

LABORATORIUM 4. PROGRAMOWANIE WSPOMAGANIA PROCESÓW BIZNESOWYCH. PRACA W PROCESS BUILDER I FLOW DESIGNER.

Zadanie 4.1. Stworzenie procesu z wykorzystaniem Process Buildera

Property P-0000

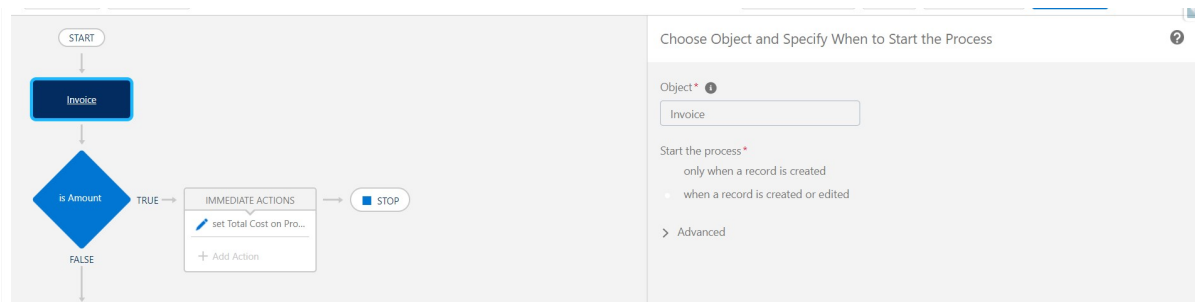
Related Details

Property Name
P-0000

Description
sdvsfdvbsd

Account
Brittenet

Total Cost
500.00



START

Invoice

is Amount

TRUE

IMMEDIATE ACTIONS

set Total Cost on Pro...

STOP

FALSE

+ Add Criteria

TRUE

IMMEDIATE ACTIONS

+ Add Action

STOP

FALSE

STOP

Define Criteria for this Action Group

Criteria Name * 1

is Amount

Criteria for Executing Actions *

Conditions are met

Formula evaluates to true

No criteria—just execute the actions!

Set Conditions

1

Field *

[Invoice_c].Amo...

Operator *

Is null

Type *

Boolean

Value *

False

Conditions *

All of the conditions are met (AND)

Any of the conditions are met (OR)

Customize the logic

Advanced

Save

Cancel

START

Invoice

is Amount

TRUE

IMMEDIATE ACTIONS

set Total Cost on Pro...

STOP

FALSE

+ Add Criteria

TRUE

IMMEDIATE ACTIONS

+ Add Action

STOP

FALSE

STOP

Update Records

Action Name * 1

set Total Cost on Property

Record *

[Invoice_c].Property

Criteria for Updating Records *

Updated records meet all conditions

No criteria—just update the records!

Set new field values for the records you update

Field *

Total Cost

Type *

Formula

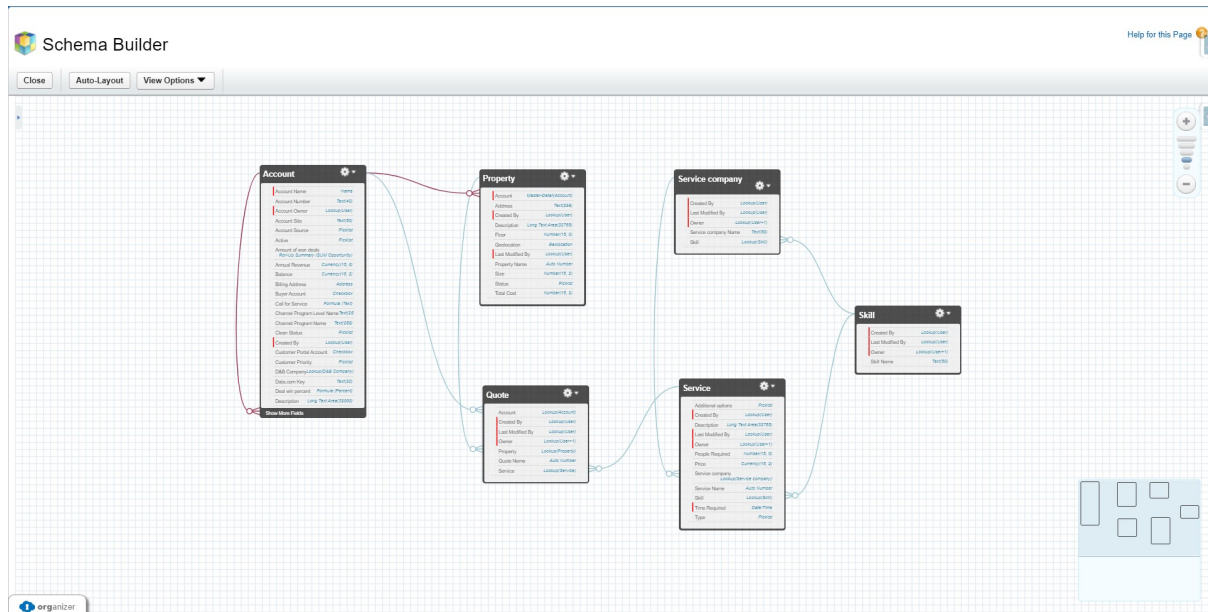
Value *

[Invoice_c].Amount_c

Save

Cancel

Zadanie 4.2. Stworzenie Flow





Start

Record-Triggered Flow

Object: **Service company**

[Edit](#)

