About this Guide

Who is the audience for this guide?

This guide introduces **professional services firms** and **developers** to Crowd Machine's Crowd App Studio.

What is Crowd App Studio?

Crowd App Studio is a **no-code application development environment** for the design, development and management of business applications, it automates most of the tasks involved in building and deploying applications.

Crowd App Studio reduces the dependence on large project teams and puts data professionals and developers in control of application delivery.

Assumed Skills

It is assumed that the reader has a reasonable level of professional experience working with computer systems and data analytics which may include any of the following.

- MS Excel
- MS Access
- Visual Basic for Applications
- SQL

This guide will also be applicable for experienced developers who wish to expedite existing application deployment through Crowd App Studio.

Extra Learning Links

Throughout this guide you will find links to extra learning material where you see these icons.

Video tutorials of the demo app.

Knowledge Base articles and glossary.

Crowd Academy guides and other guides.

► **NOTE:** You must be logged into your Crowd Machine account to access links to Knowledge Base articles.

Need more information?

Post a question to the Crowd Machine forum in two ways:

- On the **console page** - click the **Forum** button.
- From the Crowd Machine **navigation menu** - click **Resources** and select **Forum**.

Pop-up Blockers on your Browser

If you have problems viewing Crowd Machine links or pages, go to your browser settings and set them to always allow pop-ups from **https://studio.crowdmachine.com**

Access Crowd App Studio

Visit crowdmachine.com and click the **Login** button.

Alternatively type the address in a browser window:

https://studio.crowdmachine.com/design/#/portal/login

Bookmark the address to make it easier to access in the future.

Next, read the learning objectives.

Learning Objectives

This guide groups topics and learning objectives into the following sections.

Build out an Application Data Structure

Topic	Learn About...
Packages	Creating a data structure using Packages.
Package Attributes	Adding Attributes to store data values.
Attribute Types	Setting Attribute Types to store different types of data (text, numbers).
User Packages	The Organizational hierarchy that supports application users.
Package Relationships	Linking Packages to form relationships.

Design an Application User Interface

Topic	Learn About...
Page Designer	Using Page Designer to design page elements.
View the Sales Activity	Viewing a visual representation of a UI Activity in Page Designer
Page Layout	Designing the page layout for the Sales Application.
Include Panel	Dynamically displaying data using an Include Panel.

Making the Application Work

Topic	Learn About...
Running Patterns	Viewing Patterns at Runtime.
Relationships in Action	Configuring relationships to display application data.
Rules	Writing rules to add logic to the application.

Adding Email Functionality

Topic	Learn About...
Email Activity	Using an Email Activity to add an email notification.
System Settings Package	Configuring the email settings.
Email Package	Constructing the email.

Create an Integration Endpoint

Topic	Learn about...
Integration Activity	Creating a public API endpoint for our application.
Temporary Attributes Package	Using temporary attribute placeholders within activities.
Set Rule	Filtering a package retrieval by a condition.

Create a Portal

Topic	Learn about...
Portals	Creating a Portal to provide access to an application.

Next, learn about the Crowd Machine terminology and toolset.

Terminology and Toolset

Whenever you start to use a new platform there is always new terminology to learn, different interfaces to get used to, and new tools to discover - the Crowd Machine platform is no different.

To help get you started with the Crowd Machine environment, we have produced a series of worksheets that introduce you to the Crowd App Studio toolset and terminology.

Worksheet name

We recommend that you print out these worksheets for quick and easy reference as you learn about the Crowd Machine platform.

- What is an App?
- Crowd Machine Application Topology and Terminology
- Crowd App Studio Navigation
- Console and Main Navigation Menu
- Understanding Packages
- Package Activity Definition (Matrix)
- Define a UI Activity Content

You can learn about the Crowd App Studio toolset in this guide.

- Crowd App Studio

Next, we will learn about the Sales Application which we will use to demonstrate the Crowd App Studio functionality.

The Sales Application

This guide provides an example of a Sales Application that demonstrates the critical functionality that you will need to understand to build and maintain Crowd Machine applications.

- We recommend **downloading** the Sales Application from Crowd Share to use in conjunction with this guide - see instructions below.
- Alternatively, you can follow the steps in this guide to **create your own** Sales Application.

About the Sales Application

The Sales application is a simple invoicing application that allows users to create invoices, and update associated line item values.

Each time an update is made, an email is sent to a specified email address.

The list of invoices created will be viewable in a Google Sheets doc and will update in real time.

 Video: The Sales Application at Runtime

View or Download the Sales Application

- You can **view** the Sales Application.
- You can **download** the Sales Application - see the next topic heading.

Next, learn how to download the Sales Application from Crowd Share.

Download the Sales Application

You can download the Sales Application from Crowd Share by following these steps.

Steps

1. Log into Crowd App Studio.
   The Crowd App Studio Console displays.

2. Click the **Crowd Share** button (top of the page).
   The Crowd Share home page displays.

3. Click **Tutorials**.
   The Sales Application card displays.

4. Click on the text **Sales Application**.
   The download page displays.

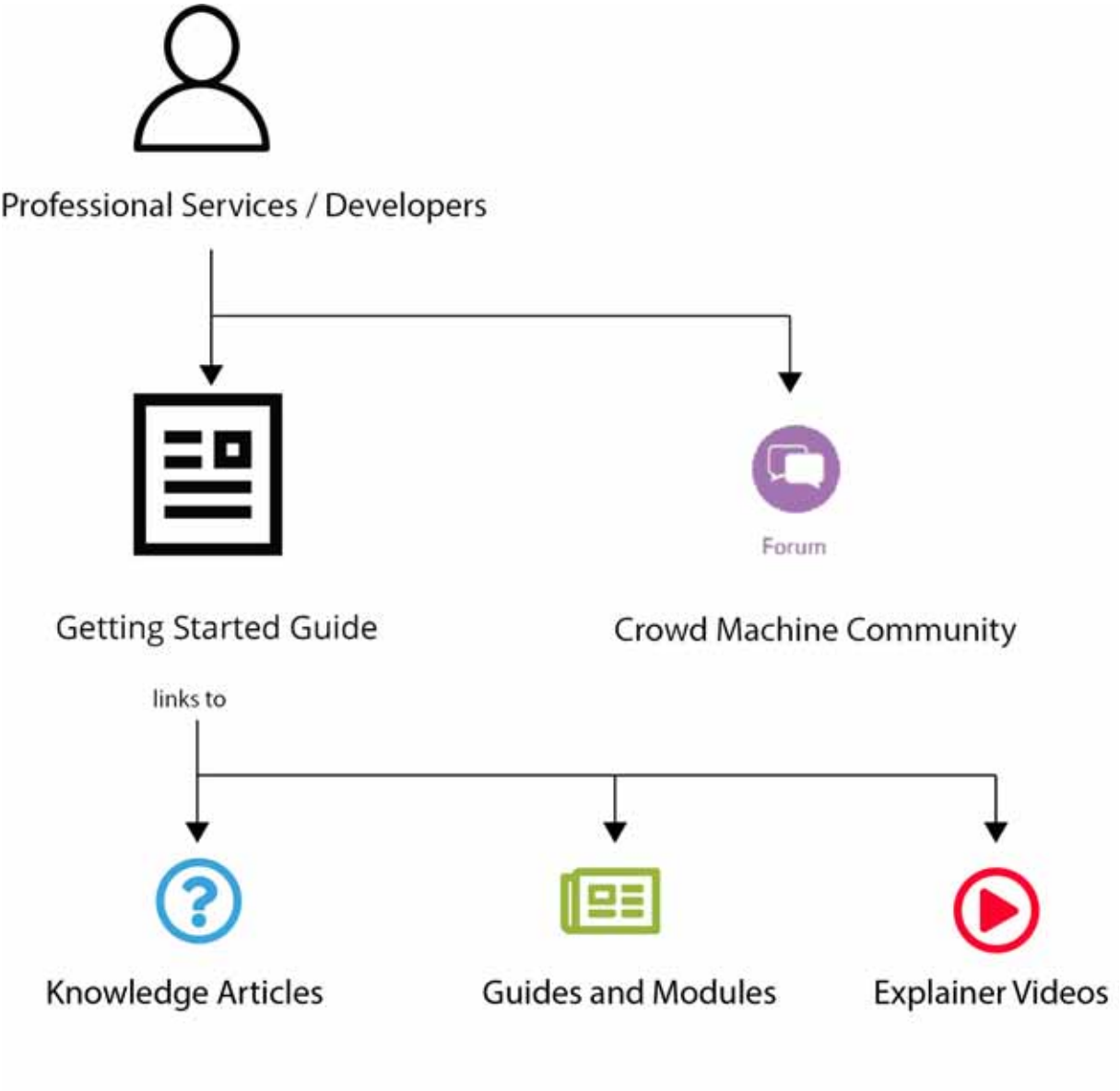
5. Click **Download**.
   A download destination org is displayed.

6. Click **Download**.
   The Sales Application downloads to your organization and is available to select in Crowd App Studio.

You can now access the Sales Application in Crowd App Studio.

Learning Pathways

This guide provides a high-level overview with links to learning pathways that provide more detailed Crowd App Studio “How to” knowledge articles, Crowd Academy guides and explainer videos.



Create your own Sales Application

You can create your own version of the Sales Application by following the steps in this guide and watching the video tutorials.

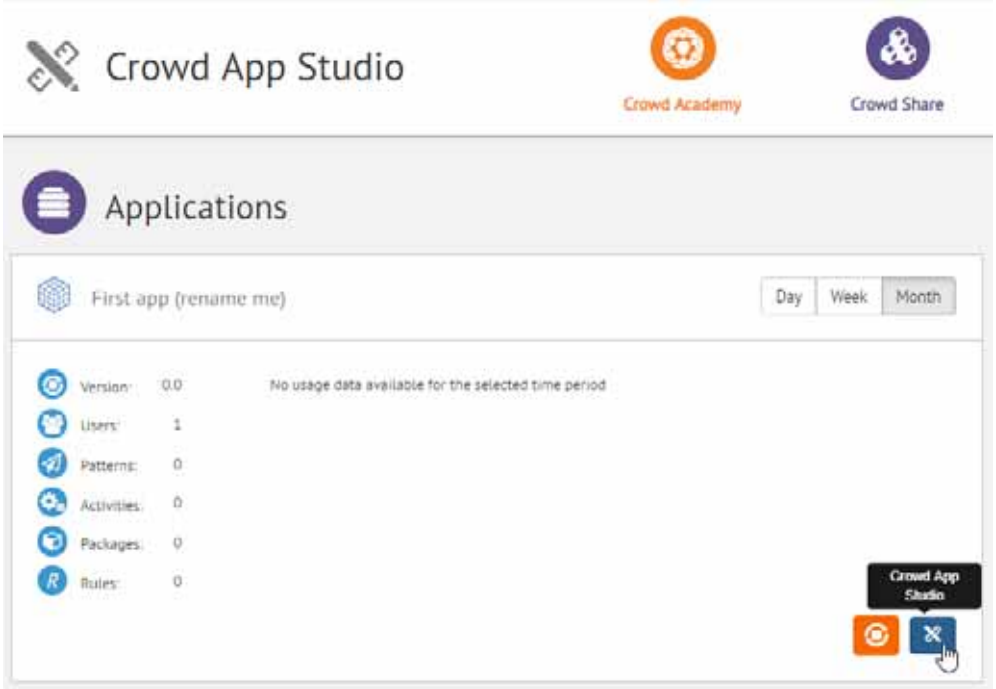
Video: Create the Sales Application

If this is your first time creating a Crowd Machine application, follow these steps to create an application name.

Steps

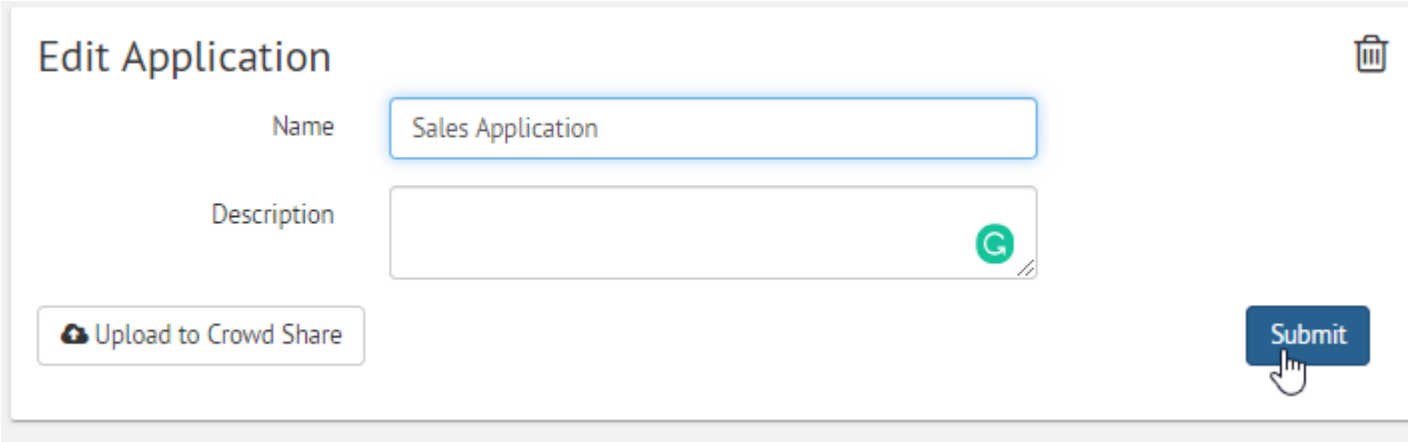
1. Log into Crowd App Studio.
   https://studio.crowdmachine.com/design/#/portal/login

2. On the console screen, locate **First App (rename me)** and click the Crowd App Studio button.



The Edit Application panel displays.

