

APPIAN

Topics under discussion

- Appian Applications
- Users and Groups
- Security



Appian Applications

- Applications are a collection of objects that make up a business solution.
- Appian recommends creating a dedicated application for every business solution.
- For example, *Customer Relationship Management (CRM)*, *Employee Onboarding*, and *Sales Opportunities* would be three different applications.



Ways of Creating an Application

There are two ways for creating an application:-

1. Creating from Scratch
2. Creating an Application using Application Builder

Creating an Application from Scratch



Concepts:-

- Applications are built using design objects that together form the user interfaces, logic, processes, and data users interact with while doing work in Appian.
- Every application that you build in Appian should represent a business solution.
- For example, you would build separate applications for customer relationship management and human resources.
- All design objects in Appian are secured separately, including the application object itself.



Key Features to keep in mind while creating an application

- An application functionally groups the design objects associated with it.
- Design objects may belong to zero, one, or many applications.
- To make an application's tasks, records, reports, and actions available to users you must configure security for the design objects, such as processes.

Steps for creating an application from Scratch



1. Log in to [Appian Designer](#) (for example, *myappiansite.com/suite/design*).
2. Click **New Application**.
3. In the **Name** field, type **<Name of the Appian Application>**.
4. Optionally, in the **Description** field, add a short description.
5. Click **Create**.
The application contents view displays.



Steps for creating an application from Scratch

6. Add Groups to the Application

- One of the first things that you need to do for each new application is to create at least two groups: one for the users who can initiate the action and one for the application's administrators.
- Groups are important building blocks of an application because they allow you to organize users and assign permissions to the groups of users as you add objects.



Steps for creating an application from Scratch

6. Create a Process Model

1. Open the application contents view of the <Appian_Application_Created >application.
2. Click **New**, and then click **Process Model**.
3. In the **Create Process Model** dialog, complete the following fields:
 - For **Name**, type <Process_Model_Name>
 - Click **Create New Process Model Folder**, and then name the folder <Process Model Folder> (you do not need to specify a parent folder)
4. Click **Create & Edit**.
5. To publish the process model
 - Click **File**, and then click **Save & Publish (Ctrl+Alt+S)**.
 - Close the Process Modeler.



Steps for creating an application from Scratch

6. Adding Security to the Application and the Process Model

7. Add an Action to the Application

- To create an application action you must first create a process model that allows a user to interact with the process.
- Make sure to set appropriate security on your process model so that users will be able to start it.
- Appian recommends adding an **Administrator** group and at least one **Initiator**, **Viewer**, **Manager** or **Editor** group to the process model

8. Publish Your Application



Steps for Creating an Application using Application Builder

Concepts:-

- The Application Builder is a feature of Appian Designer that creates an application with pre-defined set of objects.
- There are two different types: basic and full.
- Both sets of templates use the same paradigm to create applications, just with a different level of functionality.

Steps for Creating an Application using Application Builder



- **Creating a Basic Application**

- A basic template from the application builder creates only the objects necessary to create, view, update, and delete the records in your app.
- This template is especially useful if you want to jump start a brand new application. Instead of creating the base objects from scratch, this template will give you a basic application in a few minutes.

- **Creating a Full Application**

- A full template has all of the record functionality of a basic application, and also includes functionality for: audit history, collaboration, document management, task assignment, and reporting.
- This template is useful if you had planned to implement this functionality in your application anyway; otherwise, a basic application will likely be simpler to enhance and build out.



The Application Builder Wizard

- The Application Builder is a four step wizard that will guide designers through the application creation process. Depending on whether a basic or full application was selected, the defaults within the Application Builder will change.
- They are:-
 - Name
 - Interface
 - Collaborators
 - Finished



Name

- On the first page, users name and describe their application.
- Each app stores data in the form of Records in Tempo. We collect the singular and plural name not just because the record type requires them.
- We use the first letter of each word in the application name to generate a prefix for the design objects in the application.
- For example, a rule in the Support Ticket Management application might be called "STM_GetSupportTicketById". This prefix is guaranteed to be unique among all Quick Apps and generated apps on the system.



Interface

- On this page, users specify the fields of information they want to capture for each record. This serves as both data and interface design for the app, and provides the core of its unique design.
- When using the full template, three fields are pre-populated: **Title**, **Status**, and **Priority**. Of these, only **Title** is required (however, the name can be changed). When using the basic template, only the **Title** field is pre-populated.
- A Preview Form option is available on this page. It shows how the currently configured values will manifest in the Action and Records of the completed application. Specifically, it offers a preview of the start form of the Action, and the Summary view of the record type.



Collaborators

- On this page, users specify the people who should use the application.
- Collaborators have access to all application functionality.
- The people selected at this step are added as members of a group, which is created to secure application activity to.
- The application builder, will default for the full template to a sites-based application.
- When you click **Create Application** on this page, your application will be created.



