

RESUMEN

MANAGE USERS AND GROUPS

▷ **User Types**: Los grupos son uno de los primeros objetos que debes construir en tu aplicación.

Los grupos son importantes porque estos proporcionarán la fundación de seguridad y organización dentro de la aplicación.

En appian hay 2 tipos de Usuarios:

✦ Usuario Básico

✦ Usuario Administrador de sistema.

▷ De forma predeterminada, los usuarios básicos no tienen acceso a nada en una aplicación, hasta que les proporcionamos acceso.

Para dar a los usuarios básicos acceso a objetos y funciones de la aplicación, necesitamos agregarlos a los grupos.

▷ **System Admin**: A diferencia de los usuarios básicos tienen acceso administrativo predeterminado a todo el entorno de Appian, lo que incluye el acceso a las aplicaciones.

Pueden diseñar y administrar objetos, pero lo más importante es que pueden acceder a la consola de administración.

▷ En la consola de administración, los administradores pueden controlar configuraciones generales y también pueden crear nuevos usuarios o administrar los usuarios existentes.

▷ En Appian no se pueden eliminar usuarios, solamente puedes desactivar o reactivar usuarios.

▷ Para cambiar a un usuario básico en administrador debemos navegar a la vista de objetos, y buscar el grupo llamado

Designers → Add Member → Users to Add → Add

Los usuarios pueden ser creados a través de la autenticación LDAP y SAML, desde la consola de administración.

Why Groups?

Razones por las cuales se deben crear grupos.

▷ Seguridad: En cualquier aplicación debe haber grupos específicamente para!

* Desarrolladores: Quienes construyen o administran la aplicación

* Designers group:

▷ Acceso a la función de la Aplicación:

En cualquier aplicación debería haber un grupo para usuarios de negocios.

▷ Tareas y mensajería: En cargado de asignar una tarea a un grupo de usuarios

Application Groups

Cada aplicación en Appian contendrá un conjunto único de grupos que ayudarán a administrar la seguridad de la aplicación y el acceso a las funciones de la app.

Todas las aplicaciones deben contener los siguientes grupos.

• All users • Administrators • Functional and Role-Based

* Functional and Role-Based, son un grupo creado por los usuarios de negocios.

* Buena Práctica *

- * Solo crea grupos que necesites para seguridad y tareas en el momento.
- * No crees grupos antes de tiempo
- * Los grupos no deben reflejar un organigrama completo
- * Crea grupos sobre la marcha
- * Administrators es un grupo de Desarrolladores

CREATE GROUPS

- Crear Todos los Usuarios (All Users). Este será el grupo principal.
- Crear el Grupo de Administradores (Y use este grupo como seguridad para proteger a todos los usuarios (All Users).
- Otros grupos heredarán la seguridad del grupo principal (All Users)
- * No añadas usuarios a All Users, solo grupos.

▷ Añade usuarios a los grupos

▷ Añade desarrolladores al grupo de administración

▷ Añade usuarios por funciones o basada en roles.

▷ Protege tu aplicación

BASIC EXPRESSION BUILDING

Transform Your Data

Las reglas de expresion realizan una amplia gama de funciones, incluida la recuperación, el formato y la transformación de los datos.

Las expresiones pueden ser creadas y usadas en varios objetos de Appian.

Se pueden crear reglas de expresiones unicas usando el editor de expresiones.

Expression Editor		
Editor Donde se define la regla	Seccion de pruebas	Entrada de la regla * Para crear variables en la regla
Seccion de Documentación Ayuda al usuario otras reglas	Para verificar el funcionamiento de la regla	que representen datos dinamicos

DATA TYPE

- Texto (text)
- Lista de numeros (integer) (text)
- Boolean
- Dates
- Array

- * Primitive
- * Complex
- Data Subset
- CDT

➤ cast() Function: Convierte un valor de un tipo dado, a otro tipo de dato especificado.

➤ todatasubset() Function: Para buscar un tipo de dato de un subconjunto de datos

D Custom Data Type (CDT): Es un objeto que contiene campos definidos por el usuario, asociados a su tipo de datos.

CDT: Es usado para agrupar datos relacionados que se utilizaron como referencia en toda la aplicación.

CDTs son una gran manera de asegurarse de que los datos relacionados permanezcan juntos.