3. Click in the **Edit Application Name** field and delete First App (rename me).
4. Type the name **Sales Application**.
5. Click **Submit**.



Next, you will learn about Packages.

Packages

What is a Package?

Packages form an application's data structure.

Glossary definition of a Package

By data structure, we mean the underlying **collection of data values** and the **data relationships** that form the basis of an application.

Once a data structure is in place, definitions and rules can then be applied to the data to produce application behavior.

Much like a table in a spreadsheet, packages consist of a package name (table name), and Attributes (table columns).

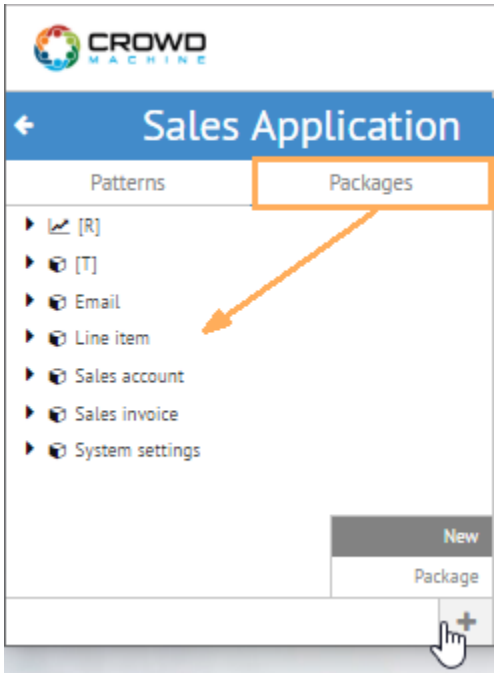
To Create a Package

To create a package, log into the Crowd App Studio and locate the **Packages** tab in the Organizational Explorer.

How to Create a Package

The Organizational Explorer

The Organizational Explorer is located on the left of the workspace and it is used to add data packages and other essential application components known as patterns.



Next, we will learn about Package Attributes.

Package Attributes

Attributes are like table columns.

The example below shows a basic Package, represented in spreadsheet form.

The columns represent **Attributes** which store data values such as **# Line items** and **Net amount** for Sales Invoices stored in the Package.

Sales Invoices					
Attributes	# Line items	Detail	Payment cleared?	Net amount	Transaction date
value 1	5	INV-2548	TRUE	12.50	31/12/2018
value 2	10	INV-2549	FALSE	12,365.25	12/01/2019
value 3	15	INV-2550	TRUE	4,385.46	03/07/2012

- **Rows** in the table represent different instances of the Package; in this case, different sales invoices.
- **Columns** in the table (attributes) contain different values, and different data types.

We therefore need to map our **data types** into the accepted formats for the Crowd App Studio.

If you've ever worked with complex data or programming languages, the range of data types can be overwhelming.

Crowd Machine makes this easy with Attribute Types.

Next, learn about Attribute Types.

Attribute Types

When we create a Package, Attributes can be set as any of the following types.

How to Create an Attribute

This table below shows the eight different Attribute Types.

- Setting an attribute **data type** specifies what type of data to hold, eg. text, number, etc.

Attribute Type	Description	You may know me as...
Text	Any alphanumeric input. Maximum length of 256 characters	string
Number (Decimal)	Any non-integer number value	single, double, long, float, complex,
Number (non-decimal)	A number with no fractional component	int, integer
Date/Time	An ISO-formatted date/time	time, date, datetime
True/False	Simple yes/no value	bool, boolean
Memo	For use when text inputs exceed text field input.	Long text
Document	Media. Can also be used to display images on-page	File
Timespan	Time difference between two points in time	TimeSpan

Next, learn about creating the Sales Package.

Sales Application Package Structure

You can create the Sales Application package structure by replicating the Packages and Attributes below.

Video: Create the Sales Application Package Structure

Steps

1. In the Organizational Explorer, click the **Packages** tab.
2. Create the **Sales Account** package and the attribute shown in the table below.
3. Create the **Sales Invoice** package and the attributes shown in the table below.
4. Create the **Line Item** package and the attributes shown in the table below.

Sales Account Package

Attribute Name	Data Type
ID	Text

Sales Invoice Package

Attribute Name	Data Type
# Line Items	Number (non-decimal)
Detail	Text
Payment Cleared?	True/False
Net Amount	Number (decimal)
Transaction Date	Date/Time

Line item Package

Attribute Name	Data Type
ID	Text
Product Code	Text
Quantity	Number (non-decimal)
Tax Rate	Number (decimal)

Unit Price	Number (decimal)

Next, we will learn about User Packages, the Directory and Organizations.

User Packages, the Directory and Organizations

This section describes the user hierarchy so that you understand how users are added to an application.

- In this exercise, is NOT necessary to create a user hierarchy for the Sales Application. Later in the guide, you will create a **User package** to add email functionality.

User Packages

User Attributes are also managed as a Package, but separately from other Packages. The Package labeled User contains generic Attributes relative to a user of an application.

How to Create application users

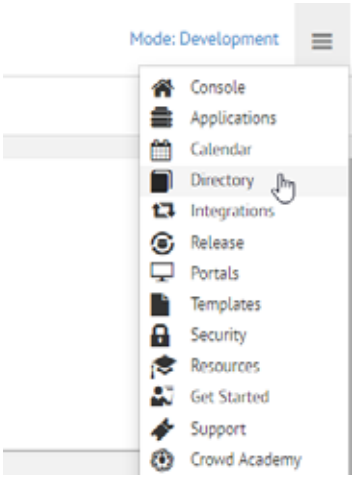
You can see these Attributes by selecting the User Package from the Organizational Explorer.

The details for the package will be displayed on the Stage and the Attributes can be seen listed in the Attributes column.

Additional attributes can be added to the User Package, which can be used to manage access permissions and preferences.

The Directory

As Crowd Machine automates application security, User Packages are managed in the Directory.



The Organization

The Directory is used to manage the Organization - the highest level platform construct that enforces security for the applications and the users it contains.

An Organization is automatically created when you set up a Crowd Machine account.

Next, learn about Package Relationships.

Package Relationships

In the same way that multiple tables can be linked together to form a relational database, packages can also be linked, by using Relationships. In a spreadsheet context, this could be compared to having multiple tables on multiple tabs, each indexing and calculating data from across various ranges within the workbook.

Packages can be related in one of two ways, known as cardinality. The relationship is always structured in a source/target manner.

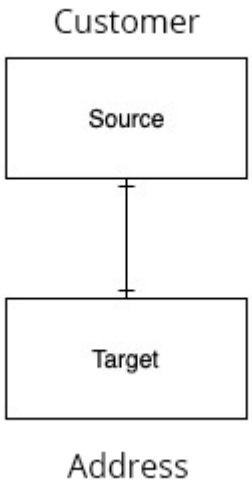
► **Important note:** Before starting work in the Crowd App Studio, you should first spend some time deciding on the design of your data structure.

It’s important that the data design is in diagram form.

Planning a data structure in this way reduces the risk of reworking your application at some point in the future.

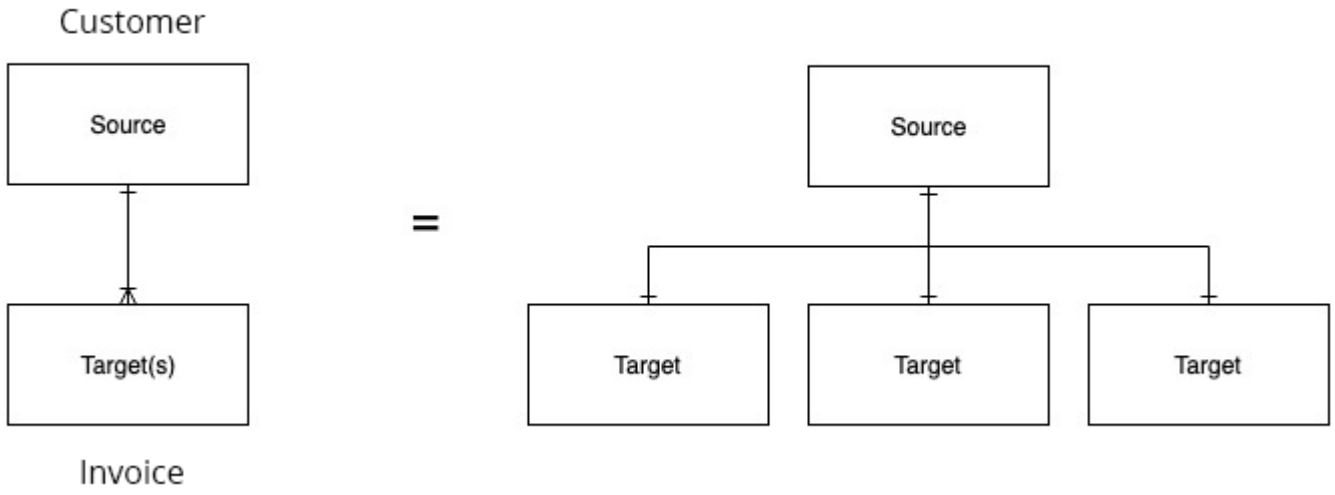
One-to-one

One source package is related to one target package.



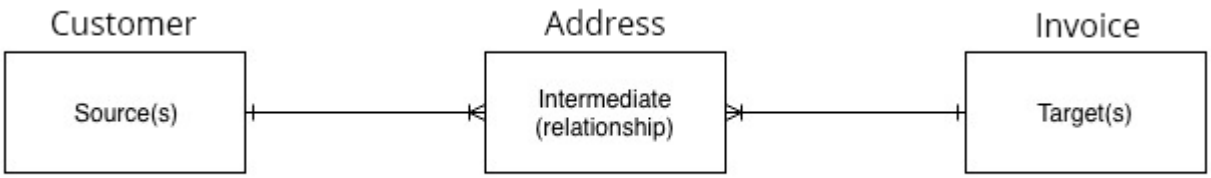
One-to-many

One source package is related to many target packages.



Many-to-many

A many-to-many relationship can be achieved by using an intermediary package. This is required where multiple sources and multiple targets can be related in the same way, but unique relationships are required.



Any data pertinent to the relationship can be stored on the intermediary package.

For more information on relationships refer to this guide:

Introduction to Data Relationships

Next, learn about creating relationships for the Sales Application data structure.

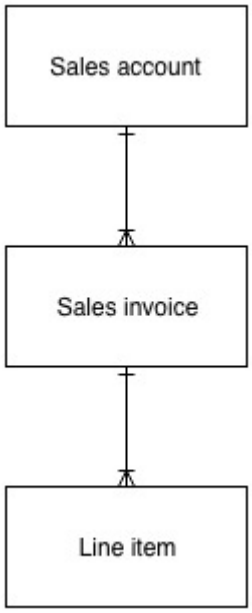
Sales Application Structure

We must first design a data structure before implementing the application in the Crowd App Studio.

Video: Creating Data Relationships for Sales Account and Line Item

For the Sales Application, we'll expand our data structure to both sides of the Sales invoice.

- The Sales account in which it resides, and
- The constituent Line items for each.



To implement the Sales invoice data structure in the Crowd App Studio, we first need to create relationships between the Sales account and Line item Packages that we created earlier.

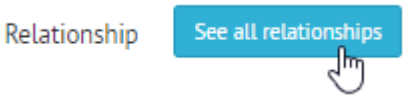
Each relationship in the above data design is one-to-many from source to target. They'll therefore both be linked in the same way.

How to Create a Linked Package

Steps

To create the Sales Account -> Sales Invoice (one-to-many).

1. Navigate to the Sales Account Package (source package) and click **See all Relationships**.



2. Click the **plus** button, select **Sales Invoice** and set the cardinality to **one-to-many**.

To create the Sales Invoice -> Sales Line Item (one-to-many).

1. Navigate to the Sales Invoice Package (source package) and click **See all Relationships**.
2. Click the **plus** button, select **Sales Line Item** and set the cardinality to **one-to-many**.

Next, learn about using Packages in Patterns and Activities.

Using Packages in Patterns and Activities

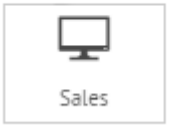
We build out applications by creating Patterns that contain one or more Activities.

Video: Creating the Sales Application Folder, Pattern and Activity

Activities

In order to manipulate, display and report on packages and their underlying data, Crowd Machine uses activities.

An **activity** is represented by an icon.

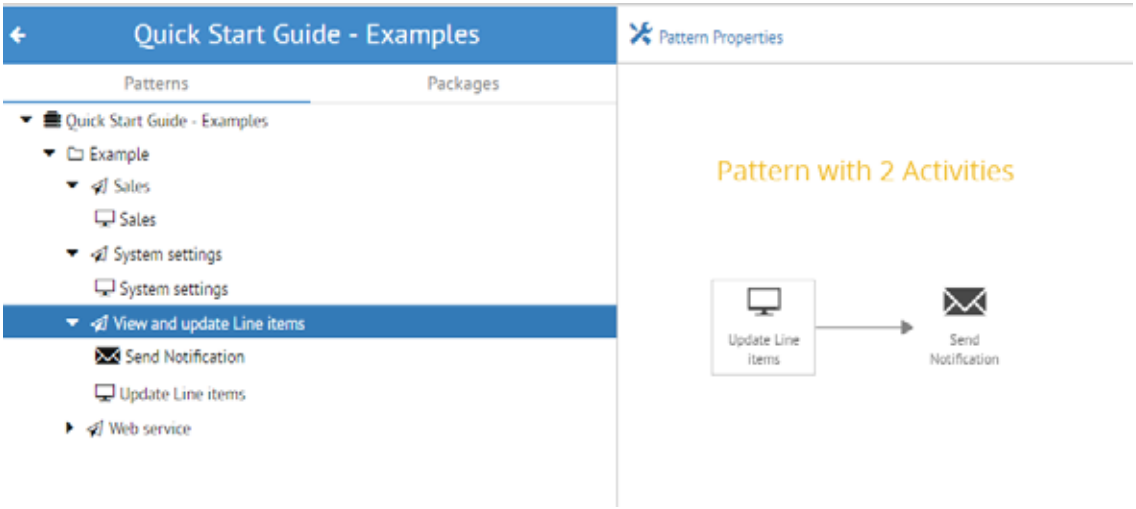


Activities are single operations and can be run independently, or in sequence.

