List and Detail Screens

Exercício – Como fazer

# 

**Esboço** 1

[**Como**](#_k6p8syz3iidi) **Fazer** 2

[Movies Screen](#_f0d2eda26zti) 2

[MovieDetail Screen](#_ogdkdme7ac1g) 9

[Link](#_drsqeniavpqw) entre as duas telas 20

[People Screen](#_2xduvxyei7na) 24

[PersonDetail Screen](#_pmibw51ajhs0) 25

[Linkan](#_bcuaddp2666w)do as Telas 31

# 

# Esboço

Nesse exercício, vamos começar construindo o UI de nossa aplicação. Vamos fazer quatro telas, duas List Screen para listar os filmes e pessoas no banco de dados e duas Detail Screens para permitir a criação e edição de novos filmes e pessoas.

* Movies Screen
  1. Tela para mostrar a lista de filmes na Entity Movie.
  2. A lista de filmes deve ser mostrada com um registro por linha, com o title, year, plot summary e os gross amounts.
* MovieDetail Screen
  1. Tela para mostrar os detalhes de um filme em particular, identificado pelo seu Id em um formulário.
  2. Tenha certeza de que a tela irá permitir criar novos filmes, bem como editar os existentes.
  3. Construa a lógica para criar/atualizar um filme no banco de dados.
  4. Adicione um Link entre a tela MovieDetail e Movies para voltar à lista, quando necessário. Na Tela Movies, adicione um Link entre cada filme à respectiva Tela MovieDetail, para exibir seus dados detalhados e um Link na parte superior da Tela para permitir a criação de um novo filme.
* People Screen
  1. Tela para mostrar a lista de pessoas na Entity Person.
  2. A lista de pessoas deve ser mostrada com um registro por linha contendo: name, surname e date of birth.
* PeopleDetail Screen
  1. Tela para mostrar os detalhes de uma pessoa em particular, identificada pelo seu Id no formulário
  2. Tenha Certeza de que a tela permitirá a criação de novas pessoas, bem como editar as já existentes.
  3. Desenvolva a lógica para criar/atualizar uma pessoa no banco de dados.
  4. Adicione um Link entre a tela PersonDetail e People para voltar à lista, quando necessário. Na Tela de People, adicione um Link entre cada pessoa à respectiva PersonDetail, para exibir seus dados detalhados e um Link na parte superior da Tela para permitir a criação de uma nova pessoa.

# Como fazer

Nesta seção, vamos descrever o passo a passo do exercício 3 - *List and Detail Screens*

## Movies Screen

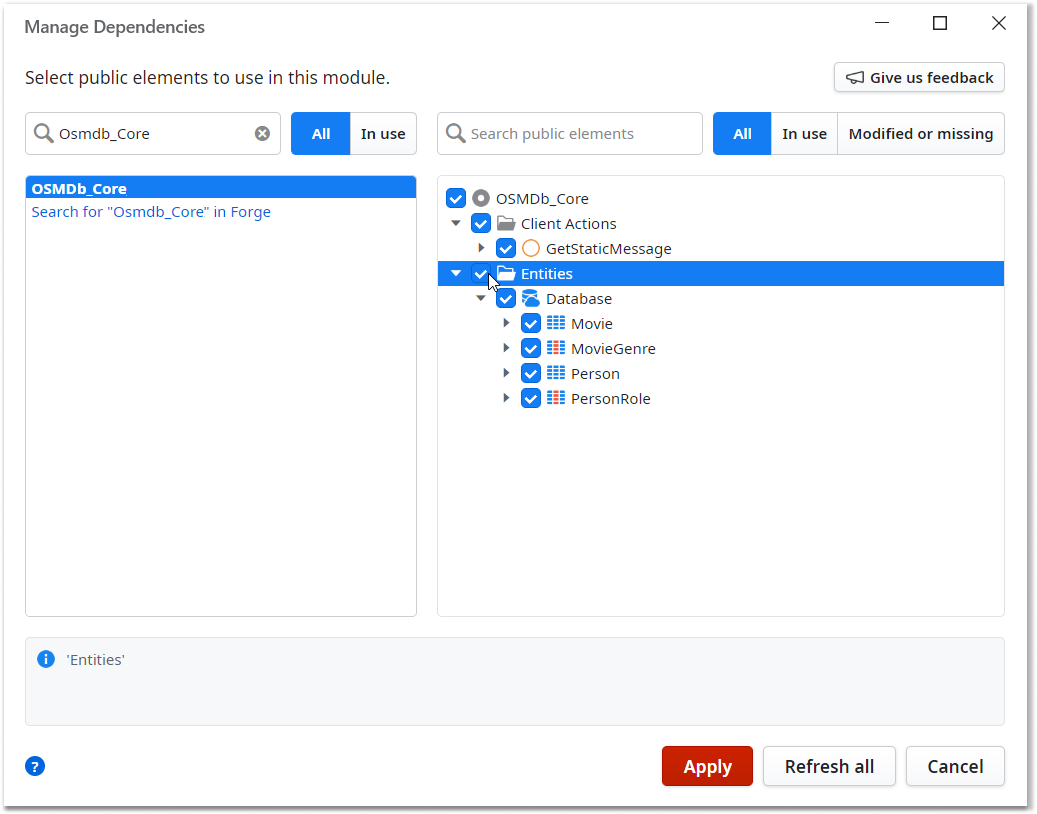
Vamos começar criando a tela de filme (Movie Screen).

1. No módulo OSMDb, referencie as Entities e Static Entities criadas no exercício anterior, usando a seção Manage Dependencies.
   1. Vá para o módulo OSMDb e abra o **Manage Dependencies**.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

* 1. Selecione seu OSMDb\_Core e selecione as quatro Entities: Movie, People, MovieGenre e PersonRole.



**NOTA:** Se as Entities não estiverem visíveis quando tentar referenciar, confira se o módulo central foi publicado ou se as Entities estão definidas como públicas. Se não estiverem, precisamos muda-las e publicá-las novamente.

* 1. Clique em **Apply** para fechar a janela.

1. Mude a página HomeScreen para a tela Movies. Delete a Expression. Na tela e crie um Aggregate para buscar todos os filmes do banco de dados. Esses filmes serão posteriormente mostrados em tela
   1. Renomeie a tela **HomeScreen** para *Movies***.**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

* 1. Clique na aba **Widget Tree** para abrir a estrutura de tela. Expanda **Content** para achar a Expression adicionada no exercício anterior. Clique com o botão direito nela e a delete.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**NOTA:** A Widget Tree é uma ferramenta muito útil para construir a UI em OutSystem. Apresenta uma estrutura hierárquica da tela, facilitando a visualização de cada elemento (e todos os outros elementos dentro dela se for o caso).

Também é muito útil para adicionar novos elementos na tela, ajudando o desenvolvedor a alocar o elemento exatamente onde deseja, como veremos em exercícios futuros.

