

PEGASYSTEMS PROVIDES THIS SOFTWARE 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT. IN NO EVENT WILL PEGASYSTEMS, THE AUTHORS, OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT, OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNECTION WITH THIS SOFTWARE OR THE USE OR OTHER DEALINGS IN THIS SOFTWARE.

# Introduction

The Pega Angular Starter Kit shows how to construct an Angular application to utilize the Pega-APIs with layout and field information.

If a business user changes the layout of section, number of fields or type of fields; then, with the information in the Pega-API, the Angular application should be able to adapt without re-coding.

The Angular Starter Kit example builds a Case Worker portal which supports *any* simple Pega application.

The CableConnect Sample Application is simple sample application that can be loaded on to your Pega Infinity 8.1.3 (and higher) system. This sample app exercises many of the Pega-API DX features.

## Pega-API

In Pega Infinity, Pega-API's for cases/casetypes/assignments were updated with new extension points to provide layout/field information. This information comes from Harness/Section model information and can be used as hints to allow your Angular displays to be model driven.

This allows your displays to be linked to the Pega application, such that if the business user changes the design via the Case Designer or App Studio, your display can instantly reflect the changes, without re-coding.

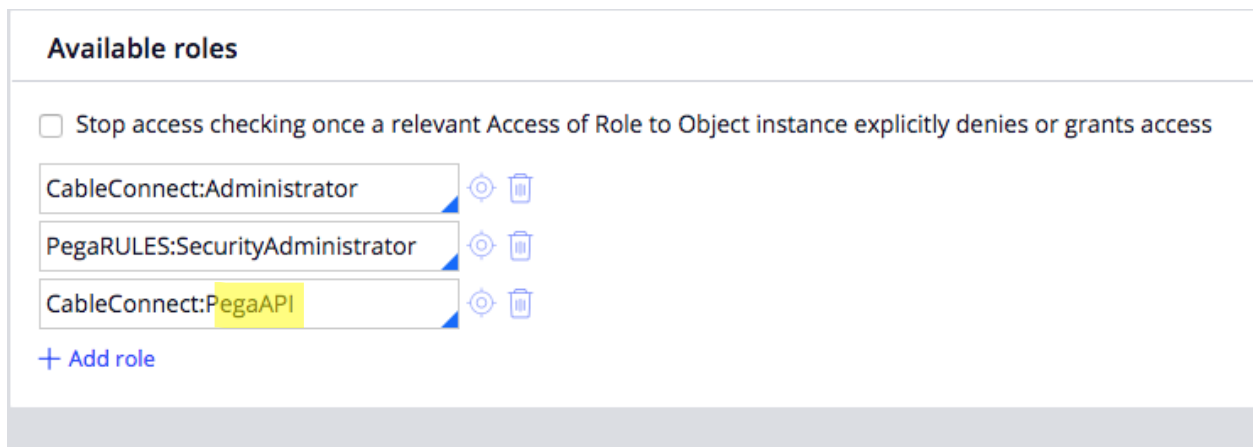
Layout and field information of a section/harness is passed through but organized differently to provide a simplified REST API JSON standard.

You can reference the Angular Starter Kit Installation Guide for more information.

# Cable Connect Sample Application

In order to quickly understand the benefits and usage of the Angular Starter Kit, we have provided a sample Pega Infinity application called CableConnect which you can install on your Pega Infinity (8.1.3 and higher) system.

Note: If you create a new application or modify this application and create new access group, ensure that your access group contains the PegaAPI role, in order to interface with PegaAPI DX (the Pega Digital Experience APIs).



## Description

Cable Connect is a simple application that consists of 3 users and admin:

- Representative – person contacted by a customer to get new service from a cable provider
- Tech – person who would fulfill the customer request and check off the request as being done
- Manager – person who can add a customer discount, if the representative requests one, and has access to multiple workbaskets.

Please note, that this application is very simple and not designed to be a full use case, but instead, be a simple application that you can use as an example.