Patterns

Patterns are containers for one or more Activities, and can be run individually or in conjunction with other patterns, to form an application.

Patterns can be arranged using **folders**, to which batch permissions can be applied.



Next, learn about Folders.

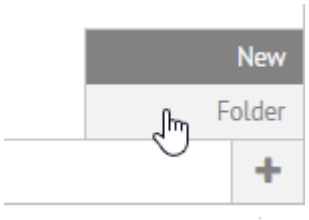
Folders

Folders are a useful way to group patterns by category, or to group patterns by permissions.

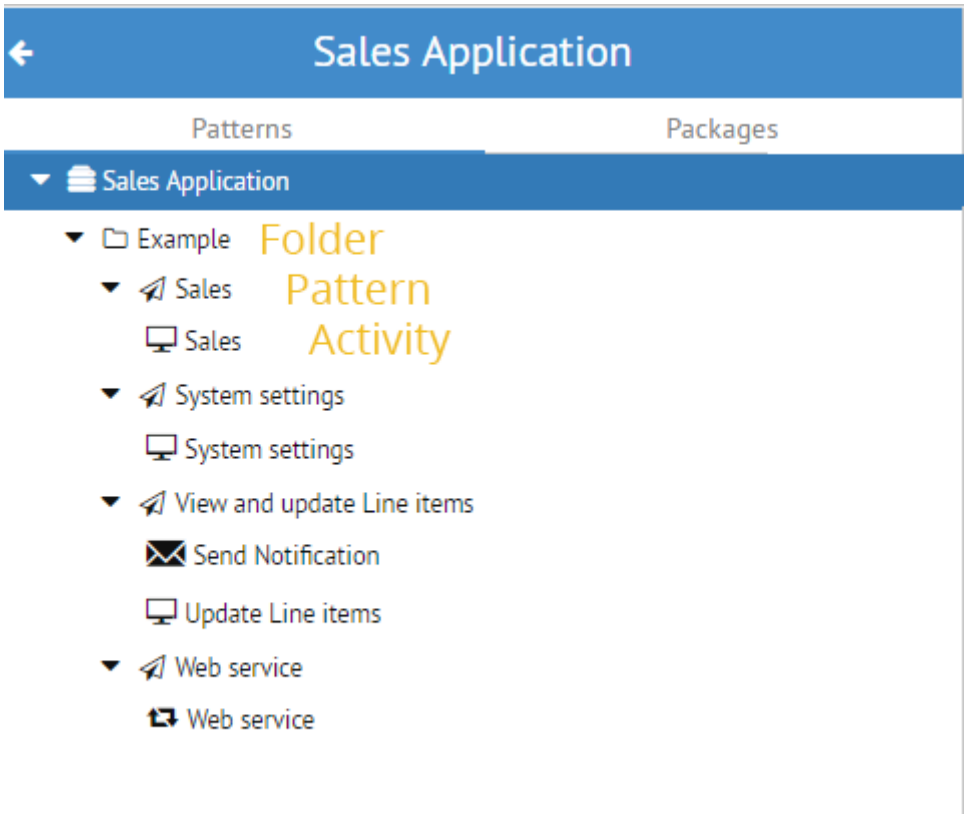
A **folder must exist** before a pattern can be created.

How to Create a Folder

New folders can be created by navigating to the patterns tab in the Organizational Explorer and selecting the '+' button in the bottom right hand corner.



The Sales Application will store all its patterns and activities in a single folder called **Example**.



Steps

1. In the Organizational Explorer, click the Patterns tab.
2. Click on the **Sales Application**.
3. Click the **plus** button and select **New Folder**.
4. Name the folder **Example** and click **Submit**.

Next, learn about Access Permissions.

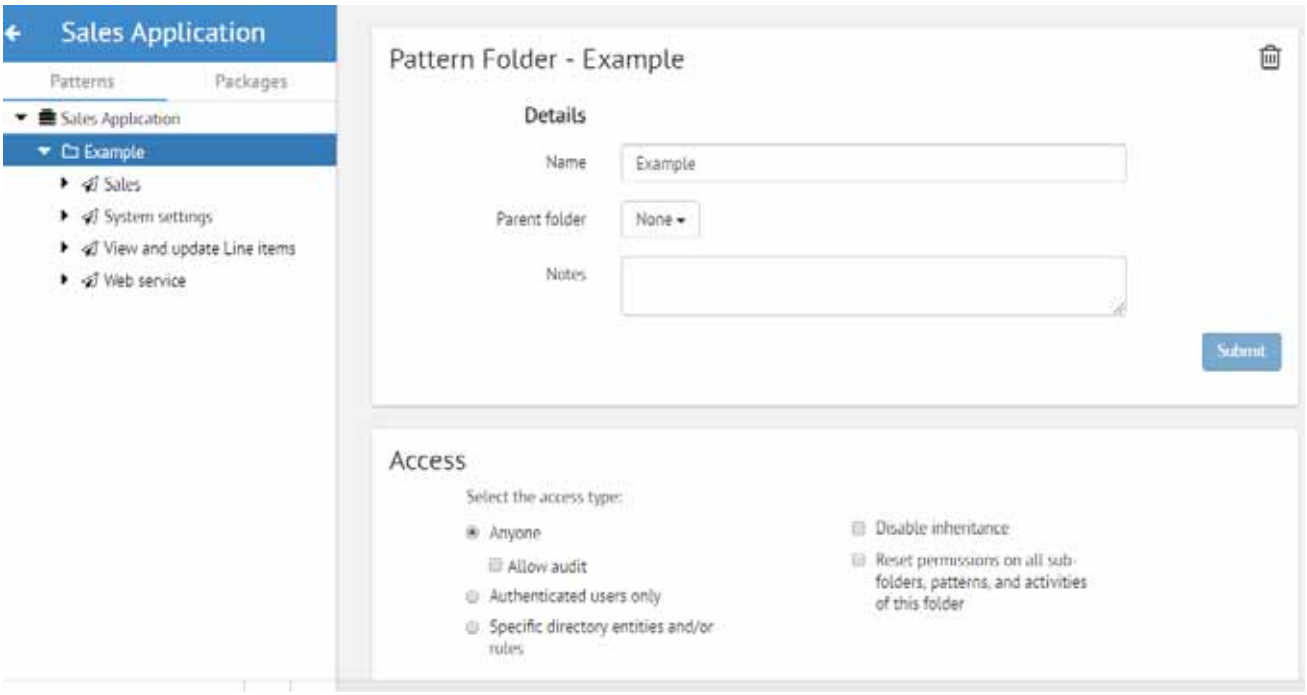
Controlling Access using Permissions

Permissions can be set to a specific pattern or a folder.

How to Define Access Permissions for a folder

In the Sales Application, permissions have NOT been set.

- Therefore, anyone who is an app user has access to the application patterns in the Example folder.



The following optional section describes how you can control access by enforcing permissions.

Enforcing Permissions (optional)

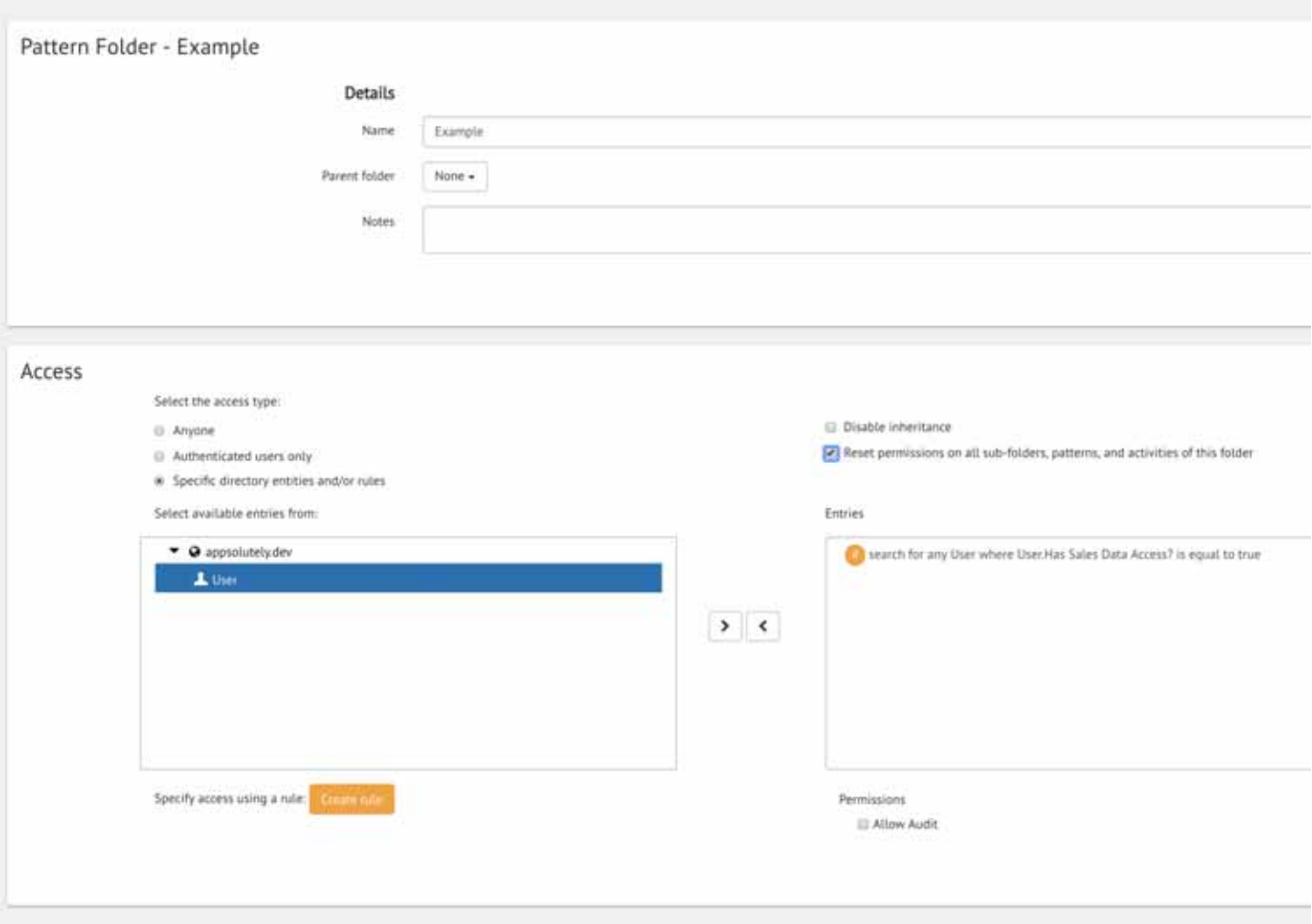
Where nested items exist, permissions can be enforced to all constituent items of the selected top-level item.

Permissions are managed by referencing the directory (user) packages and related attributes.

In the example below, only site administrators are allowed to access sales data.

Therefore, an attribute has been added to the user package **Has Sales Data Access?** from within the Directory.

The folder **Example** is now restricted to be available only to users carrying an affirmative value for **Has Sales Data Access**.



► **Note:** To force permissions to be applied to all dependent objects, ensure that 'Reset permissions on all sub-folders, patterns, and activities of this folder' is selected.

A suggested top-level folder structure for a simple app is:

- **Admin** patterns to which only administrators have access.
- **Logged In** patterns to which logged in users have access.
- **Public** patterns to which anyone has access.

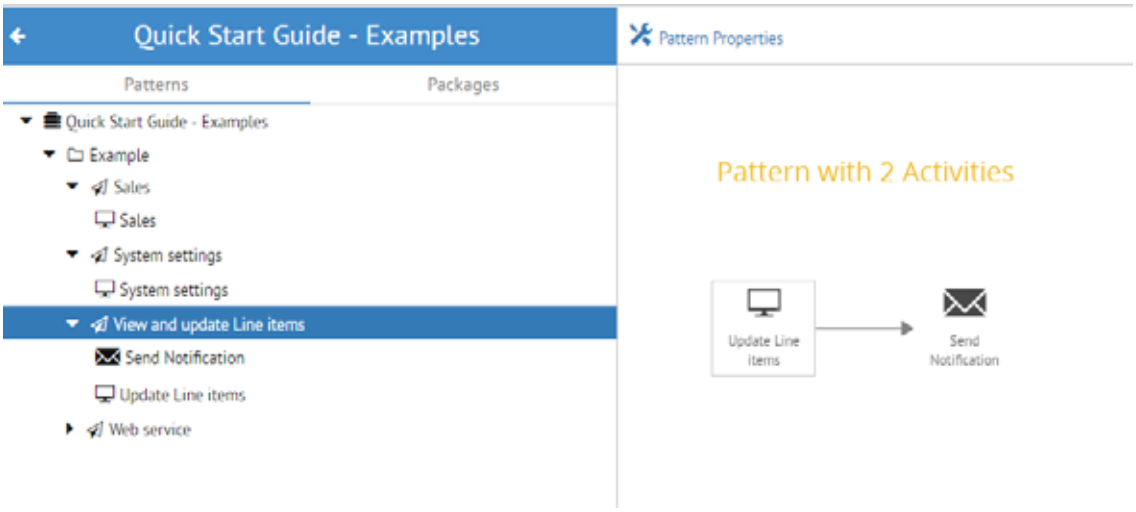
- **Engineer** patterns to which only engineers have access.