* 1. De volta na guia Elements, clique com o botão direito do mouse no Movie e escolha **Fetch Data from Database.**

*Interface gráfica do usuário, Aplicativo

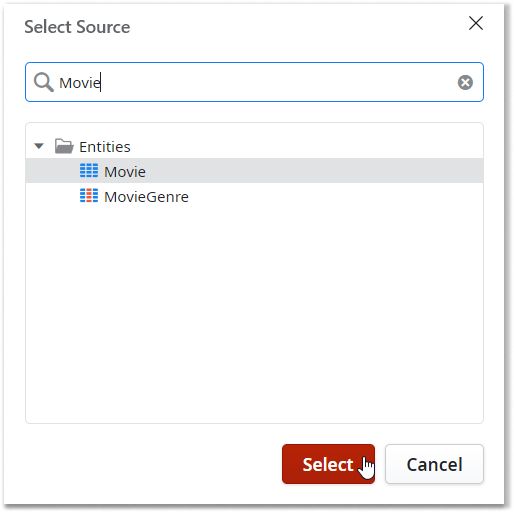
Descrição gerada automaticamente*

* 1. Como indicado nas instruções na tela, clique na área principal para adicionar um source ao Aggregate.

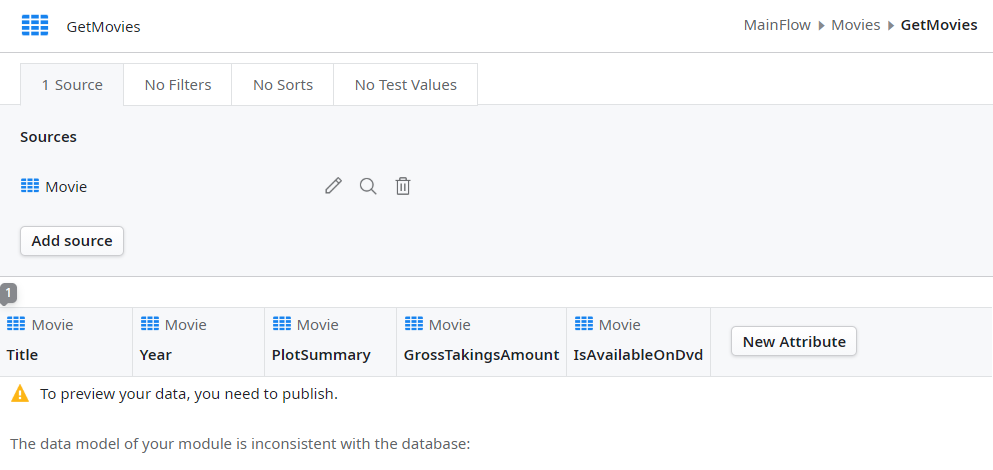
Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

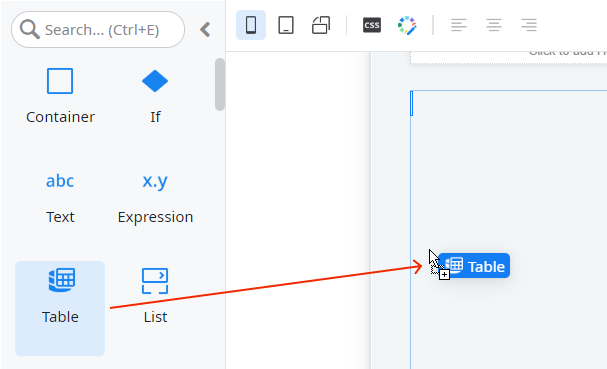
* 1. Selecione a Entity **Movie** e clique em **Select**. O Aggregate é automaticamente renomeado para *GetMovies.*

**

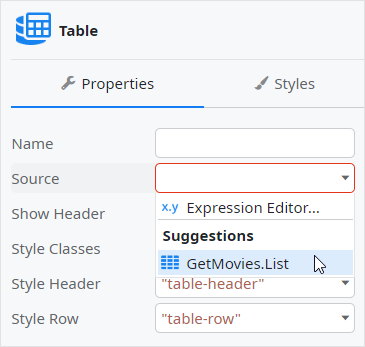
* 1. Você deve ter algo semelhante ao print a seguir.

**

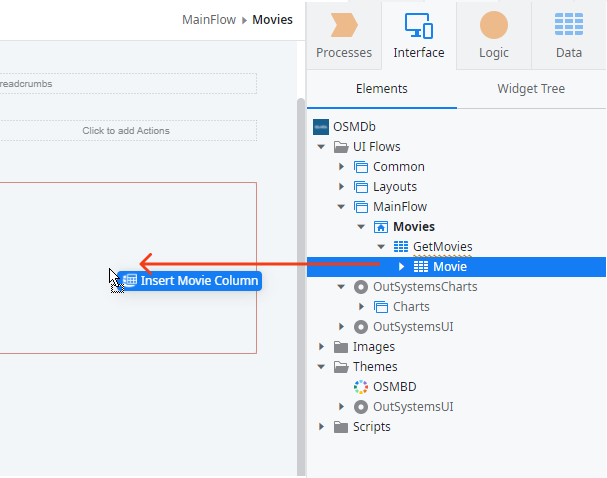
1. Defina a UI da tela Movies usando uma **Table** que irá mostrar todos os filmes com o **Title**, **Year**, **PlotSummary** e **GrossTakingsAmount**. Também se certifique se o título da tela foi definido como *Movies*.
   1. Duplo clique na tela **Movies** para abri-la.
   2. Enconte o widget Table, depois arraste e solte no conteúdo principal da tela



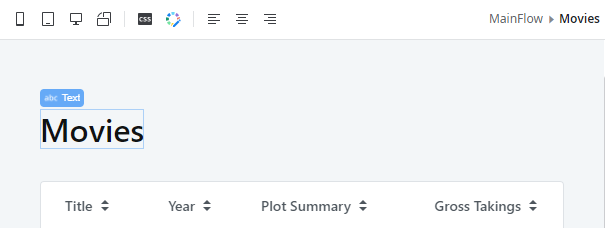
* 1. Defina a propriedade **Source** da lista para o output do Aggregate GetMovies.



* 1. Abra o aggregate GetMovies e arraste a Entity Movies para a tabela.

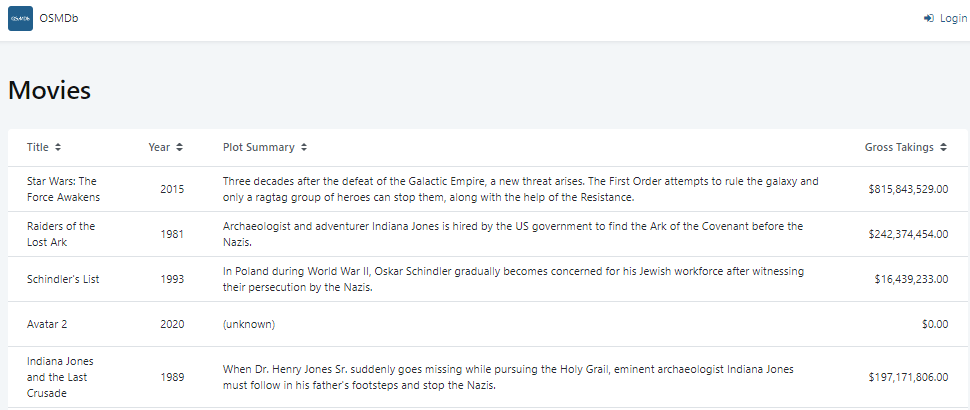


* 1. Defina o título da tela como Movies.

****

* 1. **Publish** a aplicação e abra no navegador. Se assegure que todos os filmes aparecem na tela.

