Part 1

CHAPTER 12

課題 2
写真公開アプリに機能を追加する

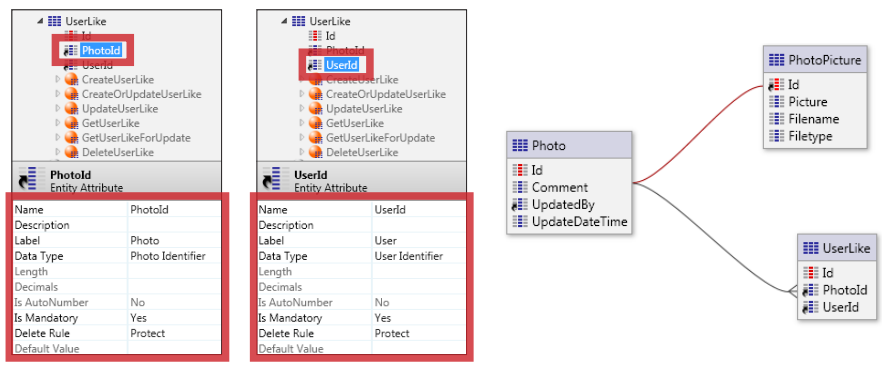
12.1. いいね機能を追加する

いいね機能を追加する

1枚の写真に対してどのユーザがいいねと言っているかを保持するためのデータモデルを追加します。

01 「UserLike」エンティティを作成し、アトリビュート「PhotoId」「UserId」を追加する。

01



メンテナンス性を高めるため、いいね機能はWebBlockで作成した後、「ShowPhoto」画面に配置するようにします。

02 InterfaceレイヤでMainFlowを右クリックして「Add WebBlock」からWebBlockを追加し、名前を「ShowLike」とする。

03 「ShowLike」WebBlockを右クリックし、Input Parameter「PhotoId」を追加する。

- PhotoIdは、DataType:Photo Idenftifier、Is Mandatory:Yes であることを確認します。

04 「ShowLike」WebBlockを右クリックし、Preparationアクションを追加する。

Preparationアクションでは、写真に対するいいねの総件数及び現ユーザが既にいいねをしているかどうかを取得する必要があります。今回はSQLを利用します。SQLではCount取得するようにするため、それを格納するOutput Structureの作成から行います。

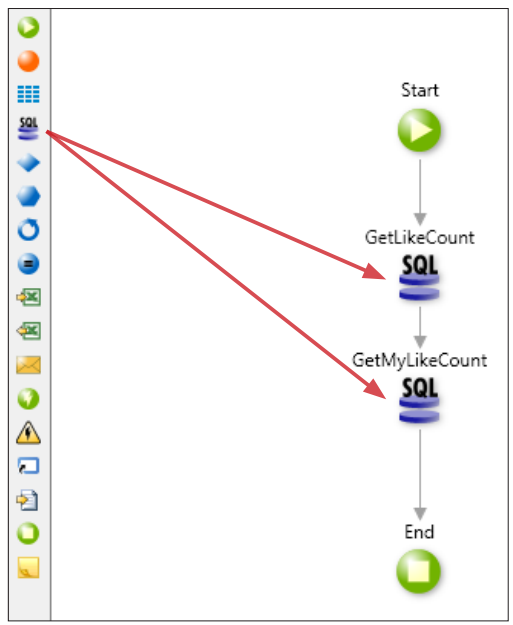
05 Dataレイヤを開き、「Structures」フォルダを右クリックして、「Add Structure」からStructureを追加して「Count」と名前を付ける。

06 「Count」Structureを右クリックし、StructureAttributeを追加して「Value」と名前をつける。

- 「Value」のData TypeはIntegerと設定します。

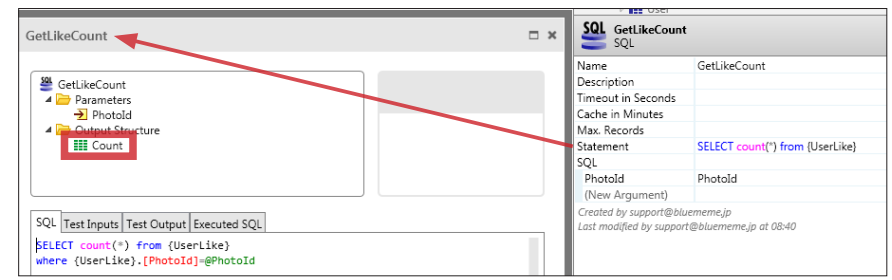
07 「ShowLike」WebBlockのPreparationアクションを開き、左のバーからSQLをドラッグしてStartとEndの間に配置する。