Next, we will learn about Patterns.

Patterns

A Pattern contains one or more Activities and is similar to a workflow or storyboard of actions within an application.

Patterns appear to the right of the Organizational Explorer - we refer to this area as the **Stage**.

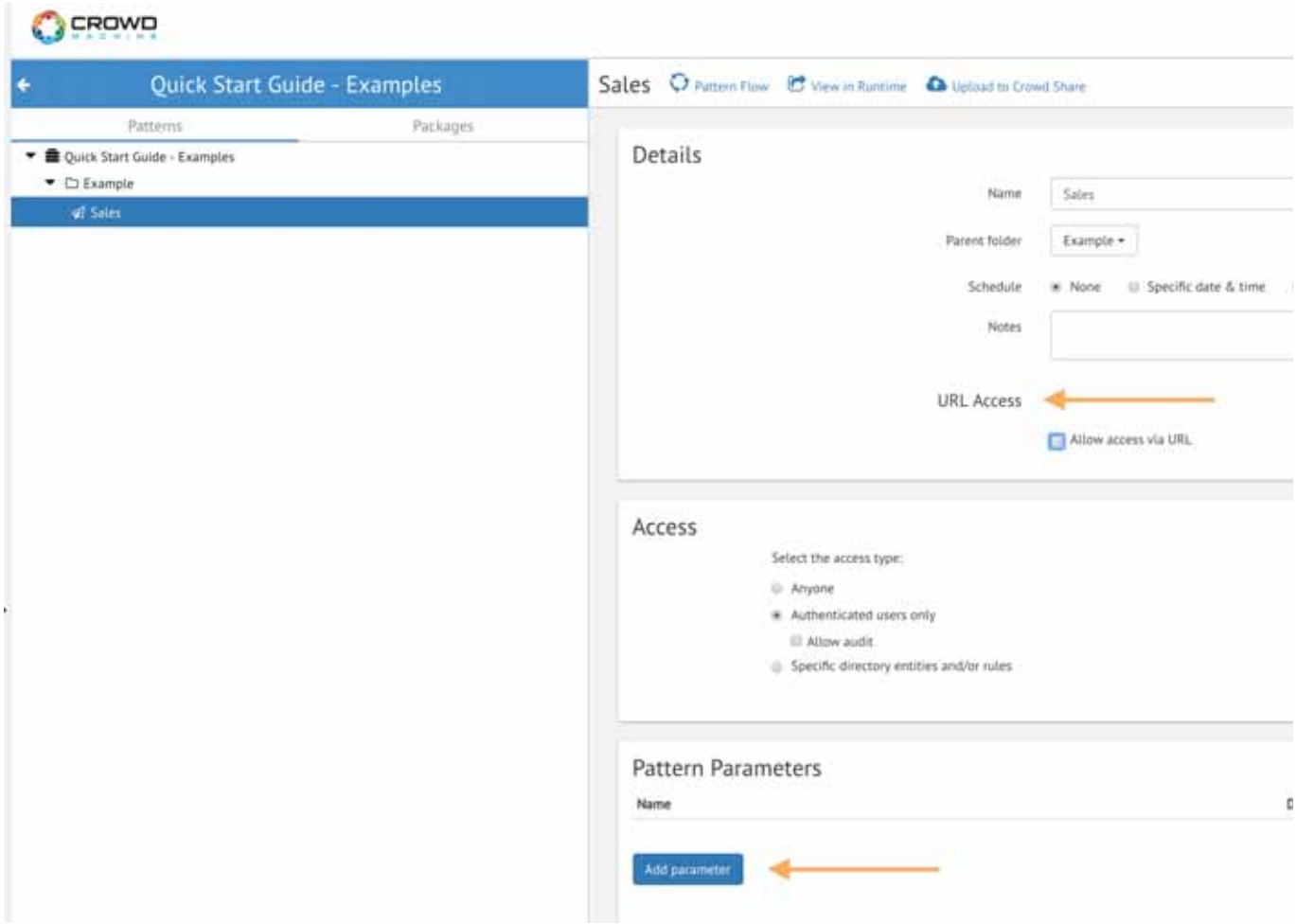


How to Create a Pattern

As Patterns act as containers for series of activities, they can be called by other Patterns (within the application) or by using a URL (from outside the application) and parameters can be passed into the pattern to be handled by the component activities.

The parameters can be passed to the pattern in a URL using **&key=value**, where the key is the pattern parameter name (URI encoded).

The pattern URL access and parameters are accessed on the pattern **Properties** page.



Follow these steps to create the Sales pattern.

Steps

1. In the Organizational Explorer, click the Patterns tab.
2. Click on the **Example** folder.
3. Click the **plus** button and select **New Pattern**.
4. Name the pattern **Sales** and click **Submit**.

Next, learn about Activities.

Activities

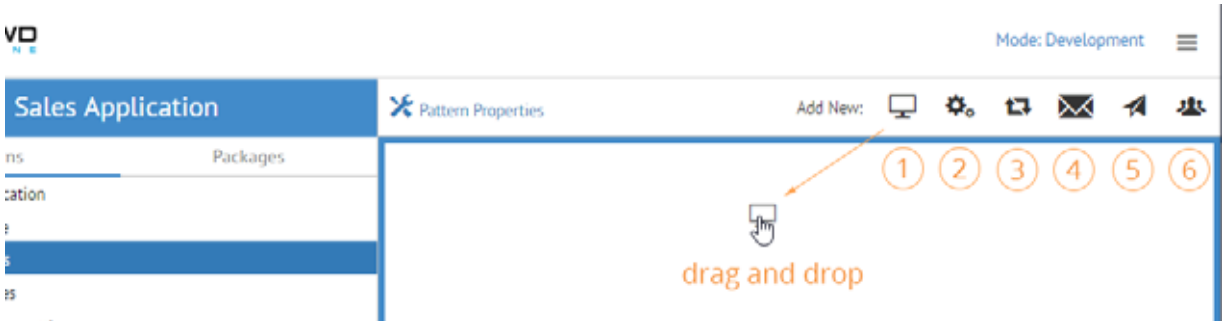
Activities are single operations that can be run independently in a pattern, or in sequence with other activities to form a **Pattern Flow**.

Because applications require different types of behavior to function, there are a number of different types of activities that can be used in a pattern to provide functionality.

Overview of Activity Types

Activity Types

Activities are added to the Pattern Flow via drag and drop.



There are six activity types available.

1. **User Input Activity:** a user interface to accept data input or to display read-only data.
2. **Automated Activity:** an activity that does not require user input to instruct it to run.
3. **Integration Activity:** an activity that can both send and receive data via an API.
4. **Email Activity:** used to send emails. Can be sent from Crowd Machine server or configured to relay to your email server via SMTP.
5. **Launch Pattern:** used to launch another pattern.
6. **Meeting Activity:** similar to an email, but with an associated calendar (.ics).

All activity types (except Launch Pattern activities) require a **data definition** prior to allowing activity properties to be configured. This means that data packages need to be added to the activity, and then the **settings defined** for how the activity will act on the Package(s).

Steps

1. In the Organizational Explorer, select the **Sales** Pattern.
2. Drag and drop a **User Input (UI) Activity** onto the stage.
3. Name the UI Activity **Sales** and click **Submit**.

Next, learn how to define an Activity.

Define an Activity

We must first transfer a Package into an Activity and then define the Activity behavior.

Video: Transfer the Sales Package and Define the Sales Activity

Transfer a Package

You need to transfer a Package into an Activity before you can define the Activity's behavior.

How to Transfer a Package into an Activity

Define an Activity

Once a package is transfered into a package, we must define how an activity alters and retrieves package data.

How to Alter Activities and Retrieve Data

An activity has no behavior until it is defined in the **Activity Definition** page.

By defining an Activity we are configuring it to act on the Package (that we transferred in) to produce a type of behavior in the application.

Define the Sales Invoice Activity

In the Sales application, we will transfer the Sales Account package (which contains the linked Sales Invoice activity) into the Sales activity.

- We will remove the linked **Line Item** package as it is not required.

- We will then define the linked **Sales Invoice** activity so that our application can create, edit and delete invoices.

Steps

1. Follow the instructions in the video tutorial to:
    - Transfer the **Sales Account** package into the Sales activity.
    - Remove the **Line Item** (linked) package
    - Define the **Sales Invoice** activity settings

When you define the **Sales Invoice** activity, it will enable a user to **create**, **edit** and **delete** invoices.

Points to note

- The above steps enable simple package retrieval so that data can be restricted by using pattern parameters, literals or relationships.
- For more complex data retrieval queries, you can use the Rule Builder within the Activity Properties.
- Relationships can be selected or created on the fly from the definitions page. This is the place to configure relationships between Directory packages (Users) and other packages.

Next, learn about configuring Attributes.

Configuring Attributes

Attributes can also be configured when defining an activity.

How to Edit / Configure an Attribute at an Activity Level

An Attribute can be configured to behave in a certain way at the **user interface**.

In this example, the Quantity attribute is set to **editable** and **mandatory** so that a user CAN EDIT and MUST ENTER a quantity.

Next, learn about the Page Designer.

Page Designer

The Page Designer provides a visual representation of the Sales UI Activity on a design canvas.

Video: Designing the Sales Invoice Page Layout in Page Designer

How it works

Once an activity has been defined, the Crowd App Studio automatically creates a default set of data objects that display as buttons and other page objects.

We use the Page Designer to modify the behavior and style of these page object controls.

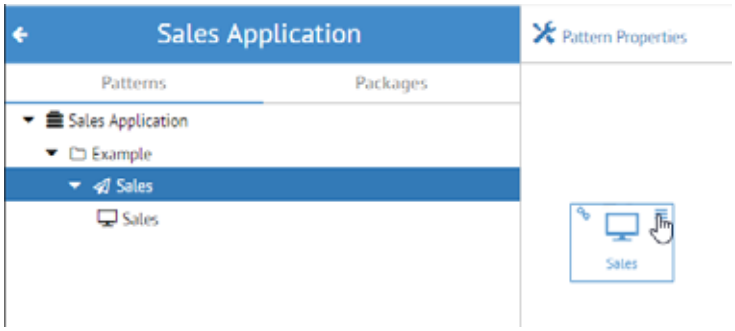
► **Note:** The Page Designer menu option becomes available only AFTER an Activity has been defined.

For more information on the Page Designer, refer to this guide:

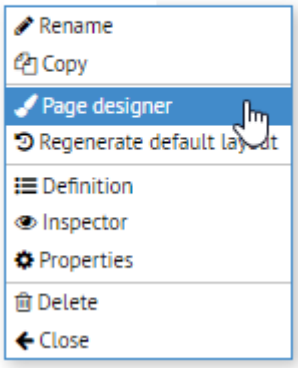
A Quick Guide to the Page Designer

Steps

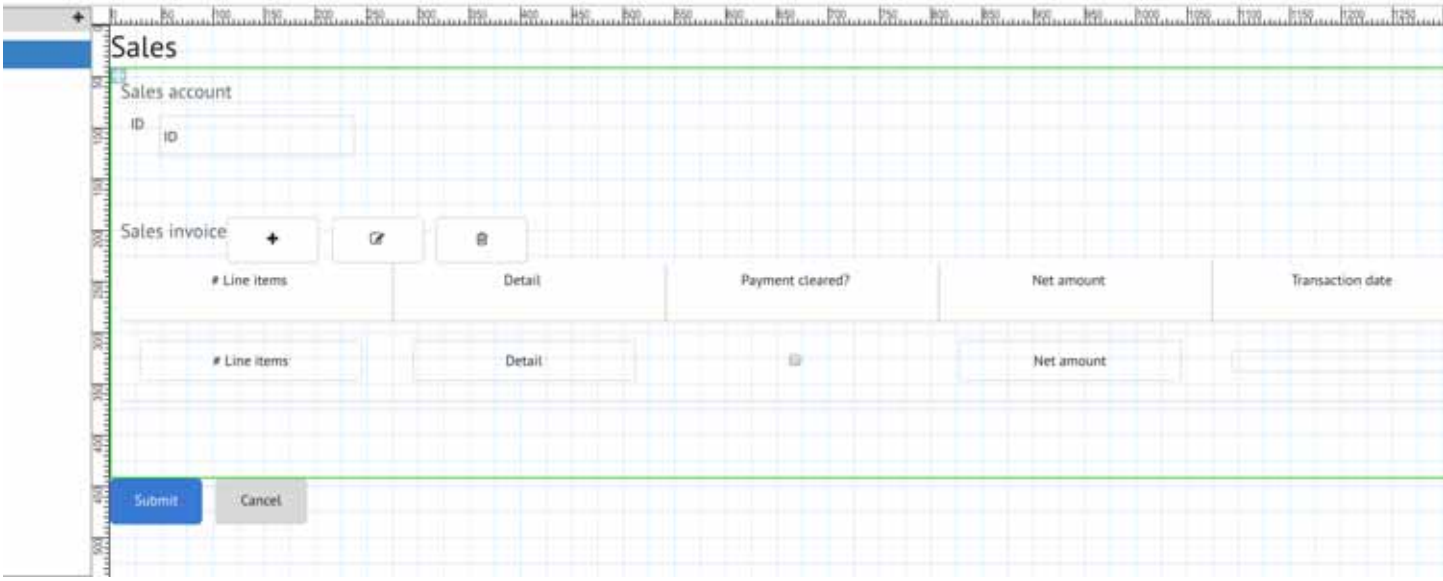
1. Ensure that the Sales pattern is selected so that the Sales activity is displayed on the stage.
2. Hover over the Sales activity and click the **menu icon** (3-bars).