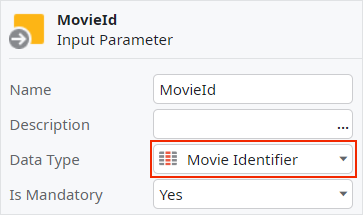
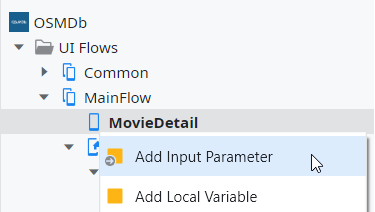




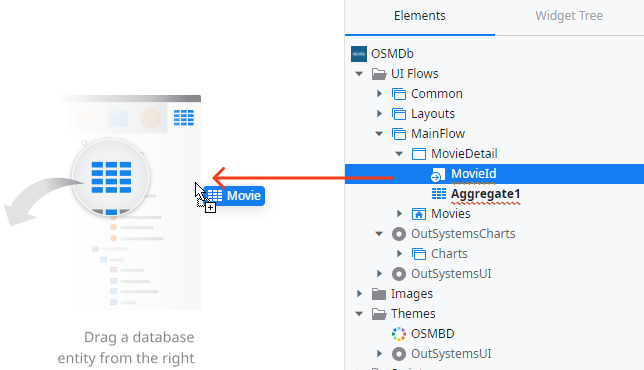
## 

## MovieDetail Screen

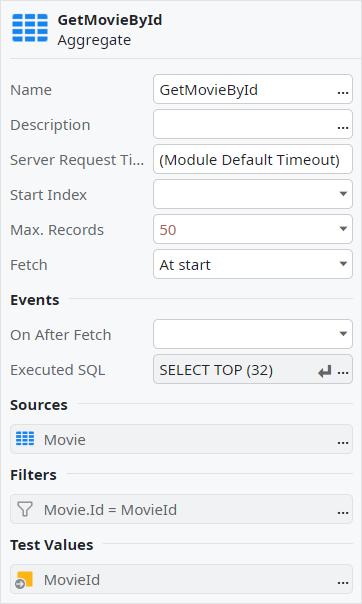
1. Crie uma tela chamada MovieDetail a partir de um Empty template. Essa tela deve ter um parâmetro de entrada como identificador de um filme, *MovieId*, e um Aggregate para buscar os filmes com o Id específico.
   1. Crie uma Empty Screen, nomeie-a como *MovieDetail* e defina como **Anonymous**
   2. Adicione um parâmetro de entrada chamado *MovieId* com o **Data Type** para *Movie Identifier*.

**

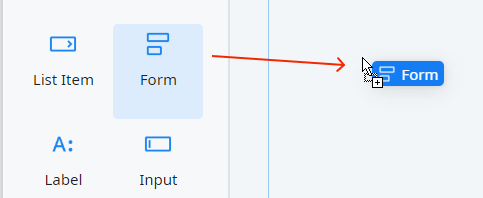
* 1. Crie um Aggregate clicando com o botão direito na tela e selecionando a opção **Fetch Data from Database**.
  2. Arraste o parâmetro de entrada **MovieId** para o Aggregate



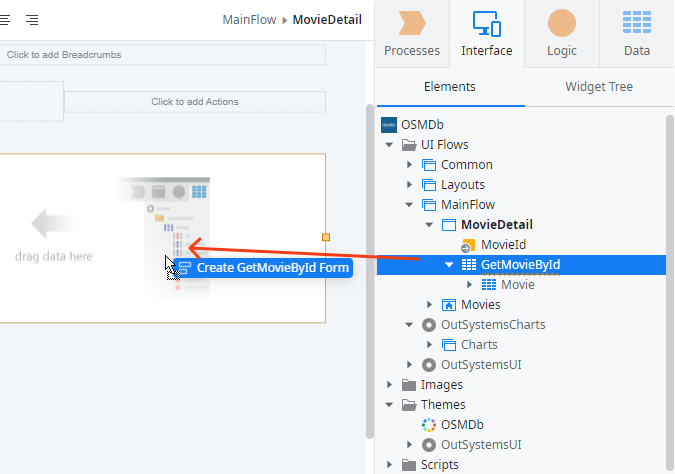
* 1. Note que o Aggregate foi renomeado para *GetMovieById*, A Entity Movie foi adicionada como source e um novo filtro usando o MovieId como parâmetro de entrada foi criado.



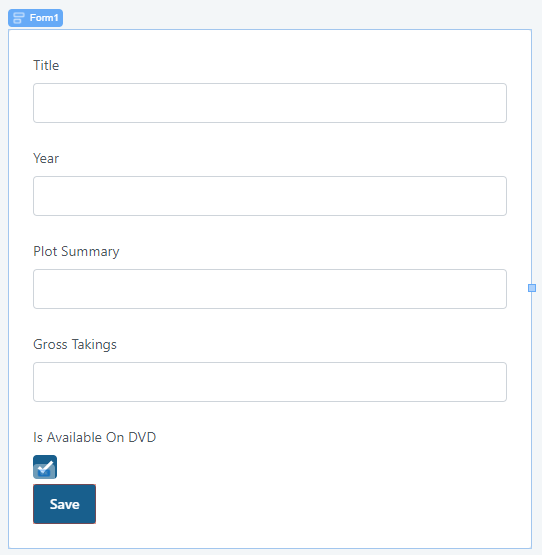
1. Crie um Form na tela MovieDetail com um campo de entrada para cada atributo na Entity Movie. Configure o **Title** da tela para mostrar Create New Movie quando o MovieID for NullIdentifier(), ou quando o valor do atributo *Title*  não for igual a Entity Movie.
   1. Duplo clique na tela **MovieDetail** para abri-la.
   2. Arraste um Form e solte na área de conteúdo da página.



* 1. Arraste o Aggregate **GetMovieById** e solte dentro do formulário

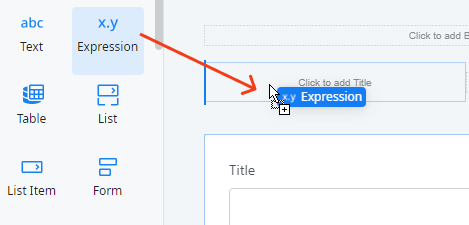


* 1. Todos os atributos da Entity Movie foram adicionados como campos de entrada no formulário. O usuário final irá usá-los para enviar as informações dos filmes.