## Installation

- Download zip
- Login to Pega system
- Import (Configure>Application>Distribution>Import)
- Log off
- Log on as admin.cableco (see below)

## Login

There are 4 logins for this application:

- rep.cableco, password: pega - case worker
- tech.cableco, password: pega – case worker
- manager.cableco, password: pega – case manager
- admin.cableco, password: pega – developer/admin

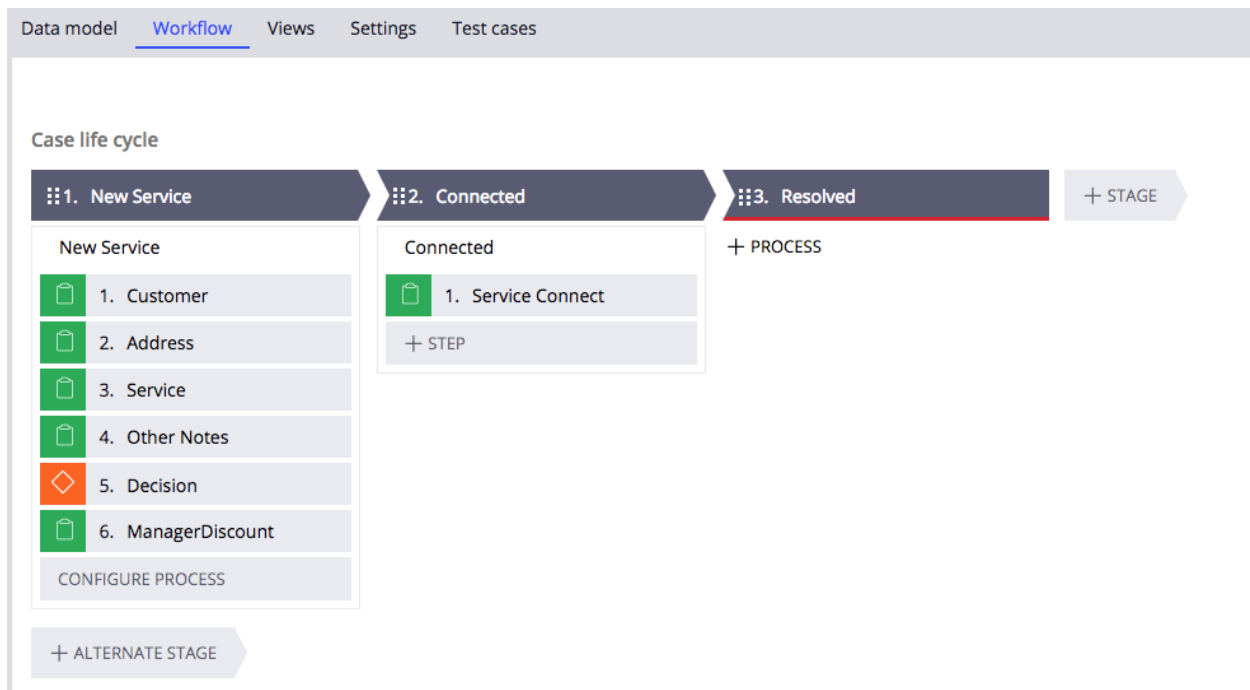
You can use these logins to update/edit/run the application. You might choose to run the application in your browser first, to understand the flow. See “step through” section for more information.

## Supporting Functionality

Cable Connect application supports the following functionality, that will be exercised in the Angular Starter Kit.

- Fields & Controls: pxTextInput, pxTextArea, pxCheckbox, pxRadioButtons, pxDropDown, pxAutoComplete, pxDateTime, pxButton, pxNumber & Disabled controls
- Modes: Editable, ReadOnly
- Layouts: Single Column, 3 Column
- Actions: Refresh with Post, Perform Action
- Validation: Property Edit Validate, Rule Edit Validate, Required
- Routing: Rep->Tech, Rep->Manager->Tech

Here is the Case life cycle.



Note: The majority of this application was created within the Case Designer.

Note: "Service" case type skips the "New" harness:

### Behavior

- ☒ Skip 'Create' view when users create a new case
- ☐ Create temporary case that is not saved until a 'Persist case' step is reached ?

If you create an application that does not "skip" the "New" harness, then you WILL need to override the "New" harness in your application, as the out of the box "New" harness currently does NOT work with the PegaAPI DX.

