グローバルイベント

同じ画面のコンポーネント間のやり取り

GeneXus™

GeneXus

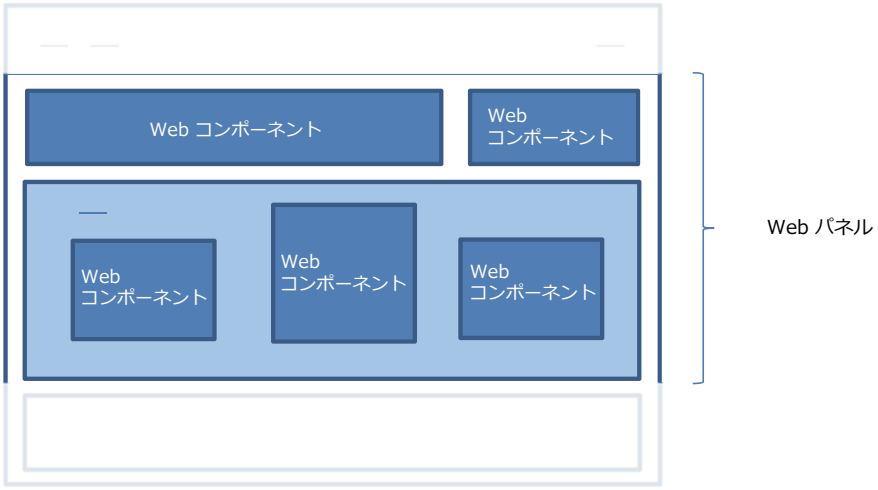
多数の画面から構成される画面

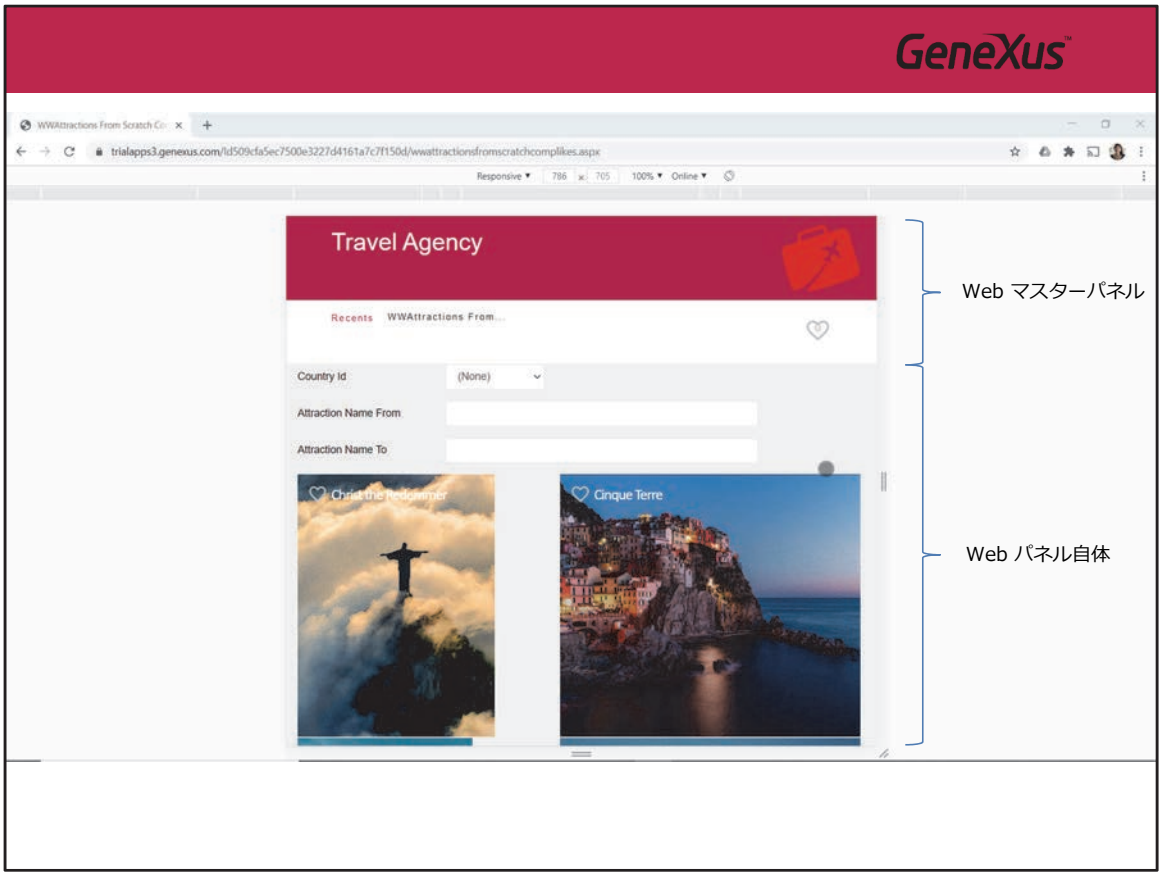