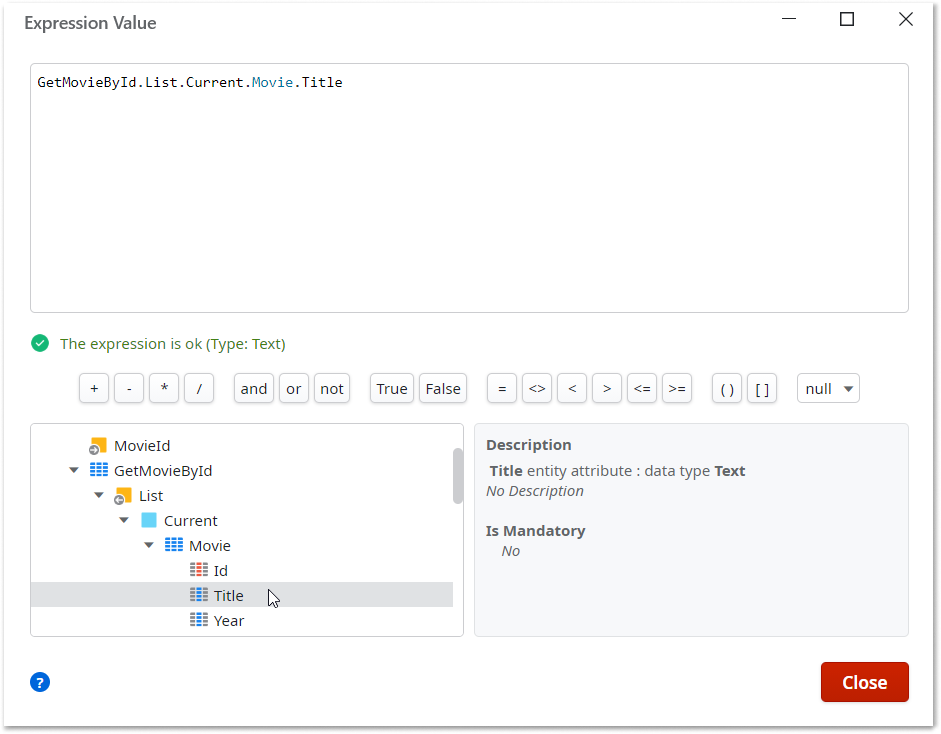


**NOTA:** Uma mensagem de erro vai aparecer, mas não precisamos nos preocupar com isso no momento.

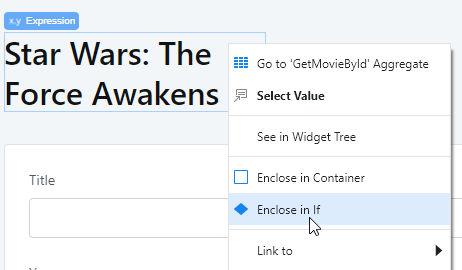
* 1. Arraste uma **Expression** para a seção de título da tela.



* 1. Defina a Expression para *GetMovieById.List.Current.Movie.Title*

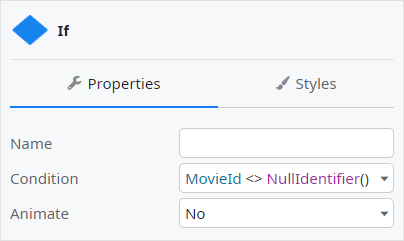


* 1. Clique com o botão direito na Expression e selecione a opção **Enclose in If**.

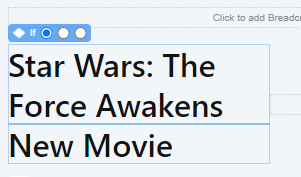


* 1. Configure a **Condition** do if para

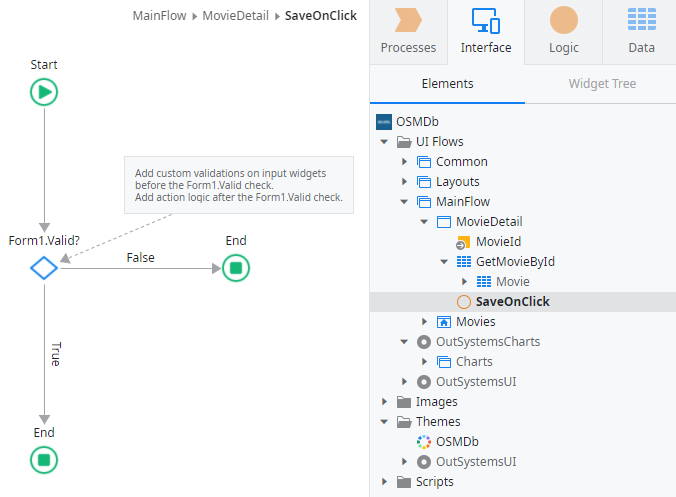
*MovieId <> NullIdentifier()*

**

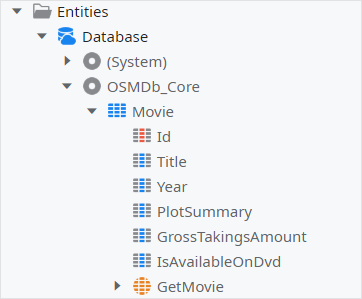
* 1. Na parte falsa do if, insira New Movie



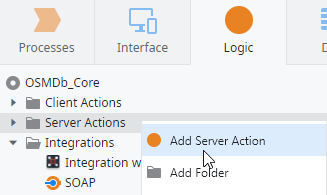
1. Defina a lógica criar/atualizar um filme no banco de dados e associe com o Save button. A lógica do lado servidor deve ser implementada no módulo central (Core), desde que a Entity seja publicada como read only.
   1. Duplo clique em **Save** button para criar uma Screen Action



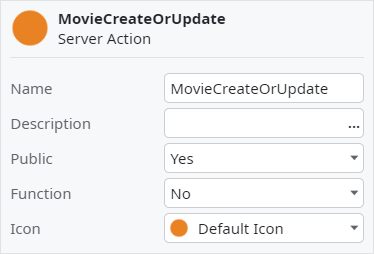
* 1. Agora precisamos adicionar a lógica criar/atualizar o filme no banco de dados. Desde que a Entity esteja publicada como read-only, só teremos acesso a Entity Action Get no módulo OSMDb. Então precisamos criar a lógica necessária no Core e publica-lá como pública, assim poderemos usar no módulo OSMDb



* 1. Vá até o Core e abre a aba Logic
  2. Clique com o botão direito na pasta Server Actions e clique em **Add Server Action**



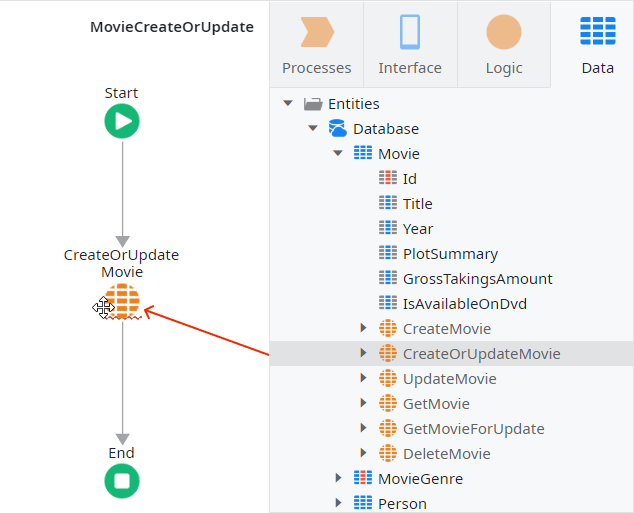
* 1. Nomeie para *MovieCreateOrUpdate* e defina como **Public**



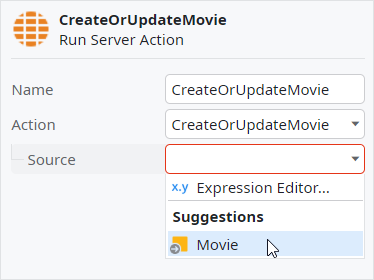
* 1. Adicione um parâmetro de entrada na Action e nomeie como Movie. Confirme que o tipo de dado seja Movie.
  2. Adicione um parâmetro de saída e nomeie como *MovieId*. O **Data Type** deve ser *Movie Identifier*.



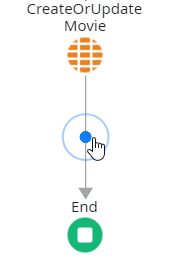
* 1. Arraste a Entity Action **CreateOrUpdateMovie** para o fluxo da Action.