ADDITIONAL FUNCTIONS

* `if()` Evalúa si lo que está en paréntesis es verdadero o falso.

* `or()` Retorna verdadero si uno de los valores es verdadero, retorna falso si todos los valores son falsos.

* `and()`: Retorna verdadero solo cuando todas las condiciones se cumplan.

* `today()` Regresa la fecha actual (8/4/2022)
m/d/yyyy

* `now()` Retorna la fecha y hora actual

* `calworkdays()` Puede retornar el número de días trabajados entre dos fechas.

```
calworkdays(  
  now(),  
  now()+30  
)
```

* `isusermemberofgroup()`: Verifico si un usuario es parte de un grupo.

SAZL - Self-Assembling Interface Layer.

a! - Funciones para cada componente en una pantalla

r! - ruleinput

a! - Usada para llamar varios sistemas y conectar funciones

VARIABLES

a! isNullOrEmpty - Verifica si una variable es nula o esta vacía.

Variables locales

```
a! localVariables {  
    local! a : 1,  
    local! b : 2,  
}
```

local! a + local! b

ARRAYS

Una colección de datos es considerado en tipo de dato completo.

{ 2022, "Hda", "Red" 2.89 } } Lista de tipos de datos delante

El primer dato se encuentra en la posición 1

{ ... } [2] - Indica que buscare el elemento del index 2.

EJERCICIOS

Acme Exercise

IMPORT PACKAGE PACKAGES COMPARE AND DEPLOY

ALL OBJECTS PLUG-INS UNREFERENCED OBJECTS

Name or description

Clear Filters

OBJECT TYPE

- Connected System
- Constant
- Data Store
- Data Type
- Decision
- Document
- Expression Rule
- Feed
- Folder
- Group
- Integration
- Interface
- Process Model
- Query Rule
- Record Type
- Report

Group Security

Tip: Groups inherit security from their parent group, if they have one. Group visibility and membership are configured in the group's properties, rather than its role map. [Learn more](#)

Name
AX All Users

User or Group	Permission Level	
Default (All Other Users)	No Access	
AX Administrators	Administrator	X

[Add Users or Groups](#)

CANCEL SAVE CHANGES

All Application Objects

Last Modified

- 7/27/2022 1:21 PM by J
- 7/27/2022 1:15 PM by J
- 7/27/2022 1:14 PM by J
- 7/27/2022 1:14 PM by J
- 7/27/2022 1:12 PM by J

Acme Exercise

IMPORT PACKAGE PACKAGES COMPARE AND DEPLOY

ALL OBJECTS PLUG-INS UNREFERENCED OBJECTS

Name or description

Clear Filters

OBJECT TYPE

- Connected System
- Constant
- Data Store
- Data Type
- Decision
- Document
- Expression Rule
- Feed
- Folder
- Group
- Integration
- Interface
- Process Model
- Query Rule
- Record Type

Rule Folder Security

Tip: Rule folders without parents cannot inherit security. Rule folders with a parent inherit security by default. [Learn more](#)

Name
AX Rules & Constants

User or Group	Permission Level	
Default (All Other Users)	No Access	
AX Administrators	Administrator	X
AX All Users	Viewer	X

[Add Users or Groups](#)

CANCEL SAVE CHANGES

All Application Objects

Last Modified

- 7/26/2022 1:51 PM by J
- 7/26/2022 1:47 PM by J
- 7/26/2022 1:44 PM by J
- 7/26/2022 1:41 PM by J
- 7/26/2022 1:33 PM by J
- 7/26/2022 1:23 PM by J
- 7/26/2022 1:20 PM by J

Acme Exercise

IMPORT PACKAGE PACKAGES COMPARE AND DEPLOY

ALL OBJECTS PLUG-INS UNREFERENCED OBJECTS

Name or description

Clear Filters

OBJECT TYPE

- Connected System
- Constant
- Data Store
- Data Type
- Decision
- Document
- Expression Rule
- Feed
- Folder
- Group
- Integration
- Interface
- Process Model
- Query Rule
- Record Type

Process Model Security

Tip: Neither process model folders nor the process models within them inherit security from their parent. Security must be set on each process model individually. [Learn more](#)

Name
AX Process Models

User or Group	Permission Level	
Default (All Other Users)	No Access	
AX Administrators	Administrator	X
AX All Users	Viewer	X