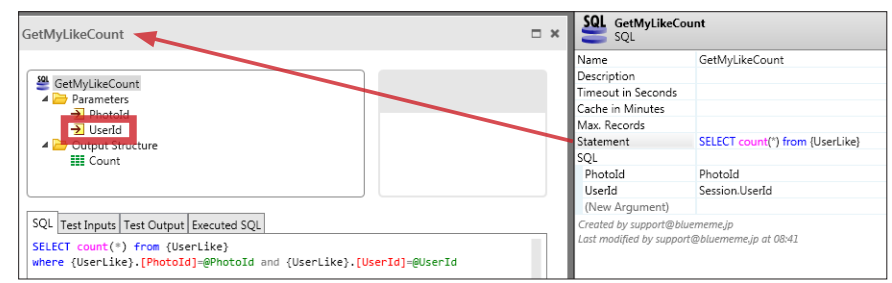
07



08 07を2回行い、それぞれ「GetLikeCount」と「GetMyLikeCount」と名付け、以下のように設定する。

08





09 「ShowLike」WebBlockをダブルクリックして開き、Containerを配置して「LikeArea」と名付ける。

10 09のContainerの中にExpressionを配置し、「GetLikeCount.List.Current.Count.Value」を設定する。

11 10に続き、「人がいいねと言っています」とテキストを入力する。

12 次に、いいねと言っていない場合には「Like」リンク、既に言っている場合には「Unlike」リンクを出すようにIfを用います。

13 ツールボックスから、「If」をドラッグし、画面にドロップする。

14 IfのConditionには「GetMyLikeCount.List.Current.Count.Value = 0」をセットする。

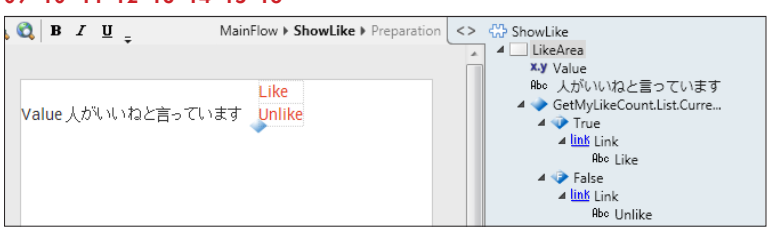
15 True領域にLinkを配置し、「Like」というテキストを入力する。

- LinkのMethodは「Ajax Submit」、Validationは「(none)」、Destinationは「(New Screen Action)」をセットする → Likeアクションが生成される。

16 同様にFalse領域にLinkを配置し、「Unlike」というテキストを入力する。

- LinkのMethodは「Ajax Submit」、Validationは「(none)」、Destinationは「(New Screen Action)」をセットする → Unlikeアクションが生成される。

09・10・11・12・13・14・15・16

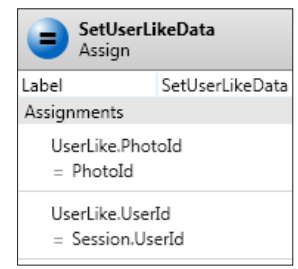


17 「Like」アクションを右クリックして、LocalVariable「UserLike」を追加する。

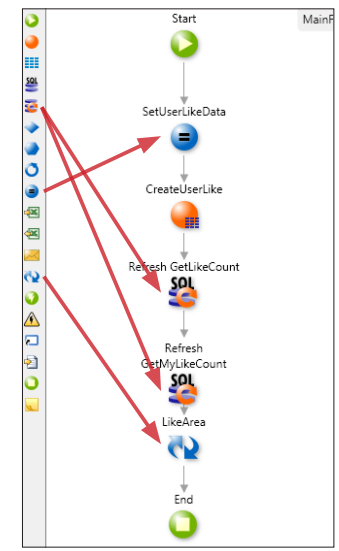
18 「Like」アクション開いて、右記のように配置する。

- Assignは、「SetUserLikeData」と名付けて、PhotoIdとUserIdをセットします。
- Dataレイヤを開き「UserLike」エンティティから「CreateUserLike」アクションをドラッグアンドドロップして、SourceにLocalVariable「UserLike」をセットします。
- Refresh Dataを2つドラッグアンドドロップし、「Select Data Source」画面で「GetLikeCount」「GetMyLikeCount」をそれぞれ選択します。
- 「Ajax Refresh」をドラッグアンドドロップし、「Select Widget or Web Block」画面で「Like Area」Containerを選択します。