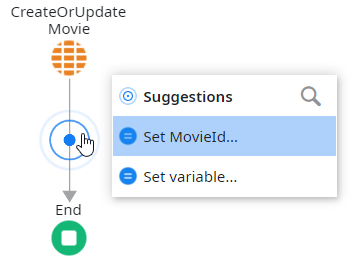
* 1. Selecione o nó CreateOrUpdateMovie arrastado no passo anterior e defina sua source para **Movie** (o parâmetro de entrada da Action).



* 1. Passe o mouse sobre a última seta do fluxo e clique no ícone azul.

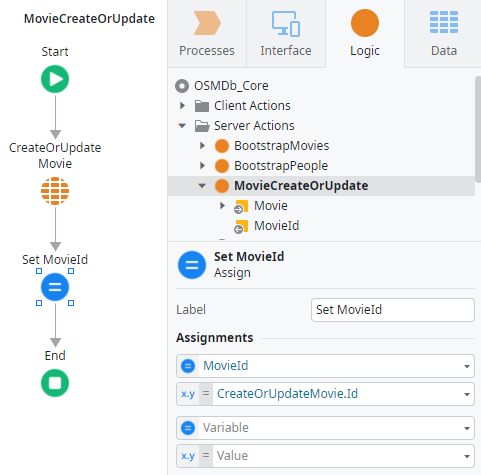


* 1. Escolha **Set MovieId** e defina seu valorpara **CreateOrUpdateMovie.Id**



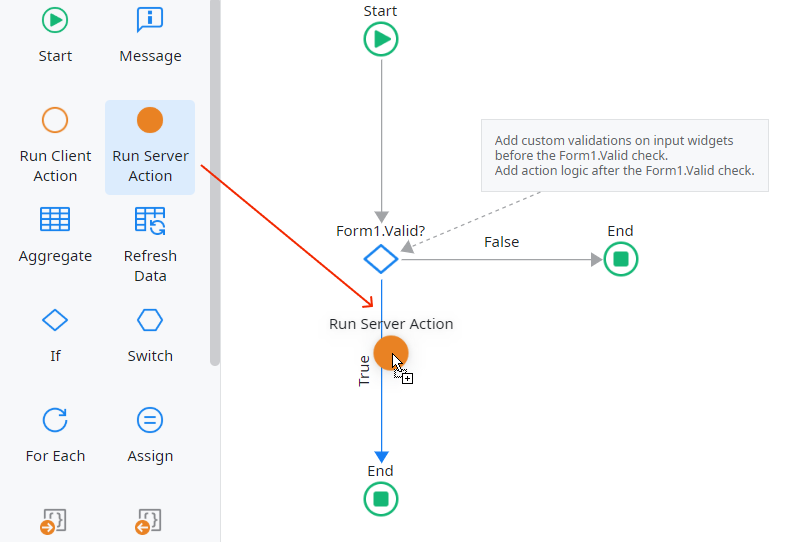
Se essa opção não aparecer e você vir a opção **Set MovieId…** selecione esta.

* 1. Automaticamente, o parâmetro de saída **MovieId** foi atribuído como saída da Action **CreateOrUpdateMovie**, que é o Id do filme criado/modificado. Se você pegar a opção **Set MovieId…** no passo anterior, o Assign é criado, mas teremos que atribuir explicitamente o valor do *CreateOrUpdateMovie.Id* para **MovieId**.

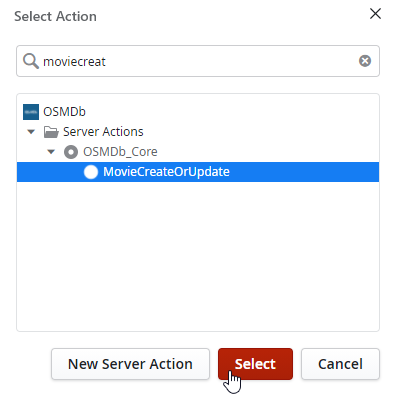


**NOTE:** Esse é um exemplo de como podemos potencializar a inteligência artificial do Service Studio, com algumas recomendações dos próximos passos no fluxo da Action. Neste caso, desde que a Action tenha um parâmetro de saída e que a Action CreateOrUpdateMovie retorne um valor com o mesmo tipo de dado. O Service Studio irá sugerir a Assign. Isso varia de aplicação para aplicação e dos padrões mais usados pelos desenvolvedores.

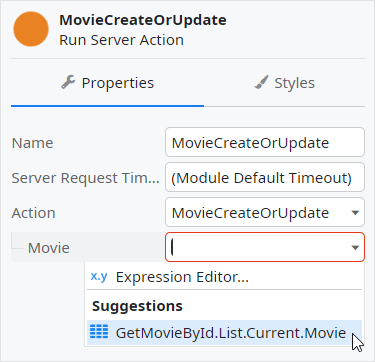
* 1. Publique o módulo **OSMDB\_Core**.
  2. Volte para o módulo OSMDb e use o **Manage dependencies** para referenciar a Server Action **MovieCreateOrUpdate** do seu módulo central (core).
  3. Abra o A Screen Action **SaveOnClick** da tela MovieDetail
  4. Arraste o nó **Run Server Action** para o fluxo e solte abaixo do IF



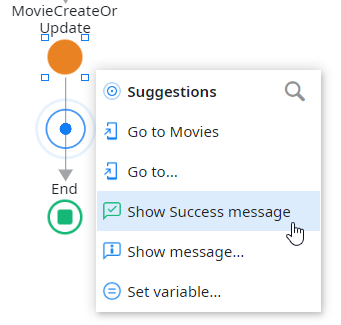
* 1. Na nova janela, selecione **MovieCreateOrUpdate** e clique em OK



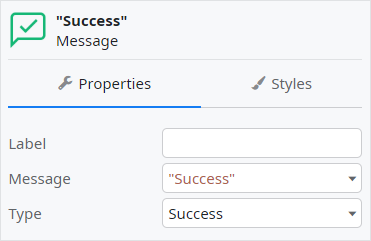
* 1. Nas propriedades da Action, defina o parâmetro de entrada Movie da Action para *GetMovieById.List.Current.Movie.* Já que esse registro é retornado pelo Aggregate que está sendo editado no Form, podemos usá-lo para passar o filme a ser criado/editado para a Action MovieCreateOrUpdate.



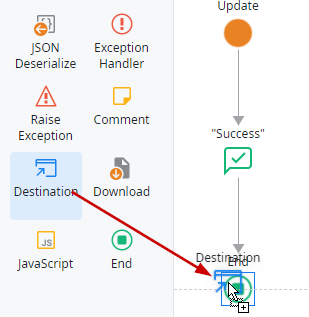
* 1. Use o ícone azul para Suggestions para criar uma mensagem de sucesso



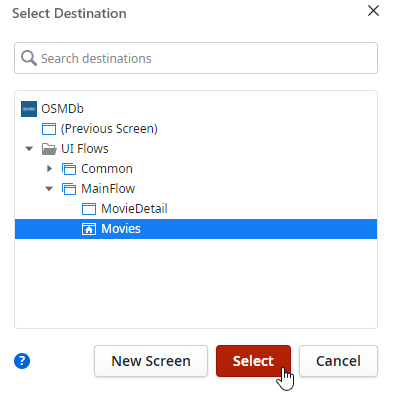
* 1. Defina a **Message** para algo do tipo “Success!”