The flow of this application is as follows:

rep.cableco

- gathers customer information, address, requested service and notes
- Rep determines whether to request a discount
- Rep is now finished – routed to Manager (if a discount is needed) or Tech to fulfill

manager.cableco

- Manager needs to get work items from manager workbasket
- If any work items request a discount, then manager can enter a discount

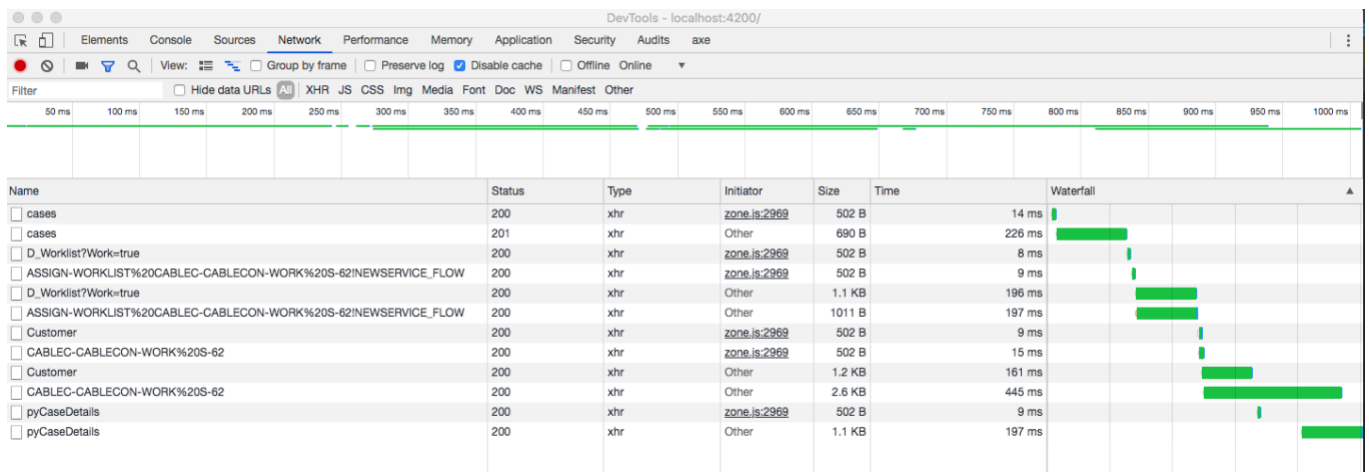
- Manager is now finished – routed to Tech to fulfill

tech.cableco

- Works is routed to Tech from either Rep or Manager (if discount)
- Tech is supposed to verify fulfillment of service
- Tech can edit “server request” and change if via “Update service”
- Tech resolves work item

A few points:

- This application is simple, so one can understand the basics.
- Work items get routed, so you can see multiple users and their corresponding work list and/or workbasket.
- Works items for manager get routed to a “manager workbasket” so you can see an example of work sent to a workbasket as well as see an example of how to show lists (display) of workbaskets based upon user’s access group.
- Not all Pega API DX functionality and information is exercised here.
- Utilize the Chrome browser Developer Tools (Network tab) when running the Angular app so you can trace the network traffic and see the API calls, and responses.



# Step Through on the regular desktop

## Customer

This is the first screen when logged in as "Rep". The following is exercised here:

- Text Input
- Required fields
- Autocomplete based upon Prompt List (Suffix)
- Email with Property Edit validate
- Date time with Rule Edit validation on Submit (date can't be in the past)

**Service (S-57)** NEW

**Customer** R

First Name \*

Middle Name

Last Name \*

Suffix

Email \*

Service Date

- Rule Edit Validate (ValidateCustomer)

PROPERTY	*Req Conditions	Edit
.ServiceDate	IF @isDateInThePast(.ServiceDate) THEN display message: Service date MUST either be TODAY or be in the future	

## Address

This is the second screen. The following is exercised here:

- Drop down with data based upon a data page (for State value)
- Phone number with Property edit validation (ValidPhoneNumber is overridden)

**Service (S-57)** NEW

**Address**

Street

1 Rogers St

City

Cambridge

State

MA ▾

Postal Code

02142

Phone number

6178666000

Cancel

Save

Submit

## Service

This is the third screen. The following is exercised here:

- Checkboxes have "Refresh" section. When checked, sub section appears
- Sub sections are visible via a Server side "when rule"
- Radio buttons

**Service (S-57)** NEW

**Service**

**TV/Cable Service**  
☒ TV  
TV Option  
☐ Basic  
☒ Basic Plus  
☐ Deluxe  
☐ Premium

**Internet Service**  
☒ Internet  
Internet Option  
☐ 25 Mbps  
☒ 100 Mbps  
☐ 300 Mbps

**Home Phone Service**  
☐ Phone

Cancel

Save

Submit



## Notes

This is the fourth screen. The following is exercised here:

- If checkbox is checked, upon submit, work item will be routed to a manager workbasket. Otherwise, will be routed to "Tech" user.