3. Select the **Page Designer** option.



The Page Designer opens and displays the Sales Activity on the design canvas.



4. Follow the instructions in the video tutorial to modify the page layout for the Sales Activity.

Next, learn about using an Include panel.

Include Panel

In Page Designer, an Include Panel is a page object that allows us to dynamically display and contain data separate to our activity definition. For example, another pattern or website.

We will use an Include Panel to show the related Line item data.

Video: Add an Include Panel

Steps

1. Follow the instructions in the video tutorial to:
    - Add an Include Panel.
    - Remove any unwanted Page Objects.

Once you have finished designing in Page Designer, you will submit the changes and return to the Sales pattern to create a Looping Connector.

Page Objects

Page Objects are UI elements that be added and configured in addition to those automatically generated by Crowd App Studio for the activity packages.

When adding page objects, the best approach is to:

- Identify the object container that you want to contain the new object, such as the Page Body itself.
- Select the object container by clicking on it on the design canvas, or by locating in the Layout panel.
- Once the container is selected, find the Include Panel in page objects and add it to the page.

Next, learn to create a Looping Connector.

Looping Connector

A Looping Connector is a connection that forms a return-path loop in an activity.

A Looping Connector causes an activity to reload in a pattern, enabling a user to input data, submit the data, and then input data again.

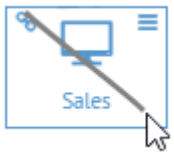
Video: Create a Looping Connector and Mark First

Follow these steps to create a looping connector on the Sales activity.

Steps

1. In the Organizational Explorer, select the **Sales** pattern.
   The Sales activity displays on the stage.

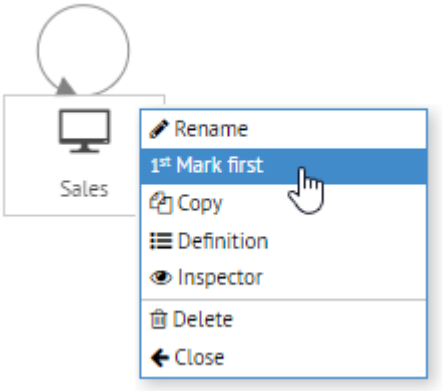
2. Click on the **chain connector icon** (in the top left of the activity) and drag the connector to the bottom right hand corner of the activity icon and release.



   The New Connector pop-up displays.

3. Click **Submit**.
   A looped connector displays above the activity.

4. Now, click on the menu icon (in the top left of the activity) and select **1st Mark first**.



Setting an activity as 1st Mark first ensures that it **runs first** in the pattern flow.

Even though there is only one activity in the pattern flow, it is necessary to mark this activity first because we added a looping connector.

Next, we will view the pattern at runtime.

View a Pattern in Runtime

Once you have designed an application pattern in the Crowd App Studio, you can view it in runtime through a web browser.

Video: View the Sales Pattern in Runtime

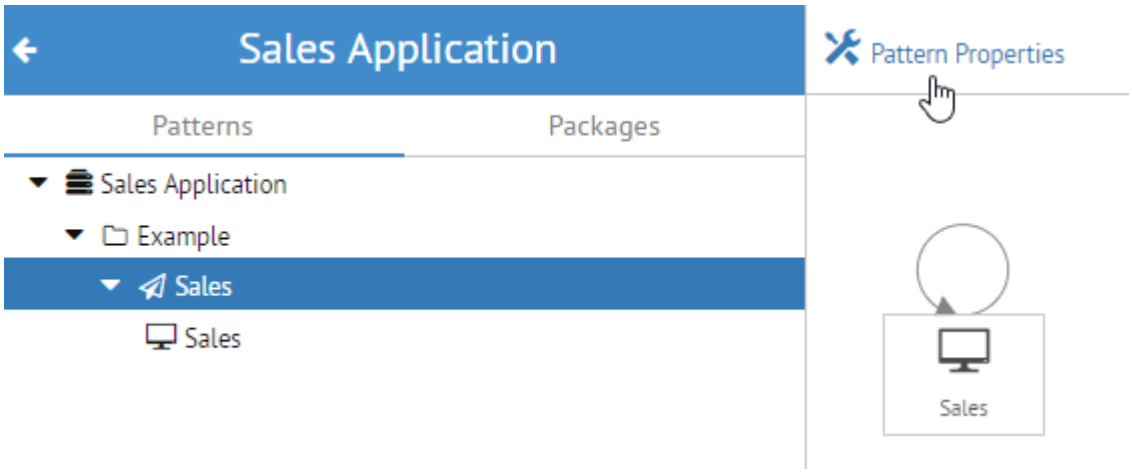
Runtime describes the environment in which a pattern is run and activities are executed.

When a UI activity executes, the user interface is displayed in a web browser.

Steps

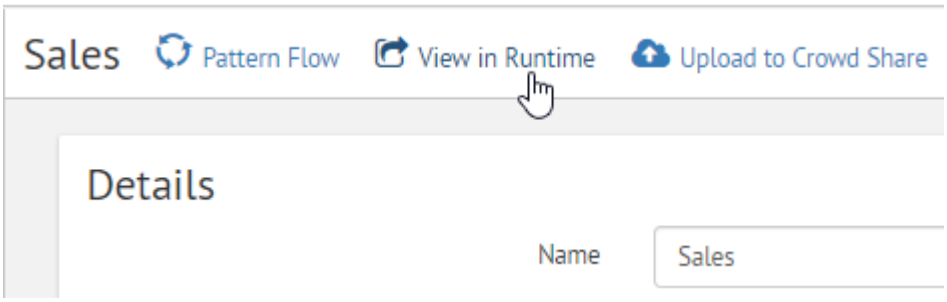
1. In the Organizational Explorer, select the **Sales** pattern.
   The Sales activity displays on the stage.

2. Click the **Pattern Properties** button.

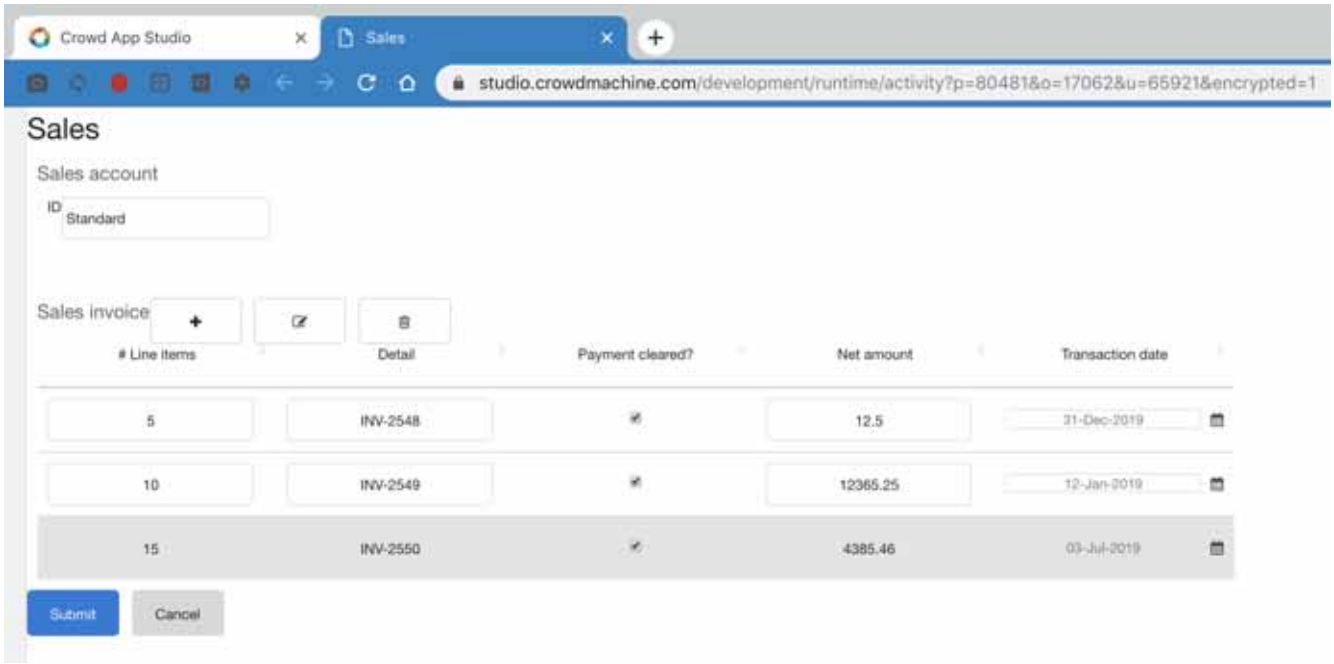


   The Pattern Details page displays.

3. Click **View in Runtime**.



A new browser window displays the Sales UI activity in runtime.



The Sales UI activity displays the Sales account and Sales invoice packages.

We can use the buttons to add, edit and delete data rows.

Next, learn about relationships in action.

Relationships in Action

We need a way to calculate and update the Line items for a given Sales invoice.

Video: Updating the Sales Invoice Line Items

The Sales Account and Line Items Relationship

Our Sales invoice package has two relationships.

There are many Sales invoices for a given Sales account, and each Sales invoice includes many Line items.

For the purposes of this exercise, we will ignore the **Sales account ---< Sales invoice** relationship, and focus instead on the **Sales account ---< Line items** relationship.

View and Update Line Items Pattern

We will need to create a new pattern called **View and Update Line Items** containing two activities.

- A UI activity called **Update Line Items** to update the line items.
- Later in this guide, we will create an Email activity that sends the updated information to a user.

Update Line Items Activity

We will create a UI activity called **Update Line Items** to update the line items.

When defining the activity, we can copy the Sales Invoice package into the activity and select the checkbox option **Linked packages** - this will also copy over any **immediate relationships** (Line items).

Steps

1. In the Organization Explorer, click on the **Example** folder.
2. Create a new pattern and name it **View and Update Line Items**.
3. Follow the instructions in the video tutorial to add the **Update Line Items Activity** and to configure the pattern.

How to Calculate Totals for the # Line Items and Net Amount

In the video tutorial, you will see that we want to calculate the totals **# Line items** and **Net amount**.

A valid approach would be to calculate these totals and update the Sales invoice attributes each time the activity is run.

We will adopt part of this approach, in that we will calculate values to be included in the subsequent Email activity.

However, later in this guide, we will introduce an alternative method to calculate these values when required (instead of on each execution of the activity).

In order to carry the calculated values from our **Update Line items** activity into the Email activity, we will use a Temporary Attributes Package.

Next, learn about Temporary Attributes Packages.

Temporary Attributes Package

When working with complex activities, in which data exists that is not necessarily part of defined packages, the use of a temporary attributes package is recommended.

Video: Configure the Temporary Attributes Package

This is effectively a table of attribute placeholders, which can then be used temporarily within activities, or throughout patterns. In other words, a Temporary Attributes Package is created to hold temporary data that will soon be discarded.

In the Sales Application demonstration app, you will find a basic temporary attributes package named **[T]**, configured as follows:

Attribute Name	Data Type
Text 1	Text
Number 1	Number (Decimal)
Integer 1	Number (Non-Decimal)

Use in Email Notification

Later in the guide, will create a row within the Temporary Attributes Package, calculate the total Line Item values/Net Amount and reference the row within the Email activity that we create, and then discard the data.

- To achieve this, we will transfer the [T] package into an **Email Notification** activity and rename the attributes to **# Line items** and **Net Amount**.

If it was the case that we did not need the data in any other pattern activity, then no definition would be required (this is demonstrated later in the guide).

Steps

1. Configure the **[T]** Temporary Package Attributes as shown in the video tutorial.

   Next, learn about rules.

Rules

Rules can be compared to spreadsheet formulae, database statements, and code logic.

In this exercise, we will look at one particular type of rule called a **UI Action Rule**.

Video: Creating UI Action Rules

For general information on rules, read this knowledge article:

Rules Overview

Rule Scope

When creating rules, application designers must understand Rule scope. Rule scope determines the region of an activity in which a rule can execute.

For example, whether a rule executes on a **currently selected package row**, OR **on the hierarchy of data from the Top Level package** in an activity (a Top Level package resides as the topmost package in an activity definition).

Scope can be modified for each statement, and be explicitly stated where necessary.

Read this knowledge article to learn more about Rule scope:

Rule Scope

About UI Actions

First, we'll need to create a new UI Action.

UI Actions are generally attached to a page object, so when that page object is clicked, the UI Action will run.

An example is an Update button in the Sales Application. When the Update button is clicked, the UI action attached to this button is run, a rule is executed to calculate line totals as well as the count of line items.

The totals for the Sales invoice are then displayed.

Here is a list of common UI Action Types - in this exercise we will use a UI Action Rule.

UI Action Type	What it does
Rule	An expression editor that enables package data and system objects to be configured dynamically.
Activity Action	Predefined actions that relate to the activity instance.
Package Action	Predefined actions that relate to activity instance packages and relationships.
Launch Pattern	Identical to a Launch Pattern activity, except run from an activity and not a pattern.
Other System Action	Predefined actions that relate to objects not included under activity/form/package actions.