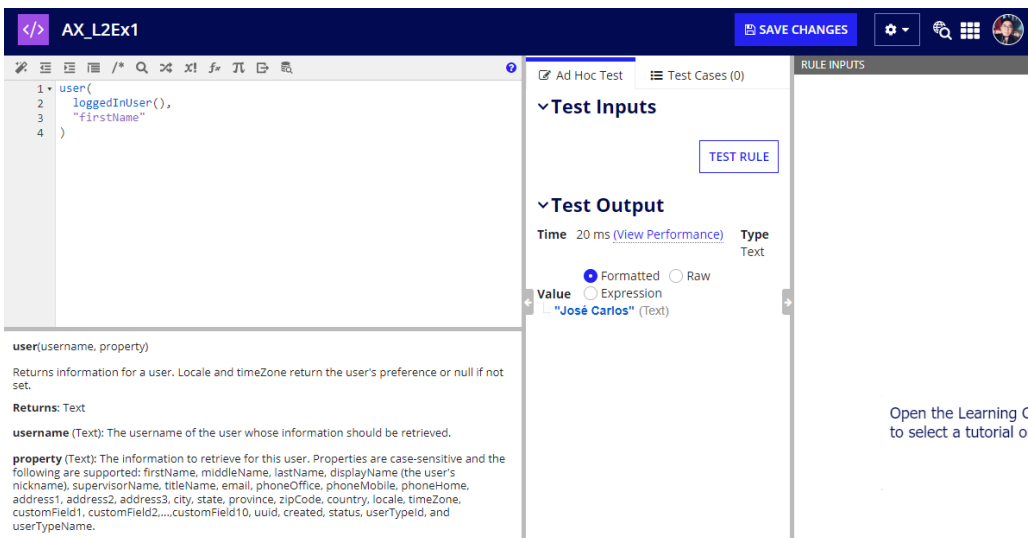
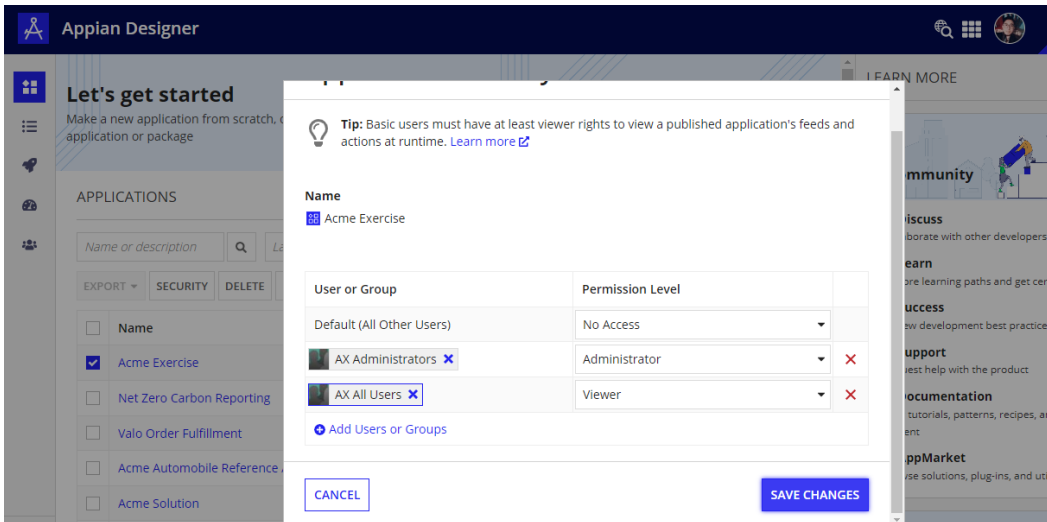
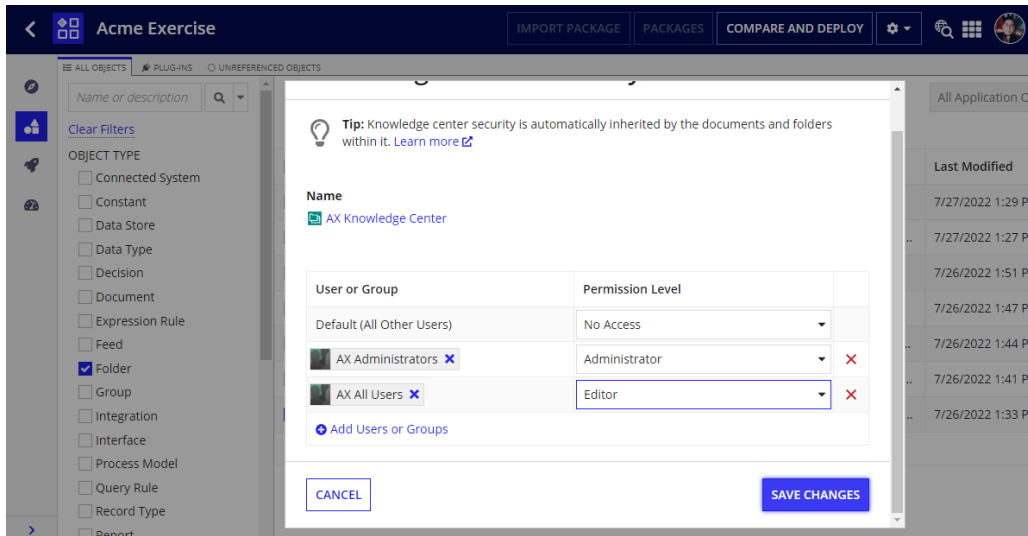
[Add Users or Groups](#)

