authorization credentials. This document will describe the use cases to be applied. These cases are for end-users, not administration.

It should be noted that there are two copies of the database-of-use, internal and external. The external copy is replaced by a reduced version of the internal one on a nightly basis.

If users are not logged in it is assumed they are a “guest” user.

There are two levels of access, “view” and “edit”. “View” access allows reading of information. “Edit” allows writing of information.

## Supporting Data

There needs to be data added to the Project table to support this. In particular, each project will get a “is-secure” field (is-public is already there), and two groups–table references, one field for view and one for edit. “Groups” does not yet exist, but will consist of an ID and a name. The actor\_groups table will now be utilized as a relationship table between actor and groups.

Since projects have a hierarchy to them (usually quite flat), care must be taken to safeguard projects whose parents are secure. When a project is marked as secure, its sub-projects must be marked as secure. When projects are added below parent projects, they must be secure if their parents are secure.

## Use Case Summary

These are the use cases, but not the variant scenarios that will be addressed.

* Viewing Project Status - looking at the project status URL/action output.
* Viewing Event Details – looking at the backlog of events and attributes.
* Downloading Event Templates – getting an unfilled form representing an event for a given project/study.
* Uploading Event – posting an event to the database.

## Viewing Project Status

In the earliest phases, this has had no controls on it except by firewall and database inclusion. This function allows viewing of only the latest event, sample or project attributes given the project. Those inside the firewall have access to an all-inclusive database, but those approaching from without (DMZ) have access to a database of only public projects. Non-public projects have a 0 value in their “is-public” field and are not pushed outside.

Moving forward, we need to control access to even public data. That is, there will be authorization required for access to some “is-public” data. Some of the data will not need authorization, and such will be marked “is-secure == false”. Any user (logged-in, not logged in, regardless of groups possessed) may view non-secured data.

When projects are marked “is-secure==true” (secure data), users will need to login, and have appropriate group-inclusions to access that data, or they will not be able to see it. If users never ask (through choice of URL) to see non-secured data, they will never need to login. As soon as a user attempts to use secured data, they will have to login. Once login has succeeded, and the set of project groups has been obtained, they will be compared to the one(s) requested. If any do not match, the data will not be returned, but silently omitted.

## Viewing Event Details

Event details should always have controls put on them. No details should be viewable unless the user is logged in. Some projects’ details should be viewable by all, however, implying a kind of default or ‘unsecured’ group to be established. Guest users may view things in this category, but nothing else. As always, non-public projects will not be viewable by users outside the firewall.

Users need only “view” access to see event details.

## Downloading Event Templates

This operation allows one to get templates (unfilled forms) usable to create events for later posting. It is controlled. Users must have ‘edit’ access in order to carry this out. This step will require a login, just to see the page. When users go to this URL they will see a login screen to be filled out. Once beyond that, they will be landed back on that page. The page will show a list of Projects and Studies (all—not just the ones of access), from which to select the event of interest. They can then drill down to the event, and if they have access to the project, will be provided with their downloadable file.

## Uploading (Posting) Events

Here, users are changing information on the site. This should be especially well controlled. There will be the usual two layers to traverse. First, the user must be logged into the system, simply to see this page. If they click this link directly, they will be prompted to login (unless they are already). Once in, they will see a dropdown of all event types from which to choose. Since this part is somewhat open (we are making users fill in a whole event worth of info, and hence do not need to make them specify it again on the web page itself), we therefore need to put our controls on the uploaded file.

The uploaded file will contain the project name. That must be checked against the user’s set of groups. This required that they are in the “edit” group for the project they are attempting to modify. Hence, a logged-in user will be pushing the “upload” button, and the backing Action object will check the file contents to make sure it is acceptable coming from the user, based on that user’s actor groups.