Rule Statements

Rules consist of Statements, individual instructions that operate on application objects.

We'll use a few commonly used rule statements to add logic to our activity.

This logic will calculate the updated total of the individual Line items, as well as the count of Line items, and assign these values to the parent Sales invoice.

- We will use an **Assign** statement and a **For each loop**.

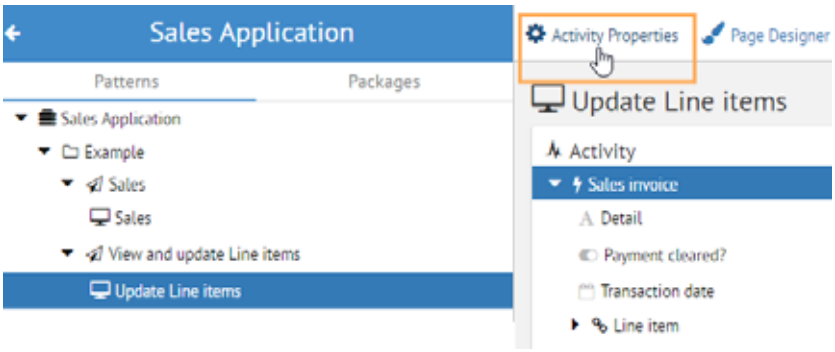
You can learn more about rule statements in this guide

A Guide to the Rule Builder

Follow these steps to create a Rule UI Action.

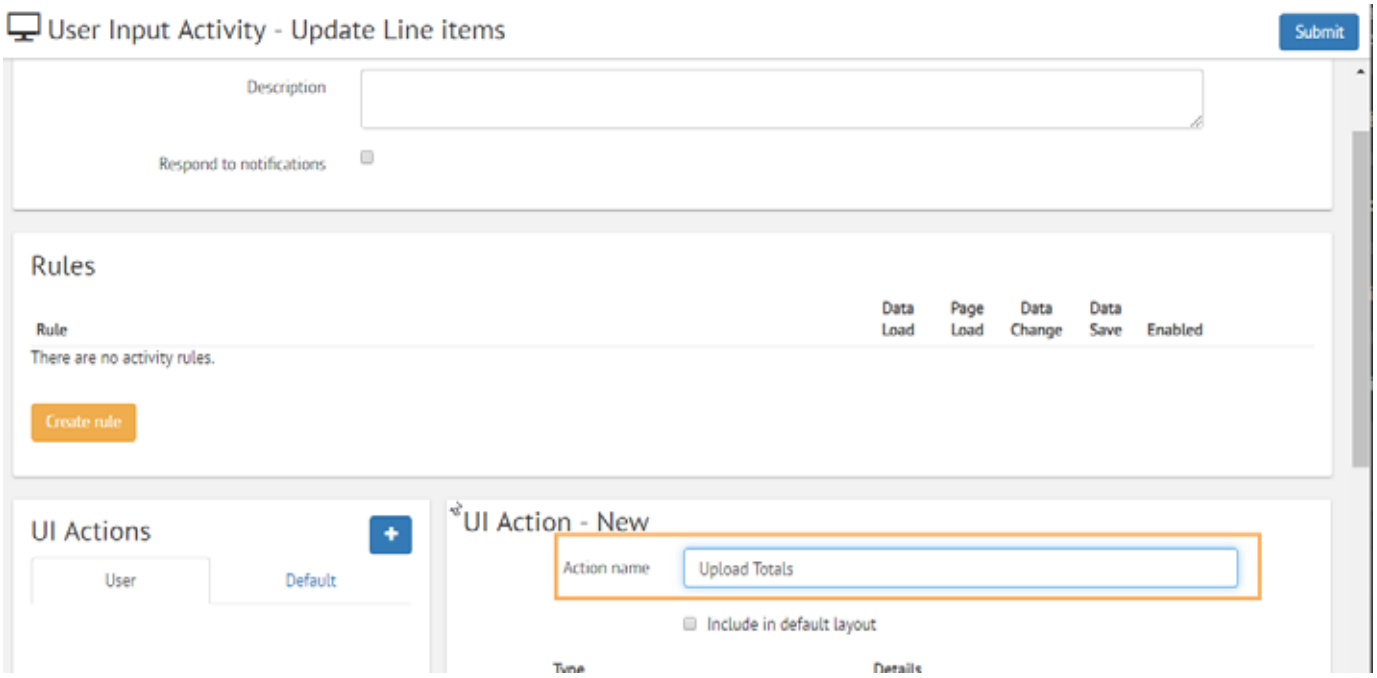
Steps

1. In the Organizational Explorer, click on the **View and update Line Items** pattern.
2. Click the **Update Line Items** Activity.
3. Click **Activity Properties**.



The Activity Properties page displays.

4. Scroll down to the UI Action panel and locate **UI Action - New**.
5. In Action name, type **Update Totals**.



6. Follow the instructions in the video tutorial to:

   - Add the **Assign** rule (this will set the Line Item count)
   - Add a **For each loop** (this will calculate the line item as ‘unit price’ multiplied by ‘quantity’).

7. Follow the instructions in the video tutorial to add an Activity Action to the UI activity which will submit the activity to ensure that data is saved.

This demonstrates that a UI Action can have multiple tasks and rules configured for it.

8. Follow the instructions in the video tutorial to attach the UI Action to the Update button (from within Page Designer).

   - Once this is complete, we will have the basic interface to handle the management of Line items, and logic in place to calculate updated totals for the Email activity.

Next, learn about Update Line Items in Runtime.

Update Line Items in Runtime

This time when we run the **View and Update Line Items** pattern in runtime, we'll be prompted to enter a value.

Video: Update the Line Items in Runtime

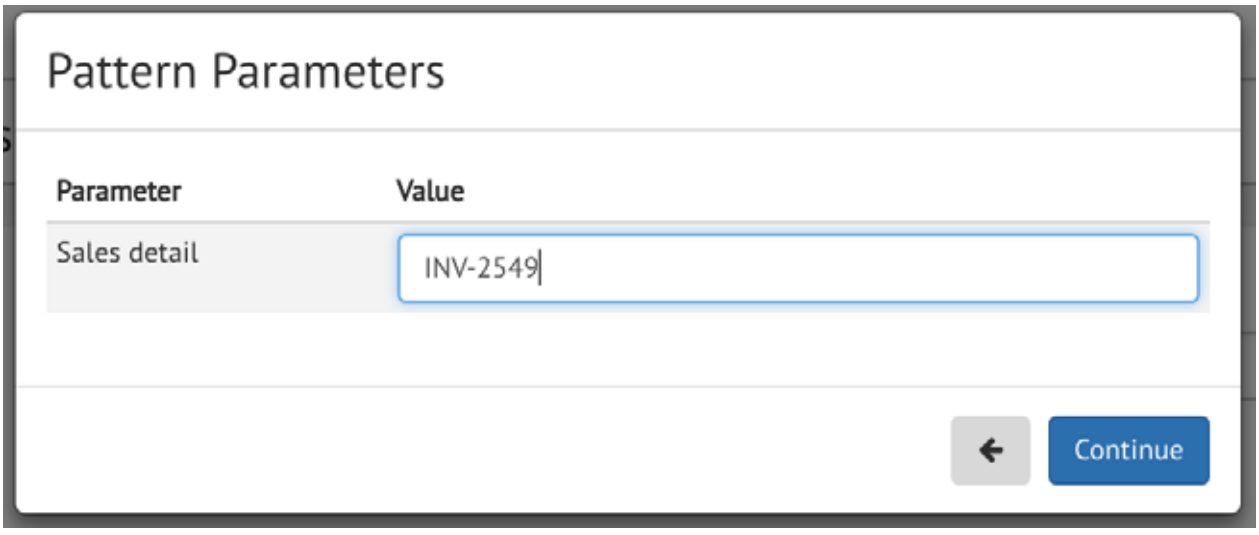
We are prompted to enter a value because we have a Pattern Parameter that is the **unique invoice ID** included within the **Sales detail** attribute.

Follow the steps below to update the line items for a selected invoice in runtime.

Steps

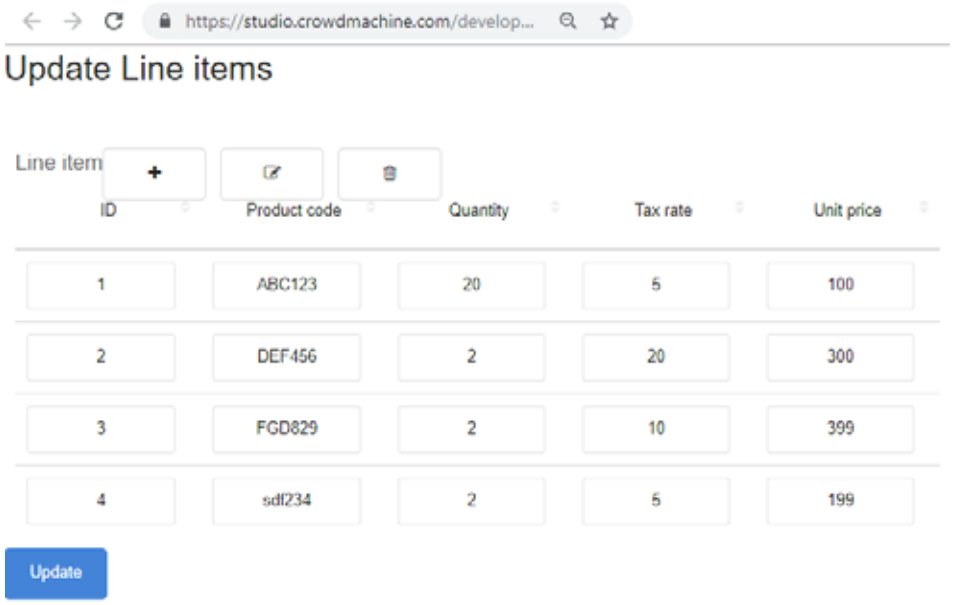
1. Navigate to **View and Update Line Items** Pattern Properties and click on **View in Runtime**.
   The Pattern Parameters pop-up displays.

2. We will enter **INV-2549** - the second value in the example invoice table.



3. Click **Continue**.

   The Update Line items page displays.



4. We can now update the Line items for the selected Sales invoice.

   We never set any for the example, and the values currently stored as totals on the Sales invoice are not yet based on Line item data.

   Note that the pattern parameter is visible in the page URL.

5. Once Update button has been clicked, reload the page and the entered data will appear.

   Next, learn about Email Activities.

Email Activities

We will add some functionality to the **View and update Line items** pattern by adding an Email activity.

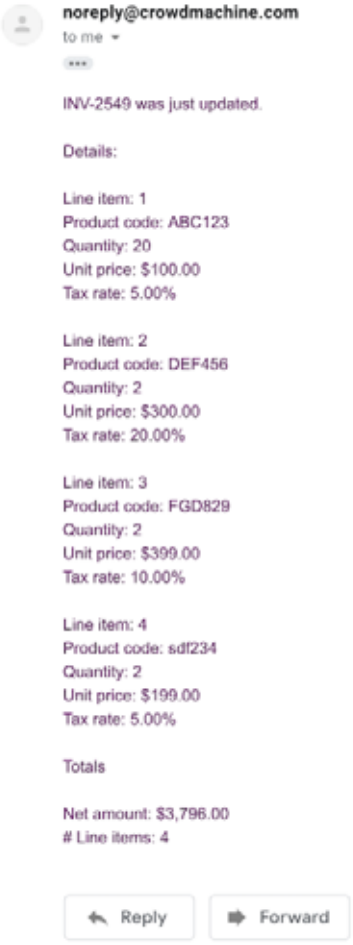
Video: Add an Email Activity

Email Notification

This Email activity will send the user an email notification each time the user updates the line items in a Sales invoice.

The email body will contain:

- A message to indicate that the invoice number was updated.
- A list of the line item details.
- Totals for the net amount and # line items.



About Email Activities

The Crowd App Studio has a specific activity type called an Email activity for sending emails.

This enables you to create emails dynamically within the app, and then either relay to your own email server via SMTP for distribution, or to use the default **noreply@crowdmachine.com**.

Required Packages

The following packages are required to create and send an email.

- Systems Settings (new package) - contains the **Default Email** attribute.
- Email (new package) - contains **ID/To/From/Subject/Body/Timestamp** attributes.
- User (new user package) contains the **E-mail Address** attribute.
- Sales Invoice (existing package) - contains **Sales Invoice** and **Line Item** attributes.
- [T] (existing temporary package) - rename the attributes to **Net amount** and **# Line items**.

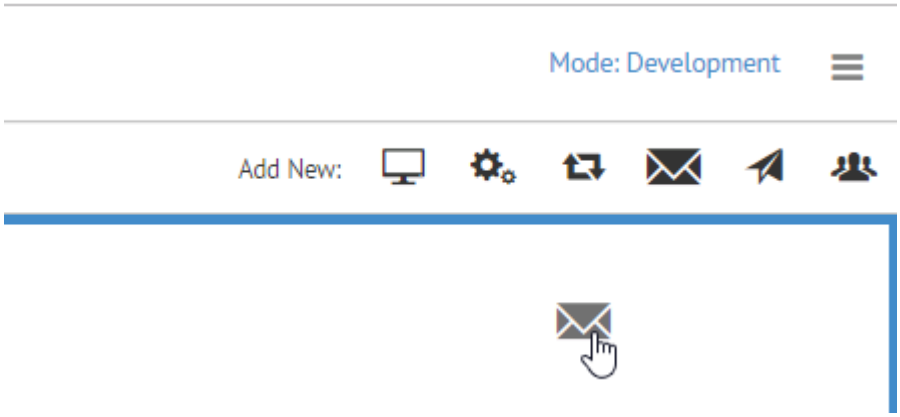
Required Patterns

The following patterns and activities are required to create and send an email.