CANCEL SAVE CHANGES

All Application Objects

Last Modified

- 7/27/2022 1:27 PM by J
- 7/26/2022 1:51 PM by J
- 7/26/2022 1:47 PM by J
- 7/26/2022 1:44 PM by J
- 7/26/2022 1:41 PM by J
- 7/26/2022 1:33 PM by J
- 7/26/2022 1:23 PM by J



Open the Learning C
to select a tutorial or

AX_L2Ex1 SAVE CHANGES

```

1 user(
2   loggedInUser(), "firstName"
3 ) & " created on " &
4 user(
5   loggedInUser(), "created"
6 )

```

Ad Hoc Test | Test Cases (0)

Test Inputs

Test Output

Time 7 ms (View Performance) Type Text

Value Formatted Raw Expression

"José Carlos created on 7/25/2022 1:12 PM CDT" (Text)

Place cursor on function, rule, or constant to display help

Open the Learning Cen to select a tutorial or vi

AX_L2Challenge SAVE CHANGES

```

1 text(
2   datevalue("07/27/2022"),
3   "mmmm dd, yyyy"
4 )

```

Ad Hoc Test | Test Cases (0)

Test Inputs

Test Output

Time 1 ms (View Performance) Type Text

Value Formatted Raw Expression

"July 27, 2022" (Text)

datevalue(date_text)
 Converts the given date into an equivalent serial number.
Returns: Date
date_text (Text): The date to be converted into a serial number.

AX_L4Ex1 SAVE CHANGES

```

1 if(rilinum1 > rilinum2,
2   "Sell", "Do not sell"
3 )
4

```

Ad Hoc Test | Test Cases (0)

Test Inputs

Rule Input Name	Expression	Value
num1 (Number (Integer))	1	100
num2 (Number (Integer))	1	200

Save as Test Case TEST RULE

Test Output

Time 1 ms (View Performance) Type Text

Value Formatted Raw Expression

"Do not sell" (Text)

Place cursor on function, rule, or constant to display help

Open the Learning Cen to select a tutorial or vi

AX_L4Ex1

```

1 if(r1num1 > r1num2,
2   "Sell", "Do not sell"
3 )
4

```

SAVE CHANGES

Ad Hoc Test Test Cases (0)

Test Inputs

Rule Input Name	Expression	Value
num1 (Number (Integer))	1	100
num2 (Number (Integer))	1	20

Save as Test Case TEST RULE

Test Output

Time 1 ms (View Performance) Type Text

Formatted Raw

Value Expression

"Sell" (Text)

Place cursor on function, rule, or constant to display help

Open the Learning Center to select a tutorial or v

AX_L4Ex2

```

1 calworkdays(
2   date(2021, 07, 01),
3   date(2021, 07, 31)
4 )

```

SAVE CHANGES

Ad Hoc Test Test Cases (0)

Test Inputs

TEST RULE

Test Output

Time 40 ms (View Performance) Type Number (Integer)

Formatted Raw

Value Expression

22 (Number (Integer))