17



18



19 「Unlike」アクション開く。

20 Dataレイヤを開き、「UserLike」エンティティをドラッグして、Startの次にドロップし、「GetMyLike」とリネームする。

21 「GetMyLike」をダブルクリックし、「Filters」をクリックした後、「Add Filter」をクリックして「UserLike.PhotoId = PhotoId and UserLike.UserId = Session.UserId」をセットする。

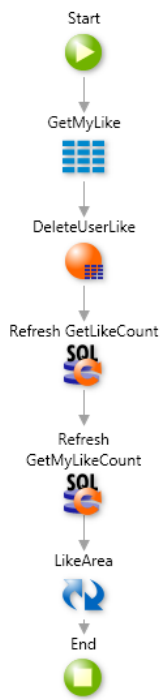
- 他に条件が入っていたらゴミ箱ボタンで削除します。

22 以降は、次ページのように配置する。

- Dataレイヤを開き「UserLike」エンティティから「Delete UserLike」アクションをドラッグアンドドロップして、Idに「GetMyLike.List.Current.UserLike.Id」をセットします。
- Refresh Dataを2つドラッグアンドドロップし、「Select Data Source」画面で「GetLikeCount」「GetMyLikeCount」をそれぞれ選択します。「Like」画面からコピーしてもよいです。
- 「Ajax Refresh」をドラッグアンドドロップし、「Select Widget or Web Block」画面で「LikeArea」Containerを選択します。「Like」画面からコピーしてもよいです。

23 「ShowPhoto」画面を開き既に配置済みのExpressionの下に「ShowLike」Web Blockをドラッグアンドドロップして、InputParameterの「PhotoId」をそのまま「Show Like」のInputParameterとしてセットする。

22・23



1-Click Publishを行って、アプリケーションの動作を確認する

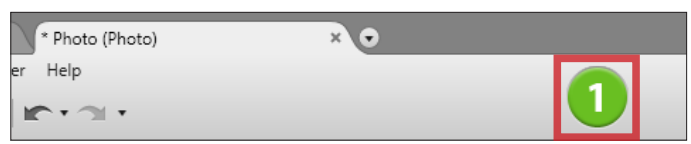
01 1-Click Publishボタンを押して、Publishを行う。

02 Open In Browserボタンを押して、ブラウザを起動する。

03 Photoアプリケーションのログイン画面が表示されたら、管理者ユーザでログインする。

04 アプリケーションの動作を確認する。

01



02

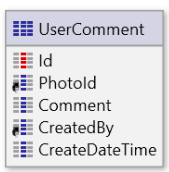
12.2. コメント機能を追加する

コメント機能を追加する

01 Dataレイヤで Entitiesを右クリックして「Add Entity」から Entityを追加し、名前を「User Comment」とする。

02 「UserComment」エンティティに以下のように Attributeを追加し、設定する。

01・02



Id
Entity Attribute

Name	Id
Description	
Label	Id
Data Type	Integer
Length	
Decimals	
Is AutoNumber	Yes
Is Mandatory	Yes
Delete Rule	Protect
Default Value	

PhotoId
Entity Attribute

Name	PhotoId
Description	
Label	Photo
Data Type	Photo Identifier
Length	
Decimals	
Is AutoNumber	No
Is Mandatory	Yes
Delete Rule	Protect
Default Value	

Comment
Entity Attribute

Name	Comment
Description	
Label	Comment
Data Type	Text
Length	50
Decimals	
Is AutoNumber	No
Is Mandatory	No
Delete Rule	Protect
Default Value	