- **System Settings** (new pattern) - contains the System Settings UI activity (new activity).
- **View and Update Line items** (existing pattern) - contains Update Line items (existing activity) and Send Notification (new activity).

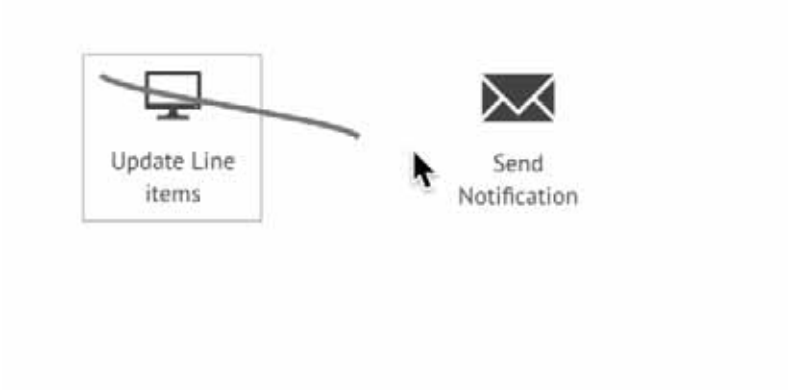
Steps

1. Open the **View and update Line items** pattern.
2. Drag and drop an **Email activity** (envelope icon) onto the stage.

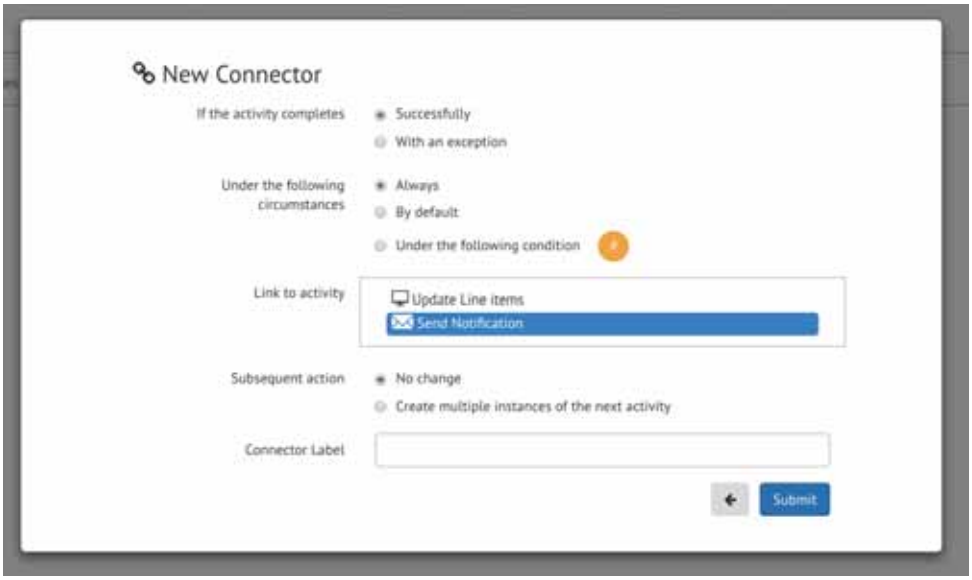


3. Name the email activity **Send Notification**.

4. Drag the connector icon between the Update Line items activity and Send Notification.



5. When the New Connector pop-up displays click **Submit**.



► **Note:** For more complex pattern flows, conditional logic can be included in connectors, to determine how a pattern should be run. In the case of this example, no logic is required, and so the standard connector settings can be used.

6. Follow the instructions in the video tutorial to:

   - Create a Systems Setting package.
   - Create a System Settings pattern and Systems Setting UI activity
   - Create an Email package.
   - Create a User package
   - Define the Email activity.
   - Define the Send Notification activity and add a retrieval rule.
   - Write a rule that populates the email content and formats HTML.
   - Assign the email attribute values to the email.
   - Test that the email notification is sent when an invoice line item is updated.

Next, learn about SMTP Settings

SMTP Settings

You may want to configure and use your own **SMPT server** for managing email distribution.

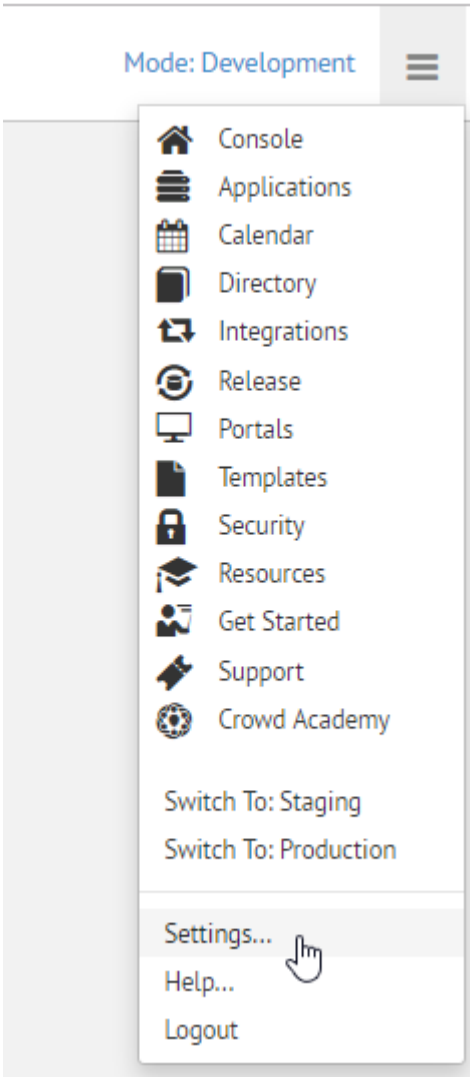
In the Email activity exercise, for testing purposes, we used the Crowd Machine default email server (noreply@crowdmachine.com).

The SMTP settings were not required.

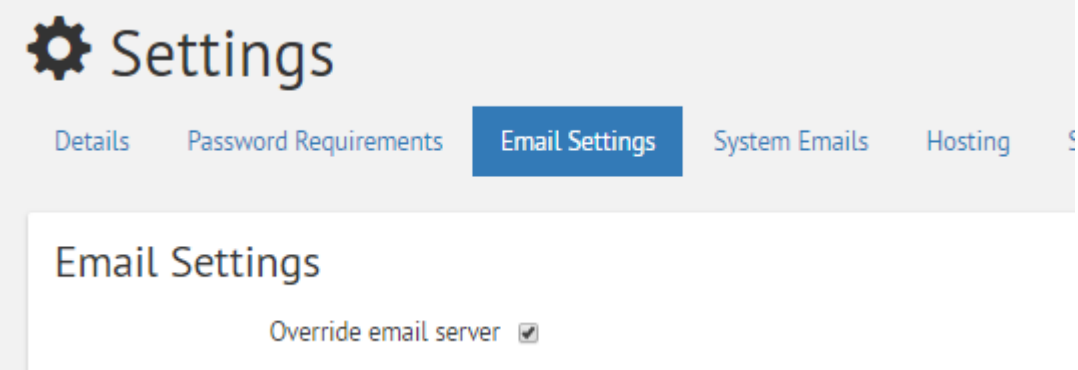
If you are using your own SMPT server, follow these steps to access SMTP settings.

Steps

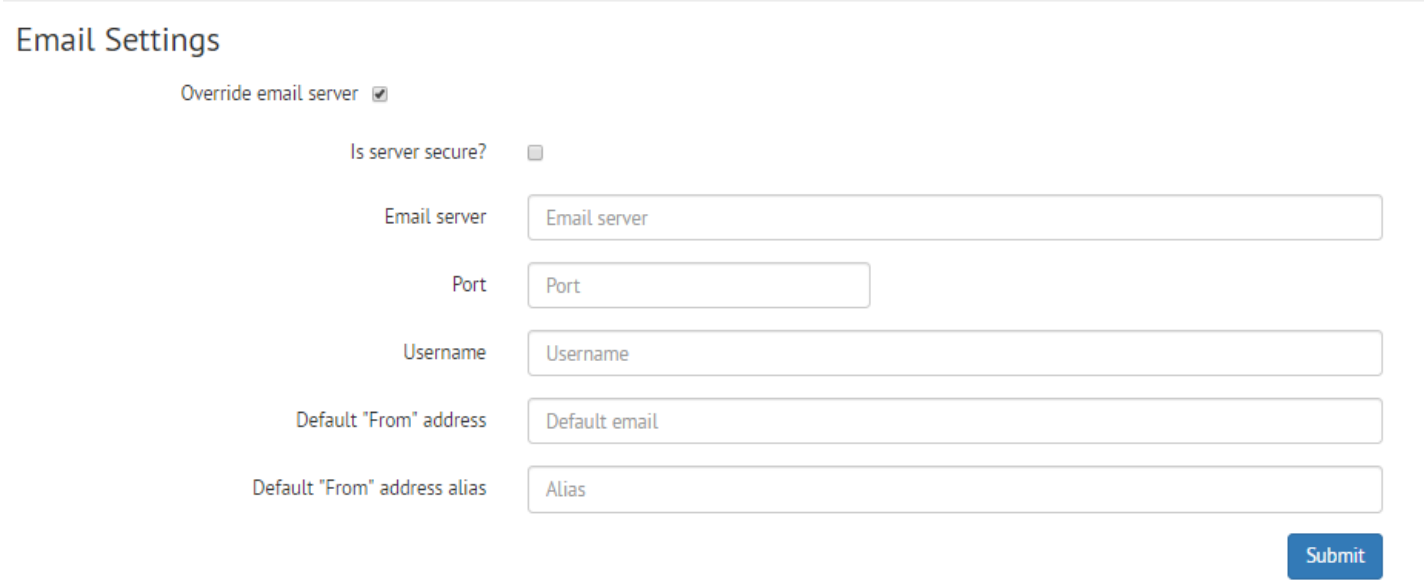
1. Click on the main navigation menu icon(3 bars) and select **Settings**.



2. Click **Email Settings** and select the checkbox **Override email server**.



The Email Settings panel displays.



3. Complete the email settings fields and click **Submit.**

Next, learn about Populating Include Panels.

Populating Include Panels

Now we need to create the Sales view for the application by combining the two patterns **Sales** and **View and update line items**.

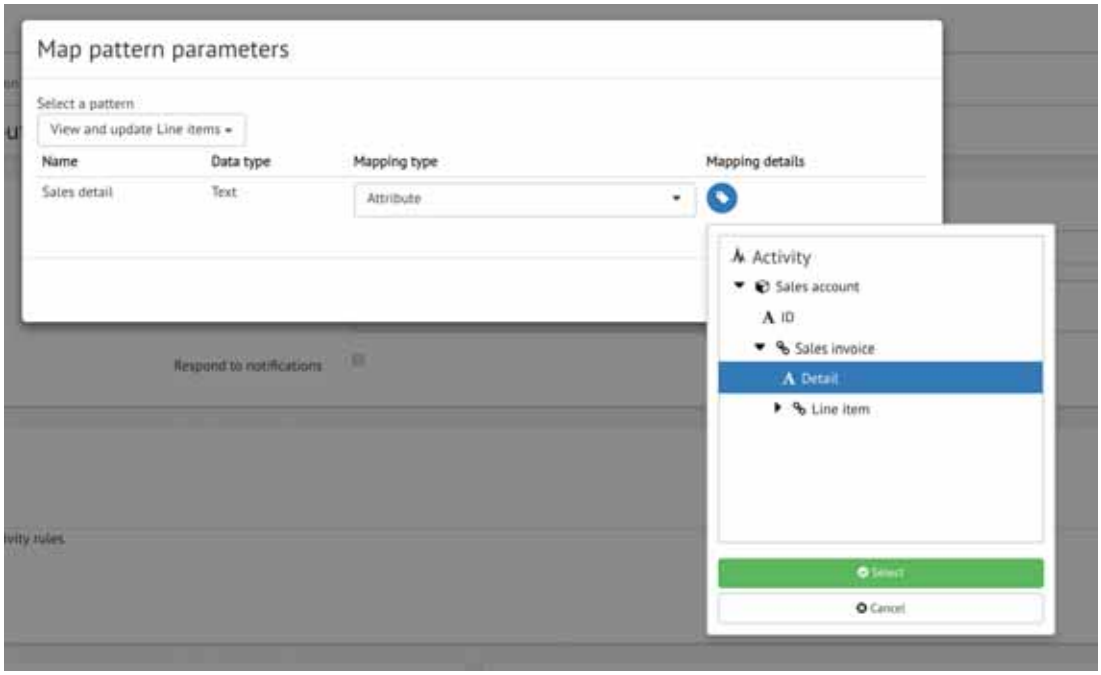
In runtime, when a user clicks on a Sales invoice row, the line items will display in a panel.

Video: Use a UI Action to populate an Include Panel

In an earlier exercise, we created the Include Panel in the Sales activity main page using Page Designer.

- As we want the include panel to be populated with Line items when a user clicks on a Sales invoice row, we need a UI Action to attach to the table rows.

To achieve this, we will create a UI Action called Show Line Items and we will map the parameters from the **View and update line items** pattern by selecting **Detail**.



Follow these steps to set up the Include panel to be populated with line items.

Steps

1. Open the Sales UI **Activity Properties** page.
2. Create a UI Action called **Show Line Items**.
3. Follow the instructions in the video tutorial to set up the UI Action for the include panel and to attach the UI action using Page Designer.