* 1. Arraste um nó **Destination** acima do nó End



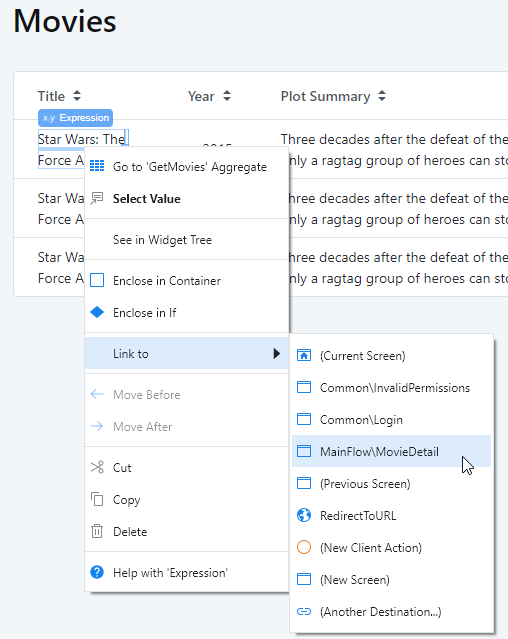
* 1. Selecione a tela **Movies** e clique em **OK**



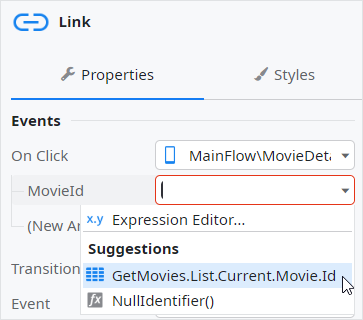
## Linkando as duas Telas

Agora que temos as duas telas para os filmes criadas, temos que nos assegurar de fazer uma navegação adequada entre elas. A tela Movies deve ter dois links para a tela MovieDetail, uma para criar novos filmes e um para os já existentes na tabela para permitir sua edição. Na tela Movie Detail, temos que ter um link que permita voltar a tela Movies.

1. Na aba Interface, clique duas vezes na tela Movies, depois com o botão direito clique em **title** expression. -> Link To -> **MainFlow\MovieDetail**

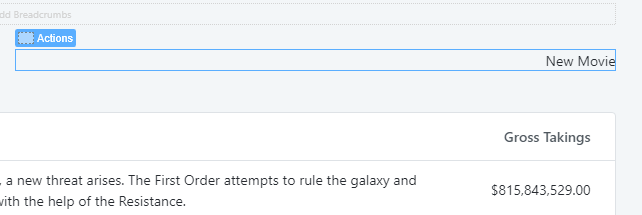


1. No parâmetro de entrada do link **On Click** Destination selecione *GetMovies.List.Current.Movie.Id.*

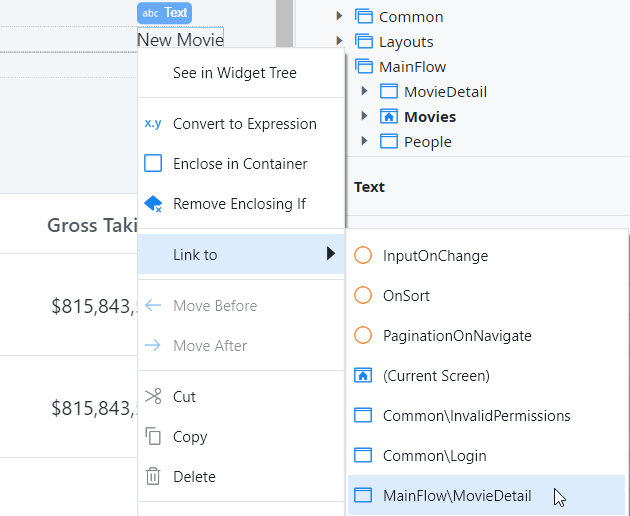


**NOTE:** Isso garante que o Id do filme selecionado é passado para a tela MovieDetail. Neste caso, O MovieDetail irá mostrar os detalhes do filme que foi selecionado e o usuário final será capaz de ver seus detalhes ou editar essa informação.

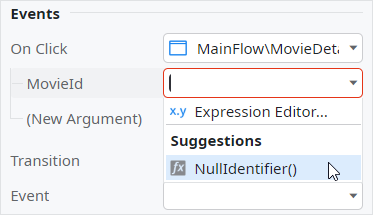
1. Na parte superior direita da tela, na área de Action, coloque *New Movie*



1. Com o botão direito clique no texto New Movie e selecione **Link to -> MainFlow\MovieDetail**

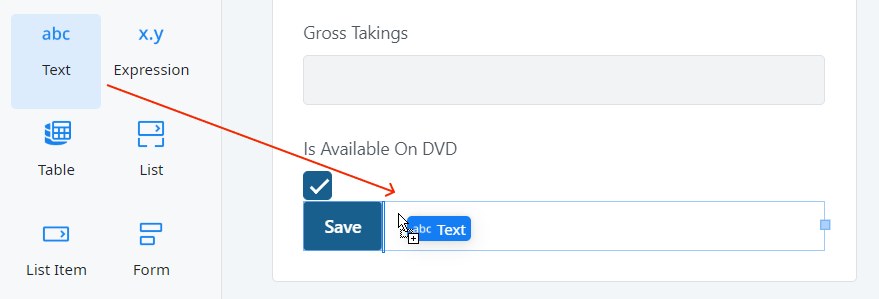


1. Use **NullIdentifier()** para o parâmetro de entrada



**NOTA:** Neste caso, quando tem um filme novo, ele ainda não possui um Id ainda, então passamos o valor *NullIdentifier()*.

1. Na tela **MovieDetail**, arraste o widget **Text** e solte próximo ao botão

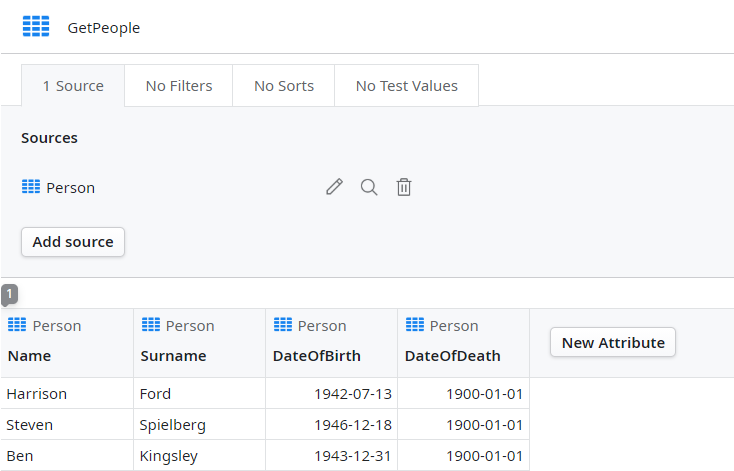
****

1. Defina o texto como *Back to Movies*.
2. Clique com o botão direito no texto e crie um Link para a tela Movies.
3. **Publish** o módulo e teste no navegador!