**Service (S-57)** NEW

**Notes** R

**Other Notes**

Notes

asking for \$25 off

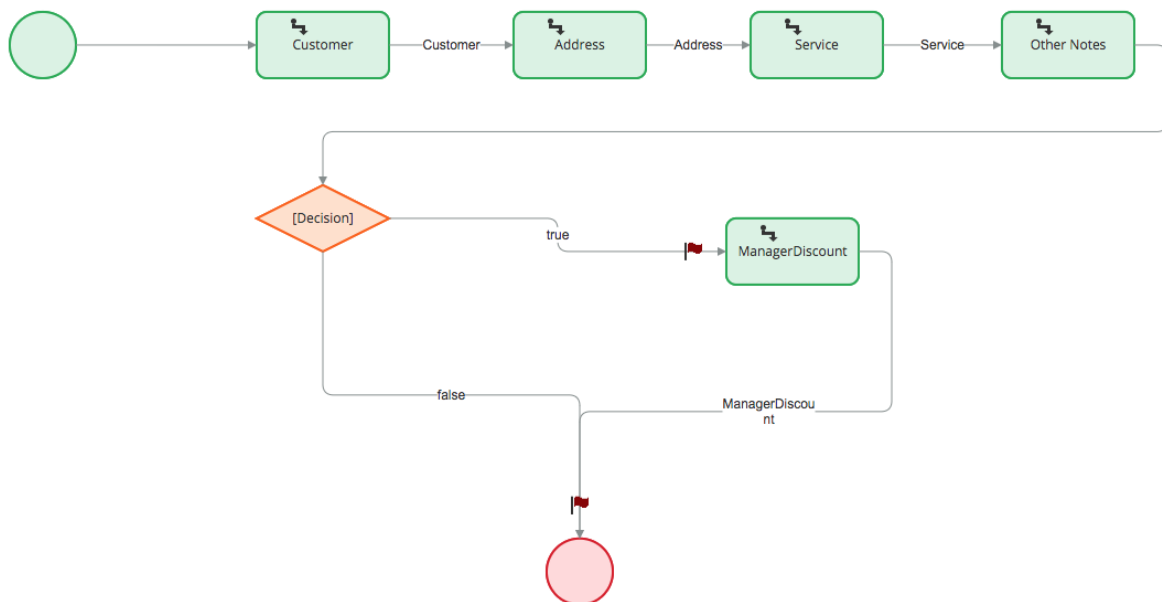
☒ Send to Manager for Discount

Cancel

Save

Submit

In the flow, we have a decision based upon the checkbox



## Manager Discount

This screen is seen by the manager. Work item will appear in manager worklist. Manager can add a customer discount. The following is exercised here:

- ReadOnly
- Number

**Service (S-57)** **PENDING MANAGER**

**ManagerDiscount**

M

Notes to Manager  
asking for \$25 off

Customer Discount

Cancel

Save

Submit

## Service Connect

This screen is seen by the "Tech" user. The "Tech" user is expected to fulfill the request and check off that which has been done. The "Tech" user has the ability to update the "service" (see update service)

The following is exercised here:

- Display Text (formatted text)
- Disabled fields
- Perform Action (call local action to update server) – Update Service

**Service (S-57)** **PENDING-FULFILLMENT**

### Service Connect

**Customer**  
First Name  
John  
Last Name  
Smith  
Email  
john@smith.com  
Phone number  
6178666000  
Expected Service Date  
Friday, February 15, 2019  
Discount  
\$25.00

**Chosen Services**  
TV Option  
☐ Basic  
☒ Basic Plus  
☐ Deluxe  
☐ Premium  
Internet Option  
☐ 25 Mbps  
☒ 100 Mbps  
☐ 300 Mbps  
Update Service