CreatedBy
Entity Attribute

Name	CreatedBy
Description	
Label	Created By
Data Type	User Identifier
Length	
Decimals	
Is AutoNumber	No
Is Mandatory	No
Delete Rule	Protect
Default Value	

CreateDateTime
Entity Attribute

Name	CreateDateTime
Description	
Label	Create Date Time
Data Type	Date Time
Length	
Decimals	
Is AutoNumber	No
Is Mandatory	No
Delete Rule	Protect
Default Value	

メンテナンス性を高めるため、コメント機能もWebBlockで作成した後、「ShowPhoto」画面に配置するようにします。

03 Interfaceレイヤで MainFlowを右クリックして「Add WebBlock」から WebBlockを追加し、名前を「Show Comment」とする。

04 「ShowComment」WebBlockを右クリックし、Input Parameter「PhotoId」を追加する。

- PhotoIdは、DataType:Photo Idenftifier、Is Mandatory: Yesであることを確認します。

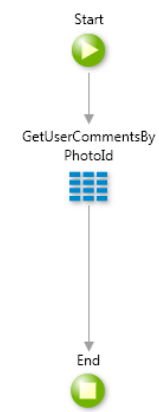
05 「ShowComment」WebBlockを右クリックし、Preparationアクションを追加する。

06 Preparationアクションを開き、Dataレイヤから「User Comment」エンティティをドラッグして、Startの下にドロップする。

07 「GetUserCommentsByPhotoId」とリネームする。

08 「GetUserCommentsByPhotoId」をダブルクリックして開く。

06・07

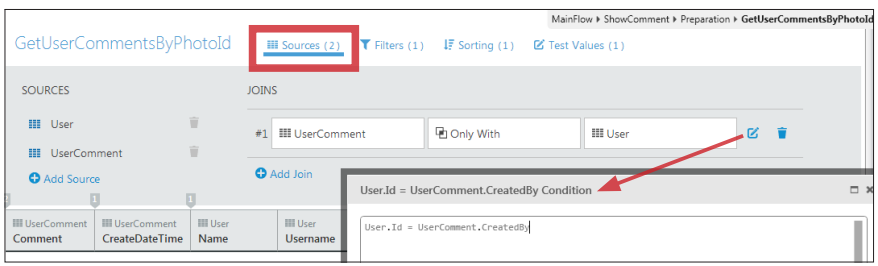


09 Sourcesをクリックして開き、以下のようにセットする。

10 Filtersをクリックして開き、「Add Fileter」を押して「UserComment.PhotoId = PhotoId」をセットする。

- 他にFilterができている場合は、削除します。

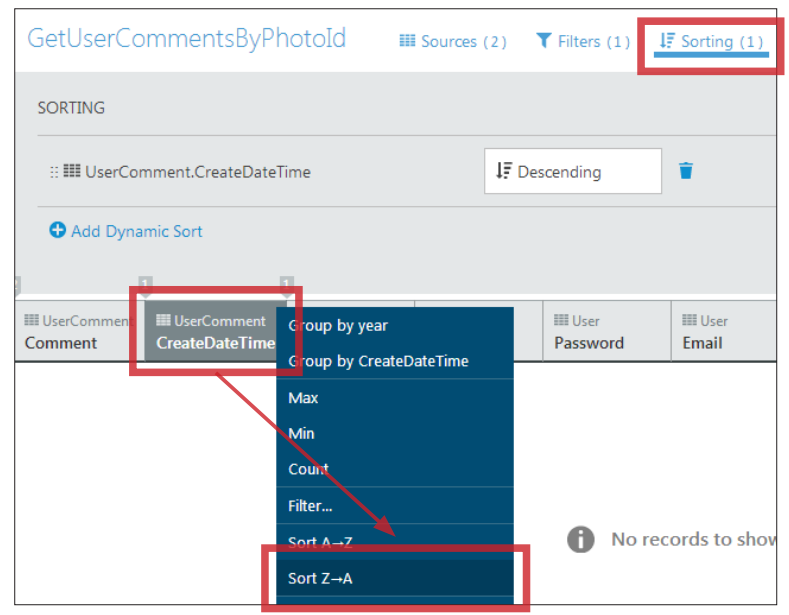
09・10



11 Sortingをクリックし、「CreateDateTime」右上の矢印から「Sort Z→A」を選択する。

- その他、Sort条件があれば削除し、以下の状態とします。

11



12 「ShowComment」を右クリックし、LocalVariable「UserCommentText」（DataType: Text）を追加する。

13 「ShowComment」WebBlockを開き、下図のように配置する。

- Containerを配置し、「CommentArea」と名付ける。
- TextInputを配置し、VariableプロパティにLocalVariable「UserCommentText」、MandatoryをTrueにセットします。
- Linkを配置し、「Send」と入力する。LinkのMethodは「Ajax Submit」、Destinationは「(New Screen Action)」を選択し、「Send」アクションができることを確認します。