Place cursor on function, rule, or constant to display help

Open the Learning Center to select a tutorial or v

AX_L4Challenge

```

1 if(
2   len(r1cadena) > r1lonMax,
3   "No debe exceder los " & r1lonMax & " caracteres",
4   "Gracias por la informacion"
5
6
7
8 )

```

SAVE CHANGES

Ad Hoc Test Test Cases (0)

Test Inputs

Rule Input Name	Expression	Value
cadena (Text)	1	Hola mundo
lonMax (Number (Integer))	1	5

Save as Test Case TEST RULE

Test Output

Time 1 ms (View Performance) Type Text

Formatted Raw

Value Expression

"No debe exceder los 5 caracteres" (Text)

if(condition, valueIfTrue, valueIfFalse)
Returns valueIfTrue if condition returns true; returns valueIfFalse otherwise.

Returns: Any Type

condition (Boolean): A test that determines whether valueIfTrue or valueIfFalse will be returned.

valueIfTrue (Any Type): The value to be returned if condition returns true.

valueIfFalse (Any Type): The value to be returned if condition returns false.

Open the Learning Center to select a tutorial or v

AX_L4Challenge SAVE CHANGES

```

1 if(
2   len(rilcadena) > rilonMax,
3   "No debe exceder los " & rilonMax & " caracteres",
4   "Gracias por la informacion"
5
6
7
8 )

```

if(condition, valueIfTrue, valueIfFalse)
Returns valueIfTrue if condition returns true; returns valueIfFalse otherwise.

Returns: Any Type

condition (Boolean): A test that determines whether valueIfTrue or valueIfFalse will be returned.

valueIfTrue (Any Type): The value to be returned if condition returns true.

valueIfFalse (Any Type): The value to be returned if condition returns false.

Test Inputs

Rule Input Name	Expression	Value
cadena (Text)	1	Hola
lonMax (Number (Integer))	1	5

Save as Test Case TEST RULE

Test Output

Time 1 ms (View Performance) Type Text

Value Formatted Raw

Value Expression

["Gracias por la informacion" (Text)]

Open the Learning Center to select a tutorial or

AX_L5Ex1 SAVE CHANGES

```

1 if(
2   (rilretMileage - rilstrMileage) > 50000,
3   "Correr diagnostico completo",
4   "Regresa a inventario"
5 )

```

if(condition, valueIfTrue, valueIfFalse)
Returns valueIfTrue if condition returns true; returns valueIfFalse otherwise.

Returns: Any Type

condition (Boolean): A test that determines whether valueIfTrue or valueIfFalse will be returned.

valueIfTrue (Any Type): The value to be returned if condition returns true.

valueIfFalse (Any Type): The value to be returned if condition returns false.

Test Inputs

Rule Input Name	Expression	Value
retMileage (Number (Integer))	1	60000
strMileage (Number (Integer))	1	1000

Save as Test Case TEST RULE

Test Output

Time 1 ms (View Performance) Type Text

Value Formatted Raw

Value Expression

["Correr diagnostico completo" (Text)]

Open the Learning Center to select a tutorial or

AX_L5Ex1 SAVE CHANGES

```

1 if(
2   (rilretMileage - rilstrMileage) > 50000,
3   "Correr diagnostico completo",
4   "Regresa a inventario"
5 )

```

if(condition, valueIfTrue, valueIfFalse)
Returns valueIfTrue if condition returns true; returns valueIfFalse otherwise.

Returns: Any Type

condition (Boolean): A test that determines whether valueIfTrue or valueIfFalse will be returned.

valueIfTrue (Any Type): The value to be returned if condition returns true.

valueIfFalse (Any Type): The value to be returned if condition returns false.

Test Inputs

Rule Input Name	Expression	Value
retMileage (Number (Integer))	1	1000
strMileage (Number (Integer))	1	6000

Save as Test Case TEST RULE

Test Output

Time 1 ms (View Performance) Type Text

Value Formatted Raw

Value Expression

["Regresa a inventario" (Text)]

Open the Learning Center to select a tutorial or

AX_L5Ex2 SAVE CHANGES

```

1 localVariables(
2     localSum: r1first + r1second,
3     "The new total is " & localSum
4 )

```

Place cursor on function, rule, or constant to display help

Test Inputs

Rule Input Name	Expression	Value
first (Number (Integer))	1	4
second (Number (Integer))	1	8

Save as Test Case TEST RULE

Local Variables

Test Output

Time 2 ms (View Performance) Type Text

Formatted Raw

Value Expression

"The new total is 12" (Text)

RULE INPUTS

Name	Type	Arr
first	Number...	
second	Number...	