**Fulfillment**  
☒ TV Connected  
☒ Internet Connected  
☐ Phone Connected

Cancel

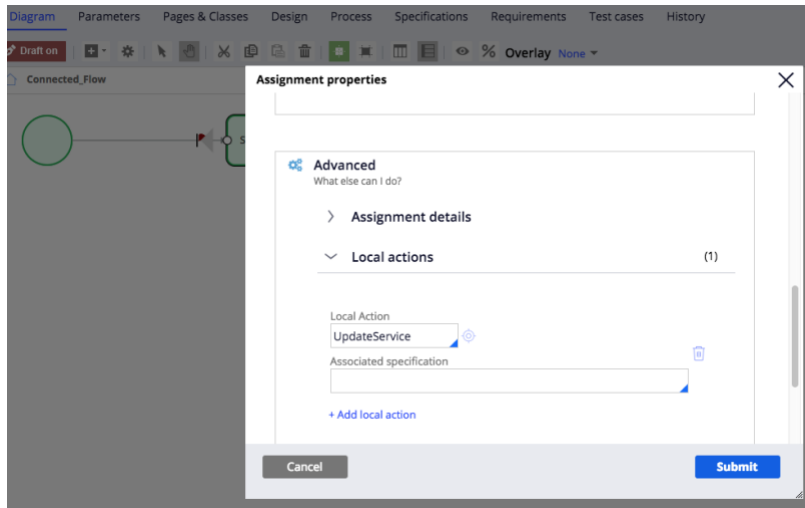
SaveSubmit

## Update Service

This screen is seen if "Update Service" button is pressed in prior screen. This is a local action that can update the "service". Once the change of service is made, submit will be pressed and the work item will be routed back to the "ServiceConnect" screen.

The following is exercised here:

- Local action on an assignment



**Service (S-57)** PENDING-FULFILLMENT

### UpdateService

**TV/Cable Service**  
☒ TV  
TV Option  
☐ Basic  
☒ Basic Plus  
☐ Deluxe  
☐ Premium

**Internet Service**  
☒ Internet  
Internet Option  
☐ 25 Mbps  
☒ 100 Mbps  
☐ 300 Mbps

**Home Phone Service**  
☐ Phone

Cancel

Save

Submit

## Confirm

During the process, the user can see a confirm screen when no further action is required. The Confirm harness has to be overridden, as the default will currently not work with PegaAPI DX.

The following is exercised here:

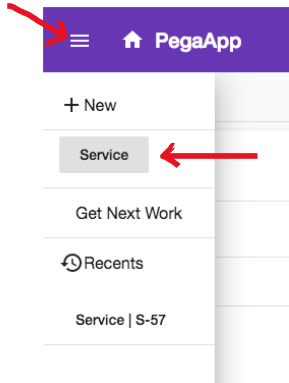
- Overridden Confirm harness with screen layout

**Thank you for your submission. No further action is required.**

# Step Through Angular Application

## After login

Create a new Service work item, select the hamburger icon, then select "Service"



## Customer

This is the first screen when logged in as "Rep". The following is exercised here:

- Text Input
- Required fields, error handling
- Autocomplete based upon Prompt List (Suffix)
- Tooltips (Angular hints)

A screenshot of the 'New Service' form in the PegaApp. The form is titled 'New Service > Connected > Resolved' and has an 'Actions' dropdown in the top right. The form is divided into two main sections: 'Customer' and 'Case details'. The 'Customer' section contains fields for 'First Name \*' (John), 'Middle Name', 'Last Name \*' (Smith), 'Suffix', 'Email \*' (john@smith.com), and 'name@provider.com' (Service Date: 2/15/2019). The 'Case details' section contains fields for 'Last updated by' (Rep.CableCo, a few seconds ago) and 'Created by' (Rep.CableCo, a few seconds ago). At the bottom of the form are three buttons: 'Cancel', 'Save', and 'Submit'.

- Autocomplete via prompt list

The screenshot shows a web application interface with a breadcrumb trail: **New Service** > **Connected** > **Resolved**. The page is titled **Customer** and **Case details**. The **Customer** section includes fields for **First Name \*** (John), **Middle Name**, **Last Name \*** (Smith), and **Suffix**. The **Suffix** field has an open dropdown menu showing options: **Sr -- (Sr)**, **Jr -- (Jr)**, **III -- (III)**, **IV -- (IV)**, and **V -- (V)**. The **Case details** section includes **Last updated by** (Paul Gagnon, 2 minutes ago) and **Created by** (Rep.CableCo, 28 minutes ago). At the bottom right, there are **Save** and **Submit** buttons.