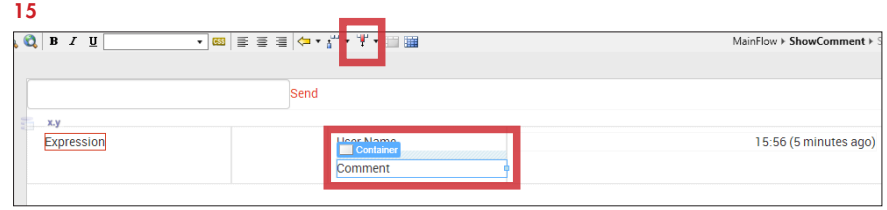
12

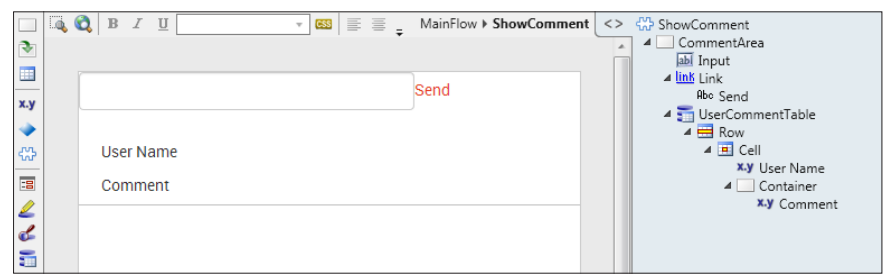
14 TableRecordsをドラッグして画面にドロップし、ShowHeaderプロパティを「No」に設定する。

15 Dataレイヤを開き、「UserComment」エンティティをドラッグして、「TableRecords1」の中にドロップする。

- Source Record Listプロパティには「GetUserCommentsByPhotoId.List」を設定します。
- 「UserName」のセルにContainerをドラッグアンドドロップで配置し、その中にCommentのExpressionをドラッグアンドドロップで移動させます。
- 「UserName」のカラム以外をDelete Columnで削除し、1列だけのテーブルにします。

15



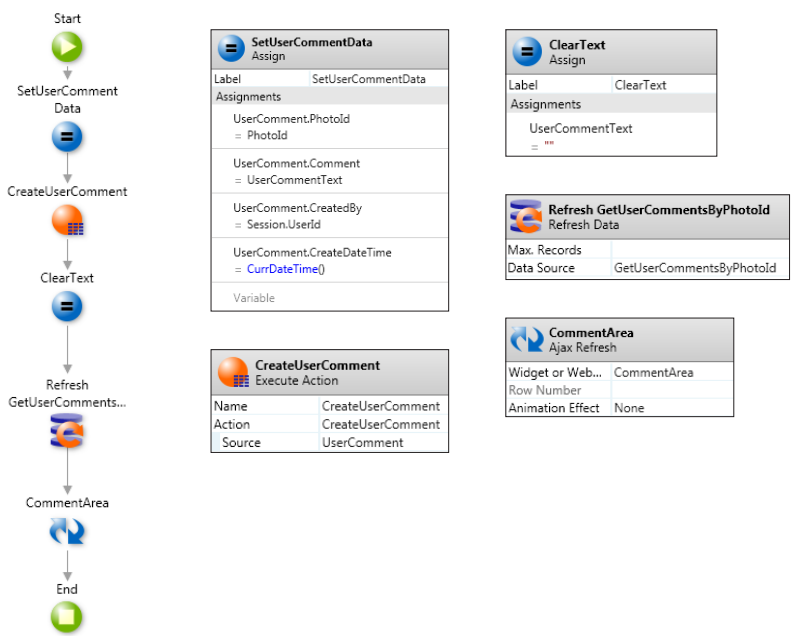


16 「Send」アクションを右クリックして、LocalVariable「UserComment」（DataType: UserComment）を追加する。

17 「Send」アクションを開き、下記のように配置及びセットする。

18 「ShowPhoto」画面を開き既に配置済みの「ShowLike」の下に「ShowComment」WebBlockをドラッグアンドドロップして、InputParameterの「PhotoId」をそのまま「ShowComment」のInputParameterとしてセットする。