Trigger: **A record is created or updated**

Optimize for: **Actions and Related Records**

[+ Add Scheduled Paths \(Optional\)](#)

[Open Flow Trigger Explorer for Service com...](#)

Run Immediately



**set Service Company on
Service**

Update Records



End

Edit Update Records

- ☐ Use the service company record that triggered the flow
- ☐ Update records related to the service company record that triggered the flow
- ☐ Use the IDs and all field values from a record or record collection
- ☒ Specify conditions to identify records, and set fields individually

Update Records of This Object Type

* Object

Service

Filter Service Records

Condition Requirements to Update Records

All Conditions Are Met (AND)

Field

Skill__c

Operator

Equals

Value

Aa \$Record > Skill X

+ Add Condition

Set Field Values for the Service Records

Field

Service_company__c

Value

← Aa \$Record > Record ID X

+ Add Field

Cancel

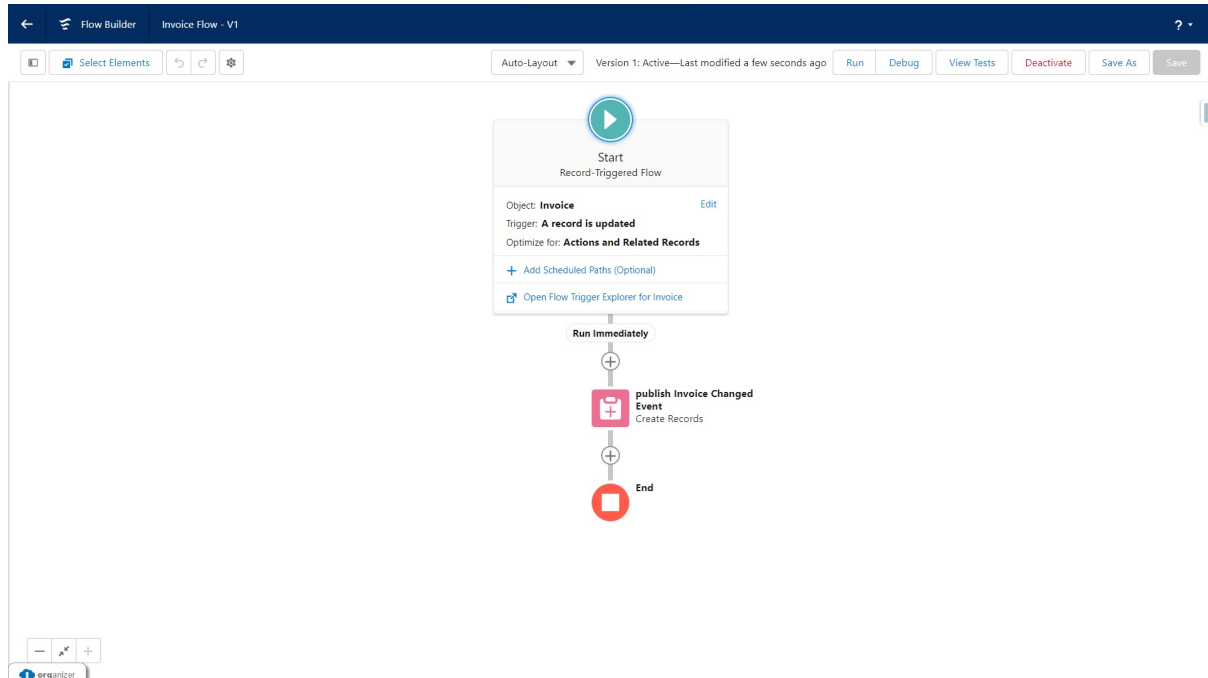
Done

Zadanie 4.3. Stworzenie mechanizmu powiadomień użytkownika

The screenshot shows the Salesforce Setup interface for Platform Events. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area is titled 'Platform Events' and displays the configuration for the 'Invoice Changed Event'. The configuration includes a 'Platform Event Definition Detail' section with fields for Singular Label, Plural Label, Object Name, API Name, Event Type, and Publish Behavior. Below this is a 'Standard Fields' table with columns for Action, Field Label, Field Name, Data Type, Controlling Field, and Indexed. The 'Custom Fields & Relationships' section shows a table with columns for Action, Field Label, API Name, Data Type, Indexed, Controlling Field, and Modified By. The 'Triggers' section is also visible at the bottom.