## Error handling

- Required – local error handling in Angular App

The screenshot shows a form field labeled **First Name \*** with the value **John**. Below the field, a red error message reads: **You must enter a value**.

- Bad email – both local in Angular App, and on the server via a submit

The screenshot shows a form field labeled **Email \*** with the value **a**. Below the field, a red error message reads: **Not a valid email**.

- Bad Date (not in the future) – on server via submit

The screenshot shows a form field labeled **Service Date** with the value **2/1/2019**. Below the field, a red error message reads: **Service date MUST either be TODAY or be in the future**. A calendar icon is visible on the right side of the field.

## Address

This is the second screen. The following is exercised here:

- Drop down with data based upon a data page (State)
- Phone number with Property edit validation (default is overridden)

Dashboard S-58 X

New Service > Connected > Resolved Actions

### Address

Street  
1 Rogers St

City  
Cambridge

State  
MA

Postal Code  
02142

Phone number  
617 866-6000  
(###) ###-####

Case details

Last updated by  
Rep.CableCo  
a few seconds ago

Created by  
Rep.CableCo  
a minute ago

Cancel Save Submit

- Drop Down via data page

Dashboard S-58 X

New Service > Connected > Resolved Actions

### Address

Street  
1 Rogers St

City  
Cambridge

State  
MA

Postal Code  
02142

Phone number  
617 866-6000  
(###) ###-####

Case details

Last updated by  
Paul Gagnon  
a few seconds ago

Created by  
Rep.CableCo  
10 minutes ago

Cancel Save Submit



## Service

This is the third screen. The following is exercised here:

- Checkboxes have "Refresh" section. When checked, the sub section appears
- Sub sections are visible via a Server side "when rule", so when you call the "refresh" API, the server will recalculate based upon "when rules" and return data that will have the new section(s).
- Radio buttons

The screenshot shows a web form titled 'Service' with a breadcrumb 'New Service > Connected > Resolved'. The form is divided into two main sections: 'Service' and 'Case details'. Under 'Service', there are three sub-sections: 'TV/Cable Service', 'Internet Service', and 'Home Phone Service'. Each sub-section has a checkbox: 'TV' (unchecked), 'Internet' (unchecked), and 'Phone' (unchecked). The 'Case details' section shows 'Last updated by' as 'Paul Gagnon a few seconds ago' and 'Created by' as 'Rep.CableCo 31 minutes ago'. At the bottom, there are 'Cancel', 'Save', and 'Submit' buttons.

- Selecting checkboxes causes refresh and visible sub sections

The screenshot shows the same 'Service' form, but now the 'TV' and 'Internet' checkboxes are selected. This has triggered a refresh, and sub-options are now visible. Under 'TV/Cable Service', there is a 'TV Option' section with radio buttons for 'Basic', 'Basic Plus' (selected), 'Deluxe', and 'Premium'. Under 'Internet Service', there is an 'Internet Option' section with radio buttons for '25 Mbps', '100 Mbps' (selected), and '300 Mbps'. The 'Home Phone Service' checkbox remains unchecked. The 'Case details' section now shows 'Last updated by' as 'Rep.CableCo a few seconds ago' and 'Created by' as 'Rep.CableCo 2 minutes ago'. The 'Cancel', 'Save', and 'Submit' buttons are still at the bottom.

## Notes

This is the fourth screen. The following is exercised here:

- If checkbox is checked, upon submit, work item will be routed to a manager workbasket. Otherwise, will be routed to "Tech" user.

Dashboard

S-58 X

New Service > Connected > Resolved

Actions ▾

Notes	Case details
<div>Other Notes</div> <div>Notes</div> <div>Asking for \$25 off</div>	<div>Last updated by</div> <div>Rep.CableCo</div> <div>a few seconds ago</div>
<div><input checked="" type="checkbox"/> Send to Manager for Discount</div>	<div>Created by</div> <div>Rep.CableCo</div> <div>2 minutes ago</div>
<div>Cancel</div>	<div>Save</div> <div>Submit</div>

## Manager Discount

This screen is seen by the manager. Work item will appear in manager worklist.