17



1-Click Publishを行って、アプリケーションの動作を確認する

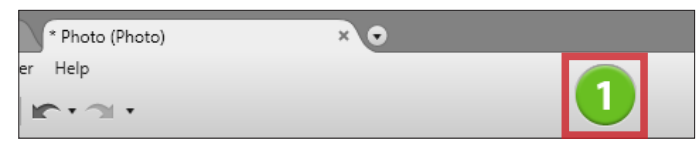
01 1-Click Publishボタンを押して、Publishを行う。

02 Open In Browserボタンを押して、ブラウザを起動する。

03 Photoアプリケーションのログイン画面が表示されたら、管理者ユーザでログインする。

04 アプリケーションの動作を確認する。

01



02

12.3. アクセス制御を行う

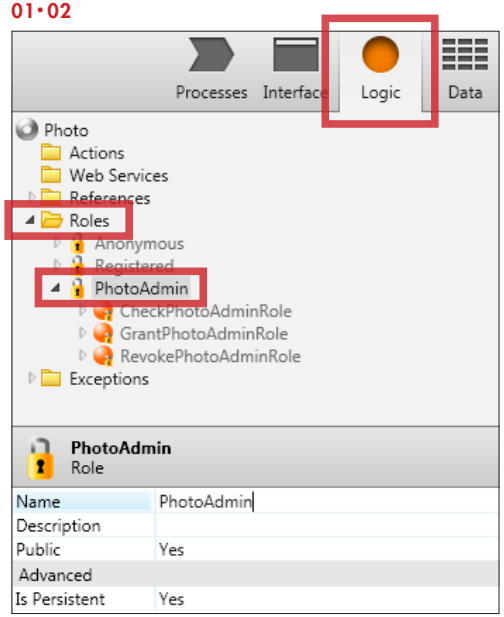
アクセス制御を行う

管理者のみが写真を追加できるようにします。

01 Logic レイヤを開き、「Roles」フォルダを開く。

02 既にできている「PhotoUser」ロールを「PhotoAdmin」にリネームする。

01・02

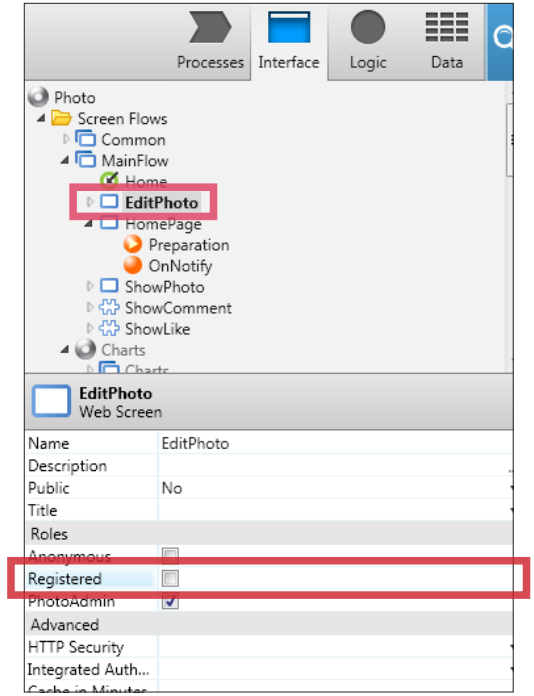


03 Interface レイヤを開き、「EditPhoto」画面を選択する。

04 プロパティの Roles > Registered に入っているチェックを外す。

- これにより、PhotoAdmin ロールを持つユーザ以外は、EditPhoto 画面にアクセスできないようになります。

03・04



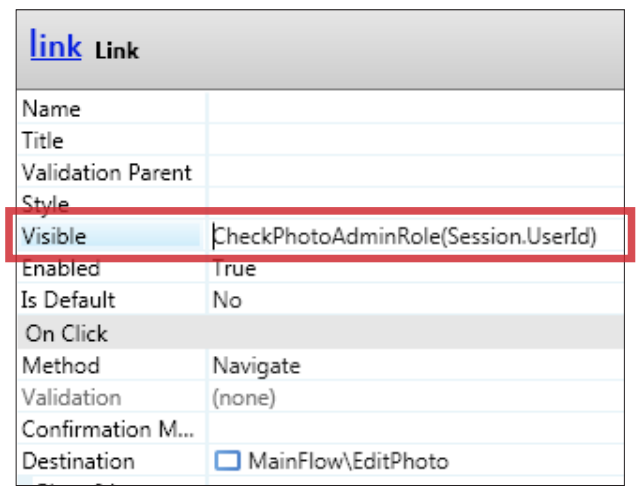
「EditPhoto」画面にアクセスはできないものの、「HomePage」画面にはリンクが見えてしまっているので、このリンクも管理者にしか表示しないようにします。

05 「HomePage」画面を開く。

06 「EditPhoto」画面へのリンクを選択し、Visibleプロパティに「CheckPhotoAdminRole(Session.UserId)」をセットする。

- これにより、PhotoAdminロールを持つユーザ以外には本リンクは見えなくなります。

06



1-Click Publishを行って、アプリケーションの動作を確認する

01 1-Click Publishボタンを押して、Publishを行う。

02 http://<host名>/Users にアクセスし、管理者権限でログインする。

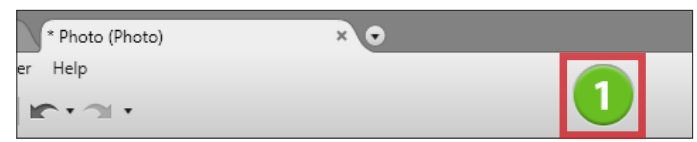
03 任意のユーザを選択し、詳細画面にてRolesからPhotoAdminロールをAddする。

