#Company\_Logo#

**Current Security Settings**

**For**

**App\_Name**

**As of**

**Date\_Create**

**At Time\_Create**

**COPYRIGHT**

App\_Name Security Settings (Date: Date\_Create at Time\_Create)

The information provided in this document, including URL and other Internet Web site references is "as-is" and for informational purposes only. The information contained herein is subject to change without notice and is not warranted to be error-free. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious and for only illustration purposes. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Turanto product. You may copy and use this document for your internal, reference purposes.

© 2016 eTelic Inc. All rights reserved.

#Table\_TOC#

# Introduction

This document represents the security settings in *App\_Name* as of *Date\_Create at Time\_Create*. Please refer to the security guide to understand how to change these settings.

To understand the document and settings better, the table below gives a quick summary of the different security concepts supported by *App\_Name*, and their impact on application behavior.

|  |  |
| --- | --- |
| Name | Description |
| Roles | Roles typically represent a group of real life user roles that have a set of permissions. A user has to be in a role to get application access and can only perform the actions defined by the permissions granted to that role. |
| Permissions | Permissions are typically the CRUD (Create Read Update Delete) rights to entities for a given role. In addition, other actions (referred to as Verbs and Self Security) are also controlled through these permissions. |
| Field Level Security | You may need to give someone view and edit rights to a record, but not to all fields in a record. Field Level Security for a role allows you to have finer grain control over who can view and edit which fields in an entity. |
| User based security | With user based security setting users see only those records that are associated with their user id. They will thus not be able to see any records connected to other users.  This allows you to enable an end customer to login and see his orders. User based security can be ignored for certain roles, allowing members of that role to view records for everyone. |
| Self-Service | If self-service is enabled on an entity for a given role, then users in that role will be able to exercise the permissions granted (e.g. edit), but only on those records that they are connected to (similar to user based security). In this scenario, they may be granted rights to edit records, but the self-service will restrict the capability so they can edit only their own record. This is different from User based Security, which essentially converts the application into a multi-tenant application – and only the records for logged in user are returned from database.  (Self-service is not applicable for all the entities but the entities that are associated directly or indirectly with ‘user’ entity. |
| Auto-Registration | In application the new users can register themselves and create a record of their own. The user though would not have privilege to see any of the application content unless provided with a role either through admin, default role for new users or dynamic role mapping.  In addition, if a record for the user does not exist (e.g. a Member entity record for a new user that just registered would not exist), then the user is allowed to create that record and connect his user id to that record. |
| Multi-Tenant Security | This is implemented at the code level and cannot be modified by the end administrators. Currently, this application (does NOT have / has) multi-tenant security (based on the entity TODO: <Entity Name>).  Thus a user can see only the records that are connected to the organization or entity that he is associated with. And all the roles or permissions that are then defined are defined within that context.  Essentially, this setup disallows multiple organizations to see each other’s data, even though they are hosted on the same system. |
| Login Attempt Control | The administrators of application can control the attempts to login into application by forcing password change, no. of failed login and locking the users from accessing the application. |

## Types of Securities displayed in this Document

The document covers security in App\_Name in these sections:

### Configurable Security Settings

1. **Role Based Security View:** The section takes one role at a time and displays the tables of entity level permissions and Field Level Security for it.
2. **Dynamic Role:** This section will cover the dynamic roles – where a user can be placed into a certain role dynamically based on the value of a certain property within an entity (e.g. Employee titled “Major” will automatically get the privileges of “Major Role” or a role that is mapped with the dynamic value of “Major”).
3. **Entity Based Security View:** The section displays the inverse view of the above – for each entity, it describes the different roles that can act on it.
4. **User based Security:** All records in an entity can be restricted to only those that are somehow related to the logged in user. This section displays all those entities that are restricted in this manner.
5. **Hierarchical Security:** This allows an entity protected by user based security to be acted upon by hierarchical supervisor.
6. **Login Security Compliance:** This allows administrators to control the login behavior of users by locking users after specific no. of failed attempts and forcing password change after particular no. of days (for example 60 days). The user once locked will be unlocked automatically on completion of lock duration or can be unlocked earlier by admin only.
7. **Custom Role Based Pages:** This section describes which roles have specific custom pages defined. This enables users of that role to see a totally different page than the default one.

### Unchangeable Settings

1. **Multi-Tenant Settings:** An application can support multiple organizations within it, while separating each organization data right at the base code and database level. Since this setting is NOT configurable, it is not covered in this document.
2. **Role Based Workflow Settings:** The actions that a user can take on a workflow are defined in the application model and are NOT configurable by an administrator. Since this setting is not configurable, it is not covered in this document.

### Not Covered

1. **Conditional Security View:** The application may choose to hide or show or make certain fields un-editable based on business rule and other settings. These conditional security settings are NOT covered in this document and can be found in the business rules document.
2. **Users in a Role:** While it is possible to display the different users that are members of different roles, this list is not included in this document.

# App\_Name Roles based security view

This section of document provides the roles and associated permissions in *App\_Name* application.

Each role description starts on a new page for ease of reading and printing permissions for review.

## Role Selection at Login

If a user is a member of multiple roles, then after login he is presented a dialog to login as a specific role of his choice. This will allow the user to decide which role he is logging in as.

Currently there are #User\_In\_Multiple# of users configured in multiple roles.

## The ADMIN Role

1. The ADMIN role is not explicitly documented; it can perform all actions in the system.
2. Its permissions cannot be changed, although the members that have admin privileges can be controlled.
3. Currently, there are #User\_In\_Admin# of users configured as Admin.

The next few sections provide more details about each role in the system.

#Table\_Role\_Permission\_FLS#

#Table\_UserBasedSecurity#

#Table\_DynamicMapping#

# Authentication Settings

#Login\_Setting#

#Table\_SCS#

|  |  |  |
| --- | --- | --- |
| Key | Value | Description |
| Apply Security Policy | #S1# | Login Security is applicable on application |
| Maximum Failed Access Attempts Before Lock | #S2# | #S2# failed login attempts will lock the account |
| Default Account Lockout Time Span | #S3# | Account will be locked for #S3# hours |
| Password Expiration Within | #S4# | Mandatory password change after every #S4# days. |

# Summary

The document covered all the security settings configured specifically for *App\_Name* by the Administrator of *App\_Name*. The document provided the security configuration as of *Date\_Create at Time\_Create*.