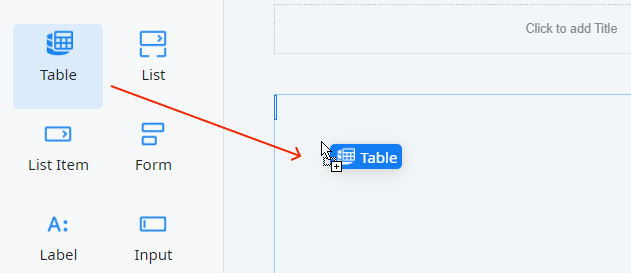
## People Screen

Agora, iremos criar as telas para as pessoas, assim como fizemos nas de filmes. Vamos começar criando a tela People baseada numa tela em branco (Empty Screen) que lista todas as pessoas do banco de dados.

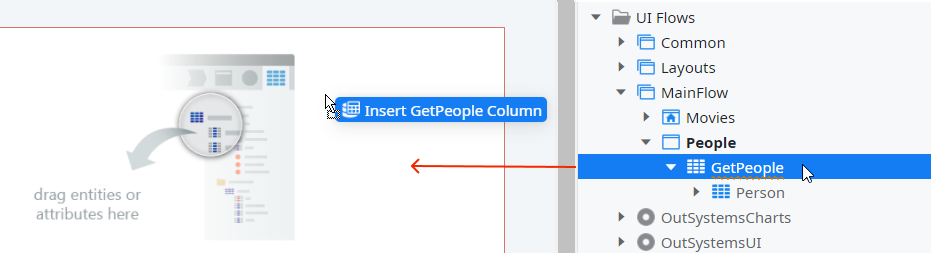
1. Crie uma **Empty** Screen e a nomeie como People e a defina como **Anonymous**.
2. Insira um Aggregate na tela com a Entity Person como source.



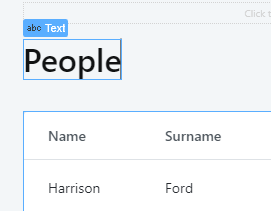
1. Arraste uma Table para a People Screen.



1. Arraste o Aggregate **GetPeople** à lista



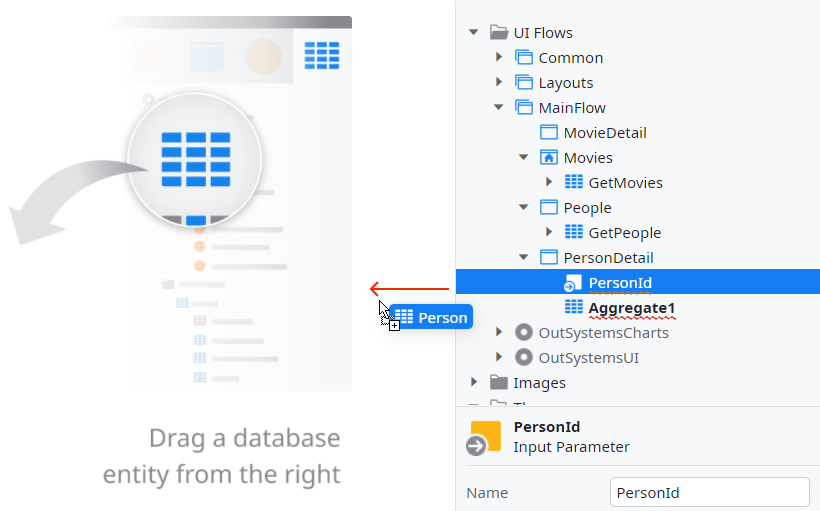
1. Defina o título da tela como *People*



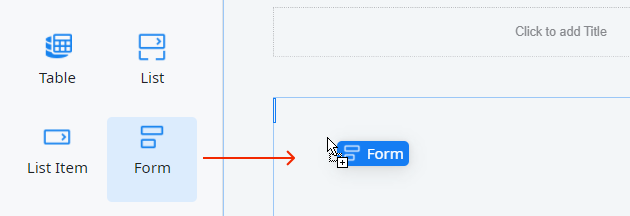
## PersonDetail Screen

A tela PersonDetail possui um Form com campos de entrada para todos os atributos da Entity Person. Assim como no MovieDetail, a tela espera o Id de Person como entrada. Essa entrada deverá ser usada para buscar a informação sobre a pessoa. Também, a lógica criar/atualizar a pessoa no banco de dados deverá ser disparada por um botão de salvar.

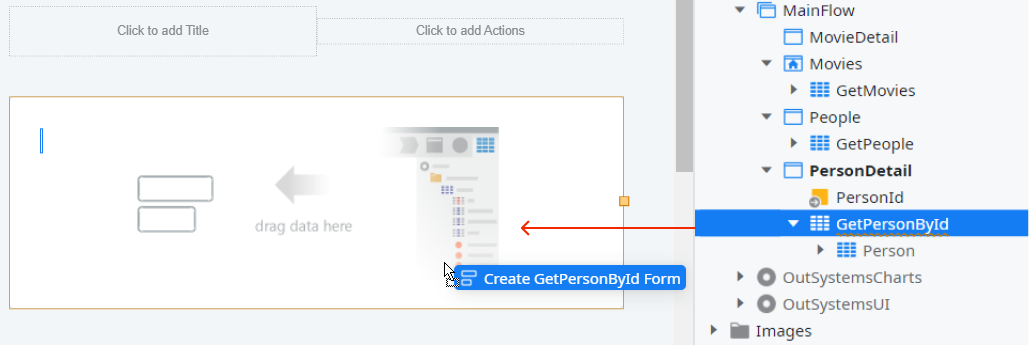
1. Crie uma Empty Screen, nomeie-a PersonDetail e defina-a para **Anonymous**. A tela deve ter um parâmetro de entrada PersonId e um Aggregate para obter as informações sobre a Pessoa.
   1. Crie uma **Empty** Screen, nomeie como PersonDetail e defina como **Anonymous**.
   2. Insira um parâmetro de entrada chamado PersonId. Se assegure que seu **Data Type** é *Person Identifier***.**
   3. Crie um Aggregate e use o parâmetro de entrada PersonId para retornar apenas a pessoa que o Id corresponda com o valor do parâmetro de entrada. O Aggregate deverá mudar o nome para *GetPersonById*.



* 1. Arraste um Form para a tela PersonDetail

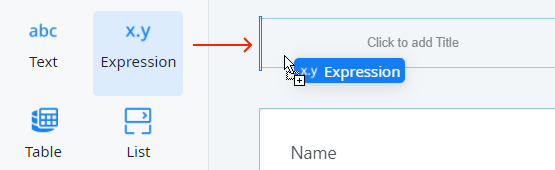


* 1. Arraste o Aggregate **GetPersonById** e solte dentro do Form



* 1. Arraste uma Expression. E solte no título da tela. Defina seu valor como:

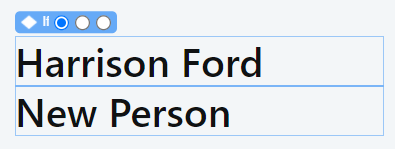
*GetPersonById.List.Current.Person.Name + " " + GetPersonById.List.Current.Person.Surname*



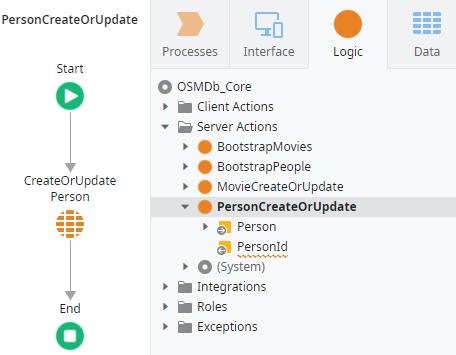
* 1. Coloque a Expression dentro de um If e defina a condição para:

*PersonId <> NullIdentifier()*

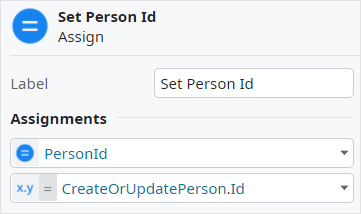
* 1. Na parte falsa, coloque *New Person*



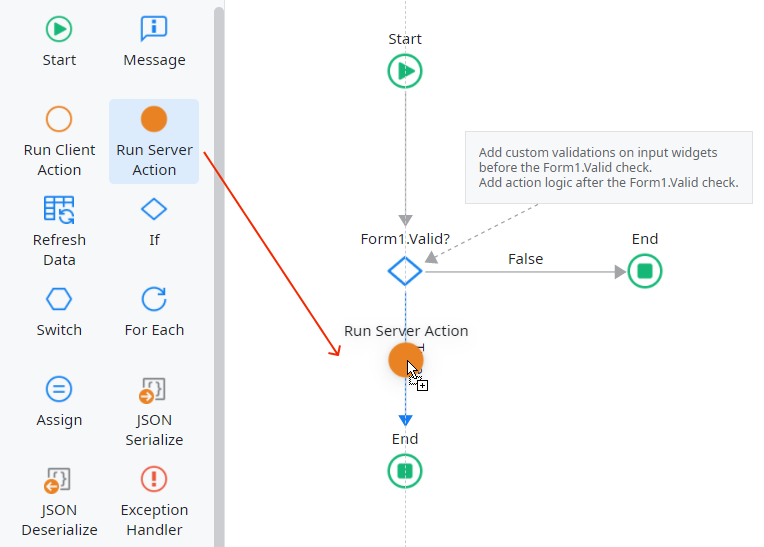
1. Crie a lógica criar/atualizar uma pessoa no bando de dados e se assegure que será ativada quando clicar no Save button.
   1. Clique duas vezes no Save button para criar a Action *SaveOnClick*.
   2. Volte para o módulo central (core) e crie uma **Server Action**. Nomeie como *PersonCreateOrUpdate* e defina como **Public**.
   3. Adicione um parâmetro de entrada na Action e nomeie como Person. Se assegure que seu **Data Type** aparecerá como Person.
   4. Insira um parâmetro de saída na Action e nomeie como PersonId. Se assegure que seu **Data Type** está definido como *Person Identifier*.
   5. Arraste a Entity Action **CreateOrUpdatePerson** para o fluxo da Action e defina sua source como o parâmetro de entrada Person.



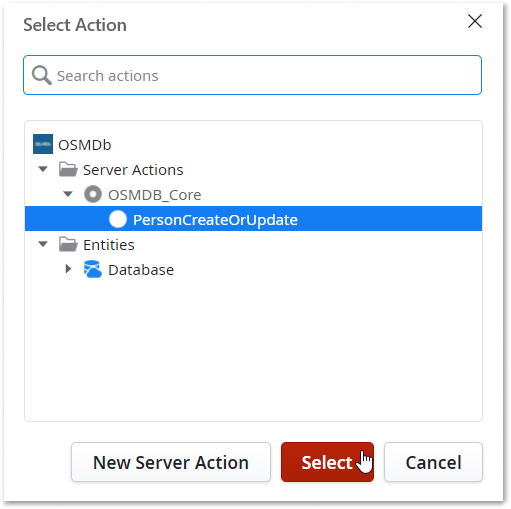
* 1. Arraste um Assign e solte depois da Action CreateOrUpdatePerson. Esse Assign deverá ser definido como parâmetro de saída com o valor de retorno da Entity Action



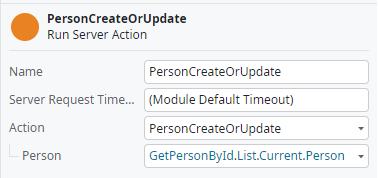
* 1. Publique o módulo Core.
  2. Volte para o módulo OSMDb e use a opção **Manage Dependencies** para referenciar as nova Action PersonCreateOrUpdate.
  3. Na Action **SaveOnClick**, abaixo da tela PeopleDetail, arraste o nó **Run Server Action** e solte depois do If.



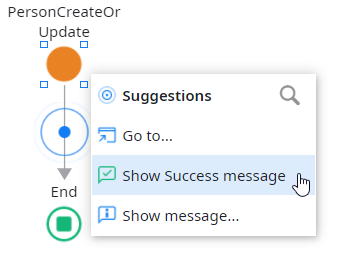
* 1. Na nova janela, escolha **PersonCreateOrUpdate**.



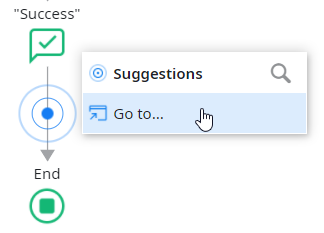
* 1. Defina o Parâmetro de entrada Person como a sugestão fornecida pelo Service Studio, que é o registro de pessoas sendo buscado do banco de dados



* 1. Passe o mouse sobre a última seta, clique no ícone azul e escolha **Show Success message**.



* 1. Defina a **Message** indicando que a operação foi feita com sucesso.
  2. Pela última vez, passe o mouse sobre a última seta e agora selecione **Go to...**



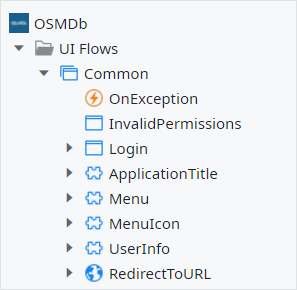
* 1. Selecione a tela **People** e clique em **Ok** para criar uma navegação para essa tela.

## Linkando as Telas

Assim como fizemos nos filmes, agora precisamos criar a navegação entre as telas People e PersonDetail.

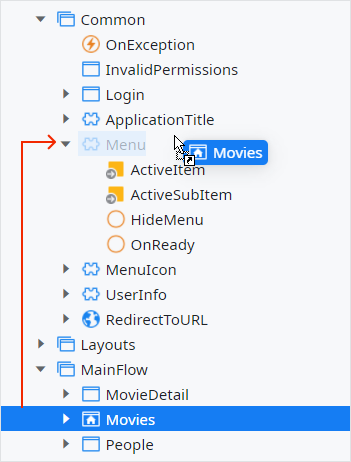
* 1. Volte para a tela **People** e crie dois Links para a tela **PersonDetail**
  2. Um Link por **Name** na tabela People
  3. Um Link *New Person* na área Screen Actions.

1. Na tela PersonDetail, crie um Link para retornar a tela People com o seguinte texto: *Back to People*
2. Publique a aplicação e teste no navegador.
3. Neste ponto, você não conseguirá acessar a tela People. Então, precisamos adicionar duas entradas no menu da aplicação: Movies e People.
4. No módulo OSMDb, abaixo da aba Interface. Expanda o Common UI Flow.



**NOTE:** Esse UI Flow tem telas em comum e blocos para a aplicação, incluindo a tela de Login e o Menu Block.

1. Para adicionar uma entrada ao Menu, arraste uma tela e solte sobre o Menu Block. Precisamos fazer isso para as telas Movies e People.



1. Publique a aplicação e teste no navegador.
2. Todas as telas da aplicação devem aparecer no menu no topo.