Dashboard		Worklist	
		CableConnect:Managers	
		CableConnect:Users	
Case	Status	Service	Urgency
S-61	Pending Manager		10
Items per page: 10 1 - 1 of 1  < < > >			

Manager can add a customer discount. The following is exercised here:

- ReadOnly field
- Number (only numbers can be entered)

Dashboard

S-58 X

New Service > Connected > Resolved

Actions

### ManagerDiscount

Notes to Manager

Asking for \$25 off

Customer Discount

25.00

\*Add a customer discount as XX.YY (dollars)\*

### Case details

Last updated by

Rep.CableCo

a few seconds ago

Created by

Rep.CableCo

3 minutes ago

Cancel

Save

Submit

## Service Connect

This screen is seen by the "Tech" user. The "Tech" user is expected to fulfill the request and check off that which has been done. The "Tech" user has the ability to update the "service" (see update service)

The following is exercised here:

- Display Text (formatted text, email, date, currency)
- Disabled fields
- Perform Action (call local action to update server)
- Triple Layout

Dashboard

S-58 X

New Service > Connected > Resolved

Actions ▾

Service Connect			Case details
Customer	Chosen Services	Fulfillment	Last updated by
First Name John	TV Option	<input checked="" type="checkbox"/> TV Connected	Manager.CableCo a few seconds ago
Last Name Smith	<input type="radio"/> Basic	<input checked="" type="checkbox"/> Internet Connected	<b>Created by</b>
Email john@smith.com	<input checked="" type="radio"/> Basic Plus	<input type="checkbox"/> Phone Connected	Rep.CableCo 4 minutes ago
Phone number 617 866-6000	<input type="radio"/> Deluxe		
Expected Service Date Friday, February 15, 2019	<input type="radio"/> Premium		
Discount \$25.00	Internet Option		
	<input type="radio"/> 25 Mbps		
	<input checked="" type="radio"/> 100 Mbps		
	<input type="radio"/> 300 Mbps		
	<button>Update Service</button>		

Cancel

Save

Submit

## Update Service

This screen is seen if "Update Service" is pressed in prior screen. This is a local action that can update the "service". After submit, the work item will be routed back to the "ServiceConnect" screen.

The following is exercised here:

- Local action on an assignment

Dashboard

S-58 X

New Service > Connected > Resolved

Actions ▾

Service			Case details
<b>TV/Cable Service</b>	<b>Internet Service</b>	<b>Home Phone Service</b>	
<input checked="" type="checkbox"/> TV TV Option <input type="radio"/> Basic <input checked="" type="radio"/> Basic Plus <input type="radio"/> Deluxe <input type="radio"/> Premium	<input checked="" type="checkbox"/> Internet Internet Option <input type="radio"/> 25 Mbps <input checked="" type="radio"/> 100 Mbps <input type="radio"/> 300 Mbps	<input type="checkbox"/> Phone	<b>Last updated by</b> Manager.CableCo a few seconds ago  <b>Created by</b> Rep.CableCo 4 minutes ago
<div>Cancel</div>			<div>Save</div> <div>Submit</div>

## Confirm

During the process, the user can see a confirm screen when no further action is required. The Confirm harness has to be overridden, as the default will currently not work with PegaAPI DX (Digital Experience APIs).

The following is exercised here:

- Overridden Confirm harness with screen layout

The screenshot displays a Pega user interface for a confirmed case. At the top, a breadcrumb trail reads "New Service > Connected > Resolved" in orange, with an "Actions" dropdown menu to the right. The main content area is divided into two sections. On the left, under the heading "Status", the text "Resolved-Completed" is shown, followed by a large message: "Thank you for your submission. No further action is required." On the right, under the heading "Case details", there are two sections: "Last updated by" showing "Tech.CableCo" and "a few seconds ago", and "Created by" showing "Rep.CableCo" and "5 minutes ago". A "Close" button is located at the bottom center of the screen.

New Service > Connected > Resolved		Actions
<b>Status</b> Resolved-Completed	Thank you for your submission. No further action is required.	<b>Case details</b>  <b>Last updated by</b> Tech.CableCo a few seconds ago  <b>Created by</b> Rep.CableCo 5 minutes ago

Close