04 Open In Browserボタンを押して、ブラウザを起動する。

05 Photoアプリケーションのログイン画面が表示されたら、管理者ユーザでログインする。

06 アプリケーションの動作を確認する。

01



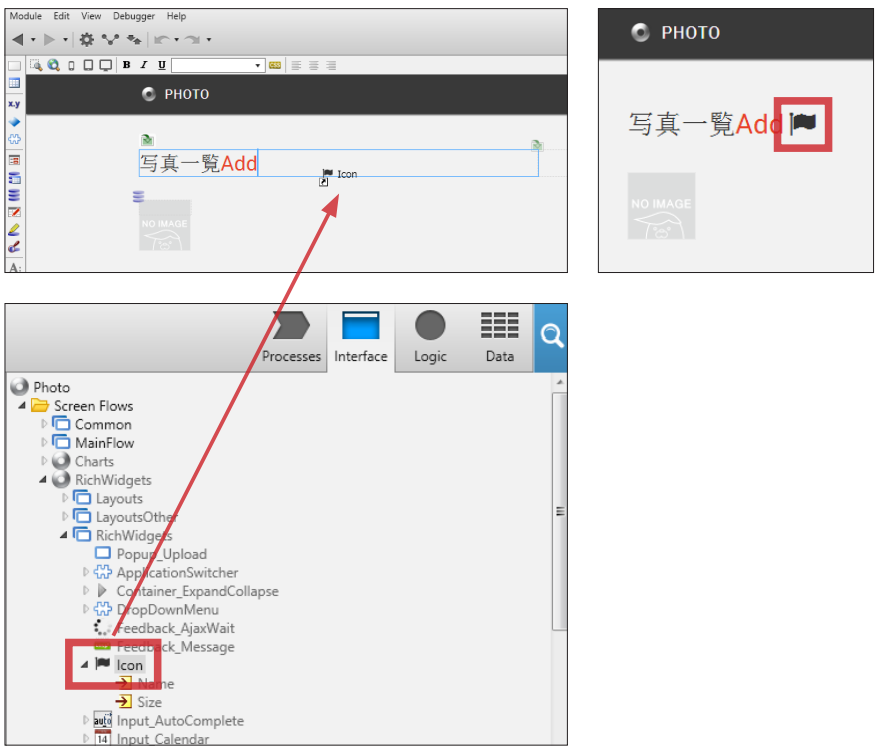
02

12.4. 標準のアイコンを利用する

標準のアイコンを利用する

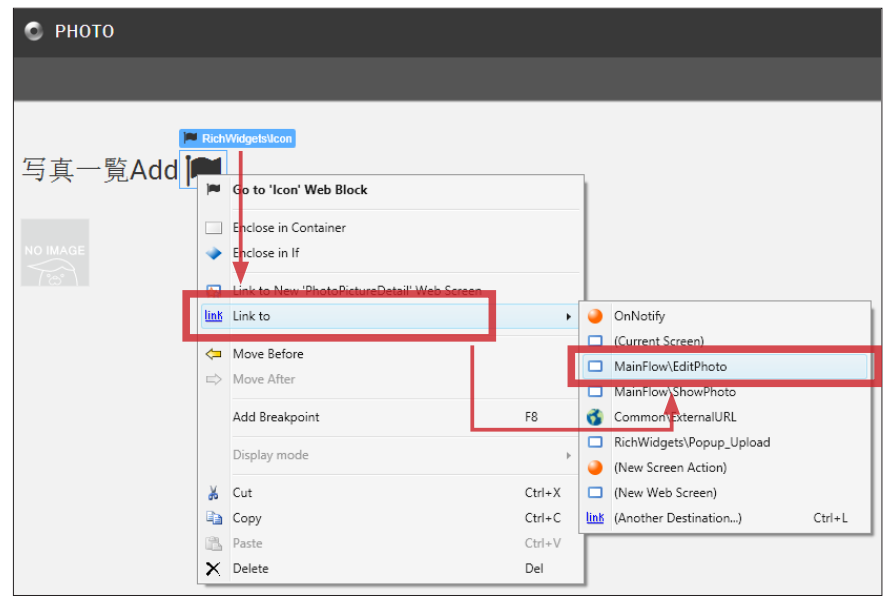
01 「HomePage」画面を開き、Interfaceレイヤの「Screen Flows」>「RichWidgets」>「RichWidgets」>「icon」をドラッグし、「Add」リンクの右に配置する。

01



02 配置したicon上で右クリックし、「Link to」から「MainFlow\EditPhoto」を選択する。