Open the Learning Center to select a tutorial or

AX_L5Challenge SAVE CHANGES

```

1 if(
2     isnull(r1value),
3     "Null Value",
4     r1value
5 )

```

if(condition, valueIfTrue, valueIfFalse)

Returns valueIfTrue if condition returns true; returns valueIfFalse otherwise.

Returns: Any Type

condition (Boolean): A test that determines whether valueIfTrue or valueIfFalse will be returned.

valueIfTrue (Any Type): The value to be returned if condition returns true.

valueIfFalse (Any Type): The value to be returned if condition returns false.

Test Inputs

Rule Input Name	Expression	Value
value (Number (Integer))	1	

Save as Test Case TEST RULE

Test Output

Time 1 ms (View Performance) Type Text

Formatted Raw

Value Expression

"Null Value" (Text)

RULE INPUTS

Name	Type	Arr
value	Number...	

Open the Learning Center to select a tutorial or

AX_L5Challenge SAVE CHANGES

```

1 if(
2     isnull(r1value),
3     "Null Value",
4     r1value
5 )

```

if(condition, valueIfTrue, valueIfFalse)

Returns valueIfTrue if condition returns true; returns valueIfFalse otherwise.

Returns: Any Type

condition (Boolean): A test that determines whether valueIfTrue or valueIfFalse will be returned.

valueIfTrue (Any Type): The value to be returned if condition returns true.

valueIfFalse (Any Type): The value to be returned if condition returns false.

Test Inputs

Rule Input Name	Expression	Value
value (Number (Integer))	1	4

Save as Test Case TEST RULE

Test Output

Time < 1 ms (View Performance) Type Number (Integer)

Formatted Raw

Value Expression

4 (Number (Integer))

RULE INPUTS

Name	Type	Arr
value	Number...	

Open the Learning Center to select a tutorial or

AX_L6Ex1

SAVE CHANGES

```
1 {"Red", "Toyota", "Prius"}
```

Ad Hoc Test Test Cases (0)

Test Inputs

Test Output

Time 1 ms (View Performance) Type List of Text String

Value Formatted Raw Expression

List of Text String - 3 items

- "Red" (Text)
- "Toyota" (Text)
- "Prius" (Text)

Place cursor on function, rule, or constant to display help

Open the Learning Cent to select a tutorial or v

AX_L6Ex1

SAVE CHANGES

```
1 {"Red", "Toyota", "Prius", 15500}
```

Ad Hoc Test Test Cases (0)

Test Inputs

Test Output

Time < 1 ms (View Performance) Type List of Variant

Value Formatted Raw Expression

List of Variant - 4 items

- "Red" (Text)
- "Toyota" (Text)
- "Prius" (Text)
- 15500 (Number (Integer))

Place cursor on function, rule, or constant to display help

Open the Learning Cent to select a tutorial or vic

AX_L6Ex1

SAVE CHANGES

```
1 {
2   Color: "Red",
3   Make: "Toyota",
4   Model: "Prius",
5   Mileage: 15500
6 }
```

Ad Hoc Test Test Cases (0)

Test Inputs

Test Output

Time 1 ms (View Performance) Type Dictionary

Value Formatted Raw Expression

Dictionary

- Color "Red" (Text)
- Make "Toyota" (Text)
- Model "Prius" (Text)
- Mileage 15500 (Number (Integer))

Place cursor on function, rule, or constant to display help

Open the Learning Cer to select a tutorial or v

AX_L6Ex1 SAVE CHANGES

```
1 index ({
2   color: "Red",
3   Make: "Toyota",
4   Model: "Prius",
5   Mileage: 15500
6 }, "Model")
```

Ad Hoc Test Test Cases (0)

Test Inputs

Test Output

Time 1 ms ([View Performance](#)) Type Any Type (Text)

Value Formatted Raw Expression

"Prius" (Text)

Place cursor on function, rule, or constant to display help

Open the Learning Cen
to select a tutorial or vi