Finished

Once your application has been generated, the final page of the form will load. Links to your new functionality will appear.



Users and Groups

- Users are those who can interact with the application.
- Groups can contain users or other groups as members.
- Groups can be further organized into group types.
- Group types allow you to classify your groups into different categories.



User Management

Users can be viewed from a few places, depending on your role.


- System administrators can access the Users page in the **Appian Administration Console** to view, create, and modify users.
- All designers can access users from the Users view in **Appian Designer**.
- The Users record type in **Tempo** contains the user profiles for users of Appian.



Creating a New User

System administrators can create new users by doing the following:

1. Navigate to the **Users** page of the Appian Administration Console. The Users grid is displayed.
2. Click the **Create** button in the toolbar above the grid. The **Create User** dialog is displayed.
3. Enter the user's **Username**.
4. Enter the remaining basic information for the user:
 - First Name
 - Last Name
 - Nickname (Optional)
 - E-mail Address (Must include the @ symbol and a domain such as .com)
 - Supervisor (Optional)
 - Title (Optional)



5. Select one of the following options from the **User Type** selection:

- **Basic User:** All users that are not system administrators. See below: Account Rights for Basic Users
- **System Administrator:** Users that have access privileges to all tools and capabilities in Appian, can edit user roles, and create new administrators. System administrators are automatically added to the Designers group through a membership rule, but do not inherently have the designer role.

6. Click **Create** to create the user and return to the Users grid, or click **Create & Add Another** to create the user and continue creating new users in the Create User dialog



Creating a Group

- All designers can create groups in Appian Designer.
- To create a group:
 1. Navigate to the application contents view of any application.
 - From here, you can optionally click on a group to open it and then directly create a child group.
 2. If you're within the application contents, click **New** and choose **Group** from the dropdown menu. If you're within a group, click **New Group**.
 3. Enter a name (required) and configure the group properties (optional).
 4. Click **Create**



Configuring Security for Groups

- Appian allows you to tailor user rights to the needs of your groups
- **Types of Group Membership**
 - Four types of membership determine the user rights available for a group - Administrator, Group Creator, Member and Viewer.
 - **Administrators** can modify group properties, add and remove administrators and members, create and modify membership rules, and delete the group.
 - **Group Creators** have administrator rights.
 - **Members** have been added as members either by the group creator or group administrators,



Security

Security in Appian in terms of Application and Object.



Application Security

- Users must have at least **Viewer** permissions to a *published* application in order to view its feeds and actions.
- Application security determines which groups and users can view, and interact with the application and its contents. By default, only the application creator and system administrators have access to the application.



Object Security

- Object security is an integral part of application development, and critical for ensuring that the right users and developers have the appropriate permissions within an application.
- Groups, role maps, security inheritance, layered security, and object visibility are important concepts to learn in order to fully understand object security.
- Appian recommends using only groups to set object security. This allows you to control object access by changing a user's group membership, rather than directly editing the object's role map.



Groups and Role Maps

- Object security is made up of two tightly coupled concepts: groups and role maps.
- Role maps are mappings between a series of groups or users and their permissions to an object. Each object in Appian has just one role map. To set an object's security, simply edit its role map.



Permission Levels in RoleMaps

- Each object accepts a different set of permission levels in its role map.
- In most cases when setting object security you will find the need to use only two permission levels: **Administrator** and **Viewer**. These can be defined as:
 - **Administrators** - Groups who have administrative permissions to an object in a specific environment. These permissions include the ability to fully edit or delete the object as desired.
 - **Viewers** - Groups who can interact with a particular object as an end user in Tempo, sites, or embedded. For example, granting a group **Viewer** rights to an interface gives them permission to view and interact with that interface from Tempo.



Permission Levels in RoleMaps

Deny Permission Level

Many objects offer **Deny** as a permission level. Giving a group the **Deny** permission level is equivalent to not listing that group within the role map, or not granting them any permissions. So when does it make sense to use **Deny**? It's most useful in situations where a group (*Group A*) should not have permissions to an object but might be nested within another group that should have permissions to it (*Group B*). In these situations, marking a group (*Group A*) with the **Deny** permission will overrule all of its other permissions.



Security Inheritance

- When an object inherits its security from a parent object, it means that it shares the same role map as its parent. When this is the case, any changes that are saved to the parent object's security role map are immediately reflected in the child's inherited role map.
- During application development, inheritance can be observed with top-level objects such as knowledge centers and rule folders. Knowledge centers and rule folders are considered top-level objects because their security is inherited by all objects nested within them by default.



Layered Object Security

- In Appian, security is layered. This means a user must have permissions to *every* object associated with an application's feature in order to see or interact with that feature.
- The benefit of applying layered object security is that it is possible to implement strict security models, and control security to objects and features at a granular level.