Action	Field Label	Field Name	Data Type	Controlling Field	Indexed
Created By	CreatedBy	CreatedBy	Lookup(User)		
Created Date	CreatedDate	CreatedDate	Date/Time		
Event UUID	EventUuid	EventUuid	Text(36)		
Replay ID	ReplayId	ReplayId	External Lookup		

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit Del	Email	Email_c	Text(255)			Jakub Labendowicz 17/10/2022, 17:50
Edit Del	ID	ID_c	Text(255)			Jakub Labendowicz 17/10/2022, 17:38
Edit Del	Name	Name_c	Text(255)			Jakub Labendowicz 17/10/2022, 17:38




×

Edit Create Records

×

Create Salesforce records using values from the flow.

publish Invoice Changed Event (publish_Invoice_Changed_Event) 

How Many Records to Create

☒ One

☐ Multiple

How to Set the Record Fields

☐ Use all values from a record










☒ Use separate resources, and literal values



Create a Record of This Object

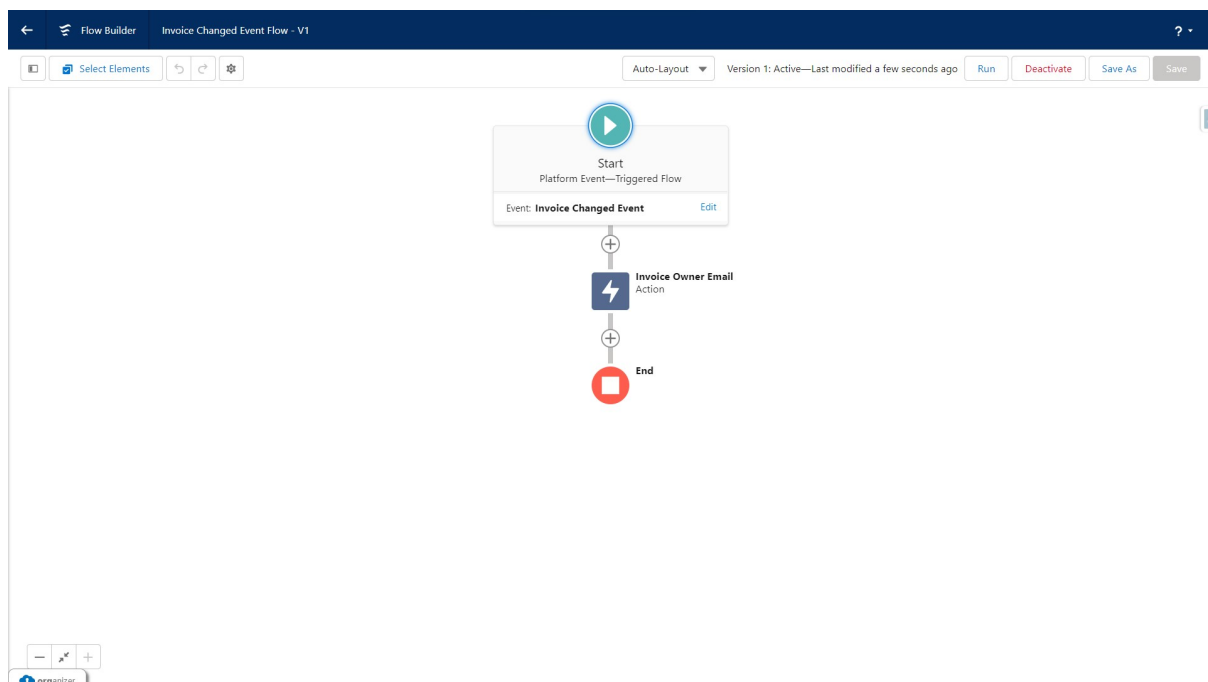
* Object

Invoice Changed Event

Set Field Values for the Invoice Changed Event

Field	Value
Id__c	  \$Record > Record ID 
Name__c	  \$Record > Invoice Name 
Email__c	  \$Record > Owner ID (User) > Email 


 





Edit "Send Email" core action

Use values from earlier in the flow to set the inputs for the "Send Email" core action. To use its outputs later in the flow, store them in variables.

Invoice Owner Email (Invoice_Owner_Email) 

Set Input Values

A_a *Body

{!InvoiceOwnerEmailBody}

A_a *Subject

{!InvoiceChanged}

A_a Recipient Email Addresses (collection)



Don't Include

A_a Recipient Email Addresses (comma-separated)

{!\$Record.Email__c}



Include

 Rich-Text-Formatted Body



Don't Include

A_a Sender Email Address



Don't Include

A_a Sender Type



Cancel

Done