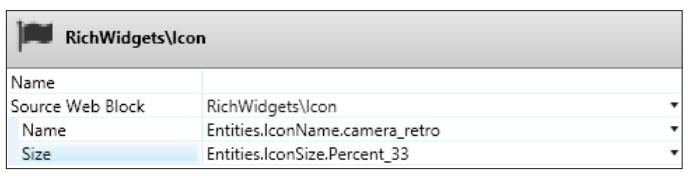
02



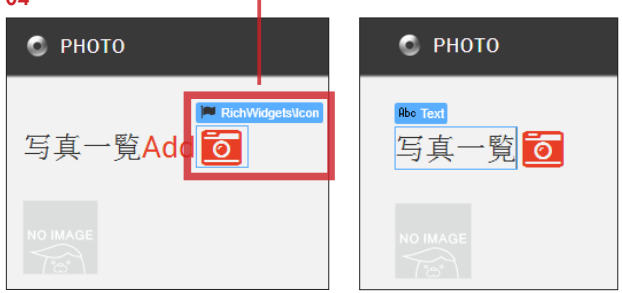
03 配置したiconのプロパティを下記の通り設定する。

04 Addリンクを削除する。

03



04



1-Click Publishを行って、アプリケーションの動作を確認する

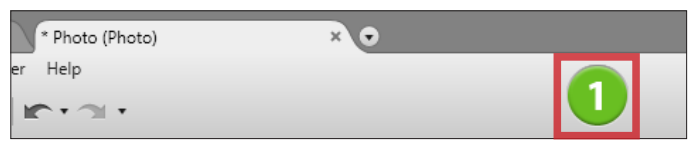
01 1-Click Publishボタンを押して、Publishを行う。

02 Open In Browserボタンを押して、ブラウザを起動する。

03 Photoアプリケーションのログイン画面が表示されたら、管理者ユーザでログインする。

04 アプリケーションの動作を確認する。

01



02

04