Next, learn about Reporting Packages.

Reporting Packages

Reporting packages are special package types that allow for package data to be queried without loading into the runtime. It is similar to building a database query for reporting against datasets.

Reporting packages are efficient for application performance when dealing with large datasets.

Video: Add a Reporting Package to Query Data

Use a Reporting Package to Retrieve Data (temporarily)

For this exercise, our goal is to show how a Reporting package can be used to temporarily retrieve the Sales Invoice total amounts **# Line Items** and **Net Amount** for reporting purposes.

The reported data is never saved, it is recalculated each time it is needed.

Replace Sales Invoice Attributes with Temporary Reporting Package [R]

To achieve our goal, we're going to replace the Sales Invoice package attributes **# Line items** and **Net amount** with our Reporting package as a temporary package named **[R]**.

The temporary reporting package [R] will contain the attributes # Line items and **Net amount** and will be transferred into each Sales Invoice package.

We will then configure the Reporting package data definition and build a query to retrieve the total amounts.

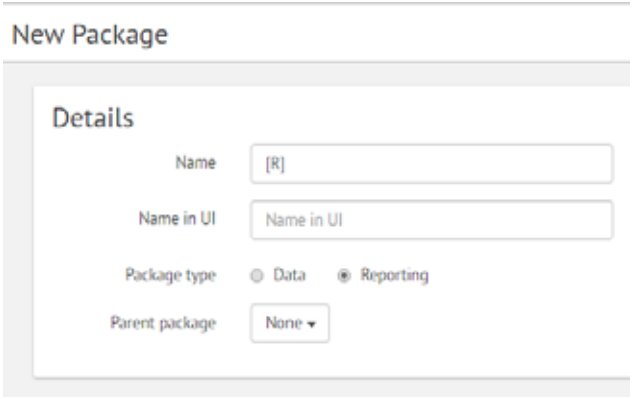
Read this knowledge article to learn about Reporting packages:

Reporting Packages

Follow these steps to create a Reporting package.

Steps

1. In the Organizational Explorer, create a new package and name it **[R]**.
2. Under Package type, select **Reporting**.



3. Create two attributes with attribute types as shown.

Attribute	Attribute Type
Number 1	Number (Decimal)
Integer 1	Number (Non-Decimal)

4. Click **Submit**.
5. Follow the instructions in the video tutorial to configure and test the Reporting package.

Next, learn about Notifications.

Notifications

When the **Update Line items** pattern is updated, the relevant attributes will be updated. However, we still have no logic that will refresh the **Sales** pattern data and therefore the total values shown in Runtime.

Video: Use a Notification to refresh Sales Pattern data

To communicate between patterns we can make use of notifications.

Required for the notification

Before implementing the notification, we need the following:

- A **Refresh Sales** UI Action in the Sales **Activity Properties**.
- A rule that will re-populate Sales Invoice Reporting package data

Follow these steps to implement notifications.

Steps

1. Follow the video instructions to:
    - Create a new UI Action named **Refresh Sales** in the Sales Activity Properties.
    - Write a UI Action rule to re-populate Reporting package data.
    - Set up a Notification handler in the Sales Activity Properties page.
    - Add new logic to the existing **Update Totals** UI Action.
    - Test the notification by running the Sales pattern.
    - Use a temporary package to reload the include panel.

Next, learn about Integration Activities.

Integration Activities

An Integration activity is an activity type that integrates with external web services and APIs.

Video: Add an Integration Activity to Update a Google Sheet

Integration activities allow application designers to perform outbound integrations to retrieve and update data stored in external systems, or inbound integrations to allow external systems to retrieve and update data in Crowd Machine.

Learn more about Crowd Machine integration here:

Integration Activities Overview

Google Sheets Endpoint

In this exercise, we will create an integration endpoint that a Google Sheets formula can call. We'll create a public API endpoint for our application, which acts as a web service, and Google Sheets can use it to query the data in our packages. The data in the Google sheet is updated to display the invoice data from the Sales Application.

Google Sheets has a formula, **=IMPORTXML()**, which allows data to be retrieved from an endpoint in XML format for display within a spreadsheet.

You can learn more about the IMPORTXML formula here:

IMPORTXML guide

XML

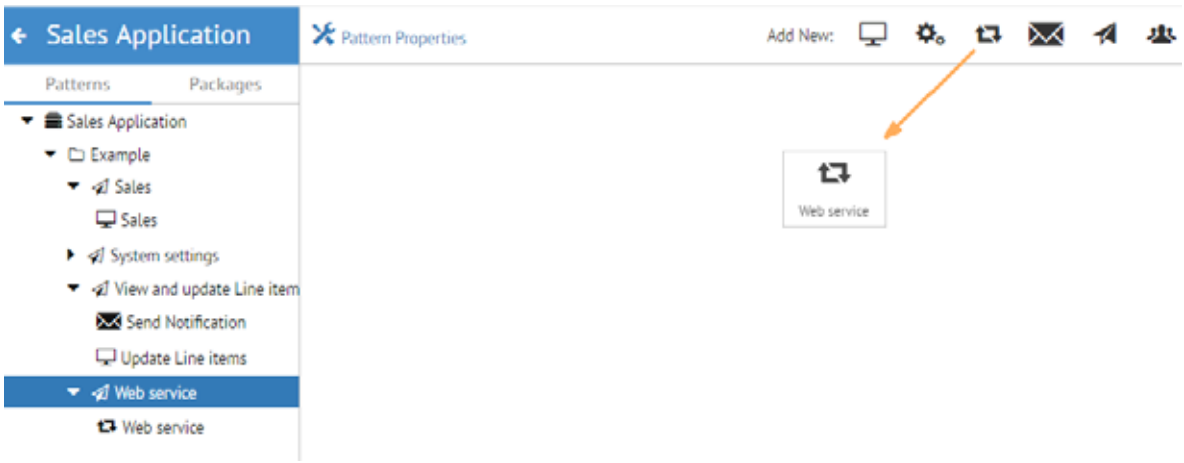
XML is a data structure which allows custom values to be nested, and so serves as an interface between our package structures and the spreadsheet tables to display.

Integration Activities use an XML Mapper to map XML data:

How to Edit XML Elements in XML Mapper

Web Service Integration Activity

To implement this functionality we will create a new Pattern called Web service that contains an Integration activity named **Web Service**.



We'll use the Web Service Integration activity to create an API endpoint (which Google Sheets will query).

Configure the API

An integration activity can interact with, and perform as, a RESTful API.

In our exercise, we need the activity to perform as an API, and so we need to configure the **Receive and respond** options.

First, we need to specify the Receive method. This includes a list of operations, all of which are standard HTTP requests.

To understand more about HTTP requests, read this guide.

HTTP Request Methods

Receive Method GET

As we are receiving an instruction to serve data, the type of request to select is GET. The Google sheet is making a request to GET data.

The Integration activity now knows what type of operation is being undertaken and as such, will have to respond with the requested data.

Response

The Response body type is where the type can be specified - different body types require different configuration. Google Sheets requires an XML and so we select XML.

Request URI

The Request URI is the endpoint which will be included as part of the **=IMPORTXML()** formula.

Within **Map Integration Message** we can configure the payloads - data exchanged by the Integration activity.

Refer to the video tutorial to see more detail on Mapping.

Steps

Follow the video tutorial instructions to implement the following:

- Create a new pattern named Web Service.
- Create an Integration activity named Web Service.
- Transfer a temporary attribute to hold an input value (Sales Invoice Detail)
- Transfer Sales Invoice and Line Item packages into the Web Service activity

- Define the Web Service activity
- Configure the Integration Activity Message parameters
- Test the integration

Next, learn about portals.

Portals

Portals provide access to applications from outside of the Crowd App Studio.

Video: Create a Portal

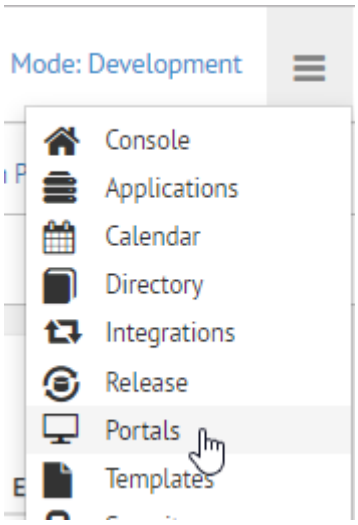
Portals point to a particular Pattern and are represented with a unique URL endpoint. As with folders and patterns, Portals can also have access restricted conditionally.

How to Create a Portal

In this exercise, we'll create a Portal and will point it to the Sales pattern, allowing anyone access to the application.

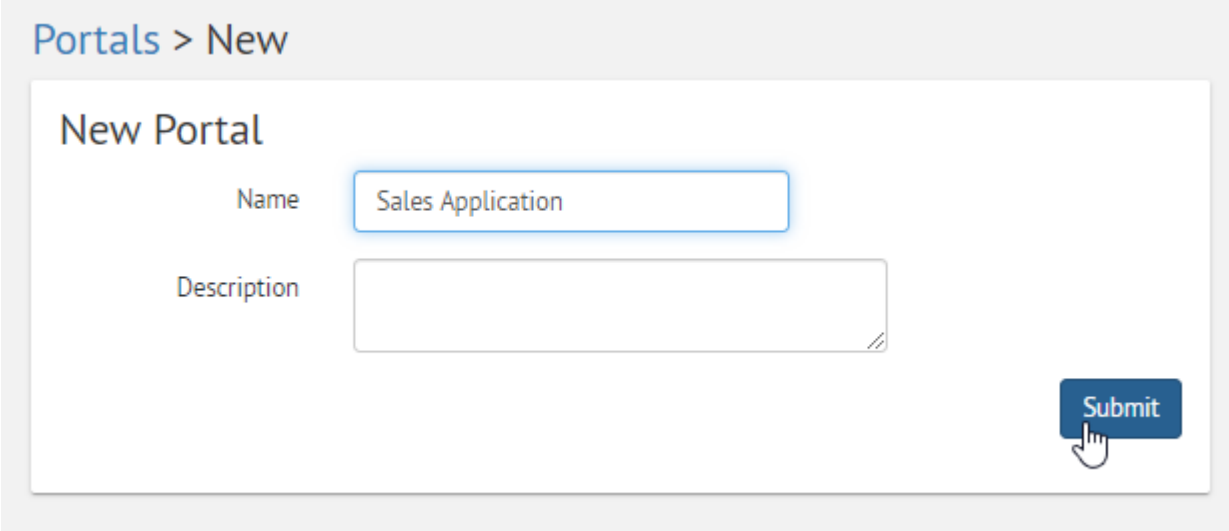
Steps

1. Select **Portals** from the main navigation menu.



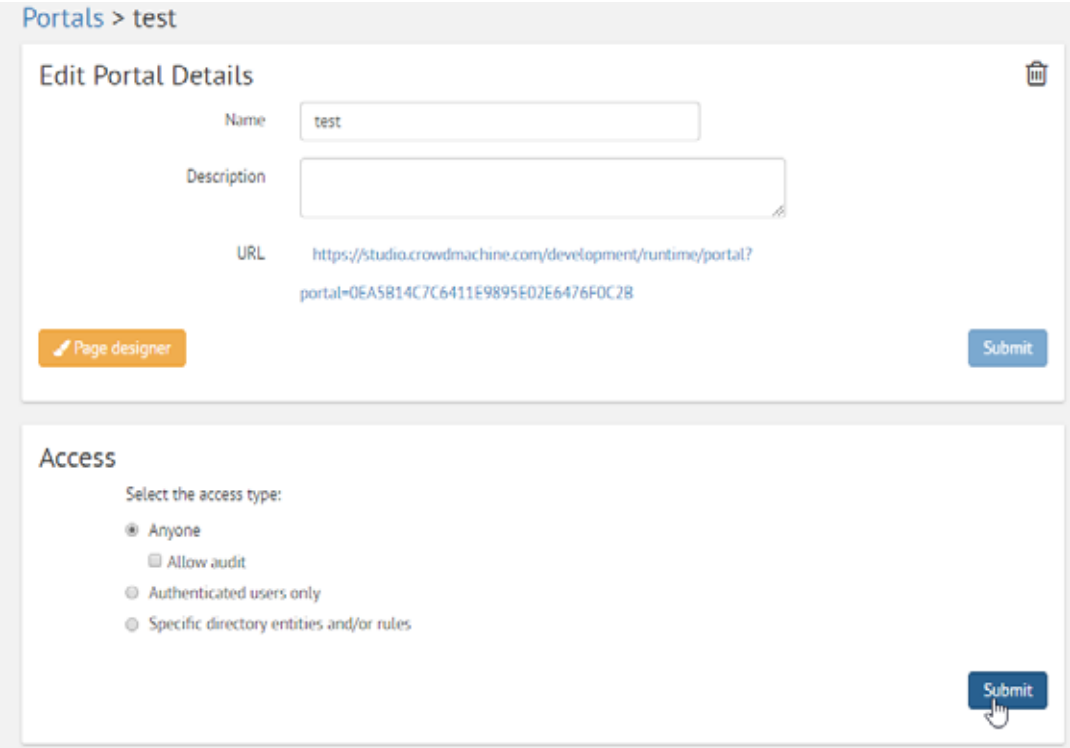
2. Click the blue plus button and name the portal **Sales Application**.

3. Click **Submit** to save the portal.



The portal is created and the URL is displayed.

4. Set the Access type to **Anyone** and click **Submit**.



5. Follow the instructions in the video tutorial to configure and test the portal.

This is the final piece of functionality to add to the Sales Application.