一般に、再利用のためには、できる限りコンポーネント化を進めます。

このため、通常はどの画面も、1 つではなく複数のオブジェクトから構成され、
その間でやり取りがあります。

GeneXus

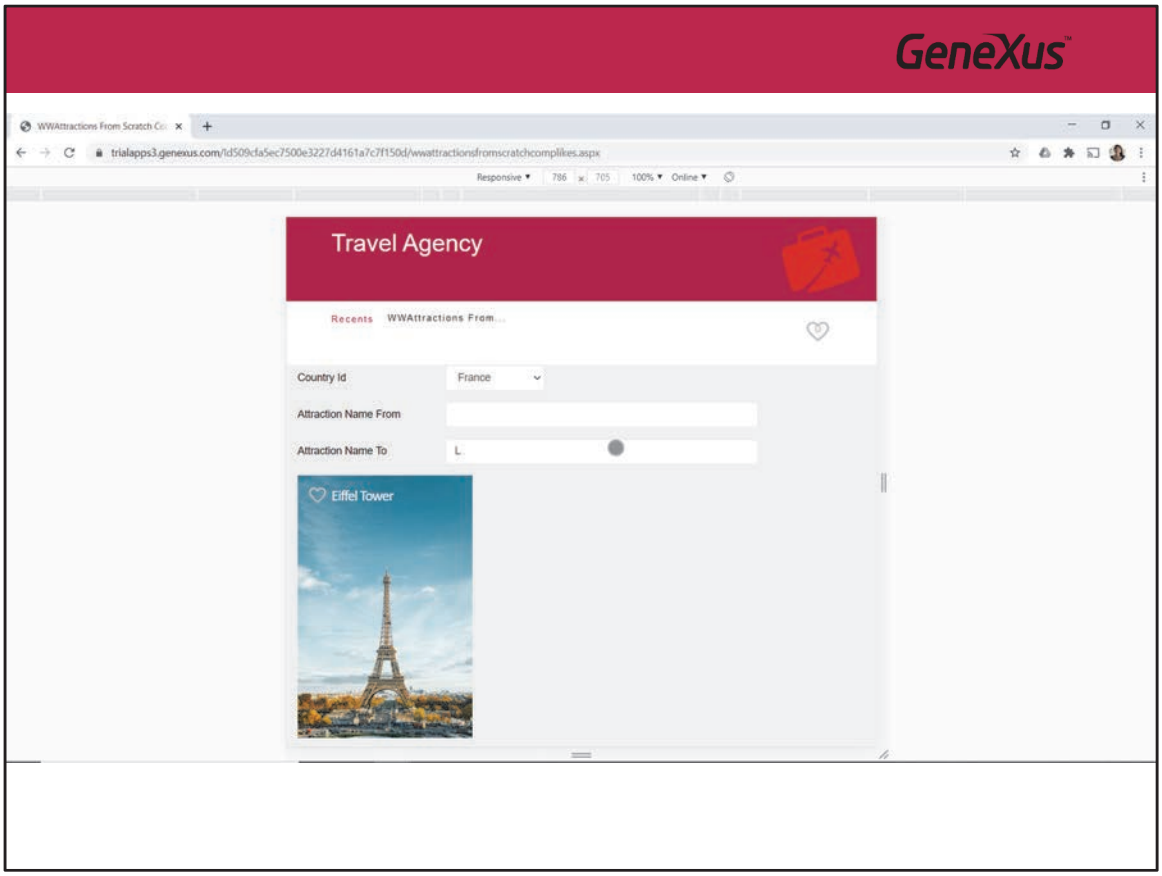
多数の画面から構成される画面



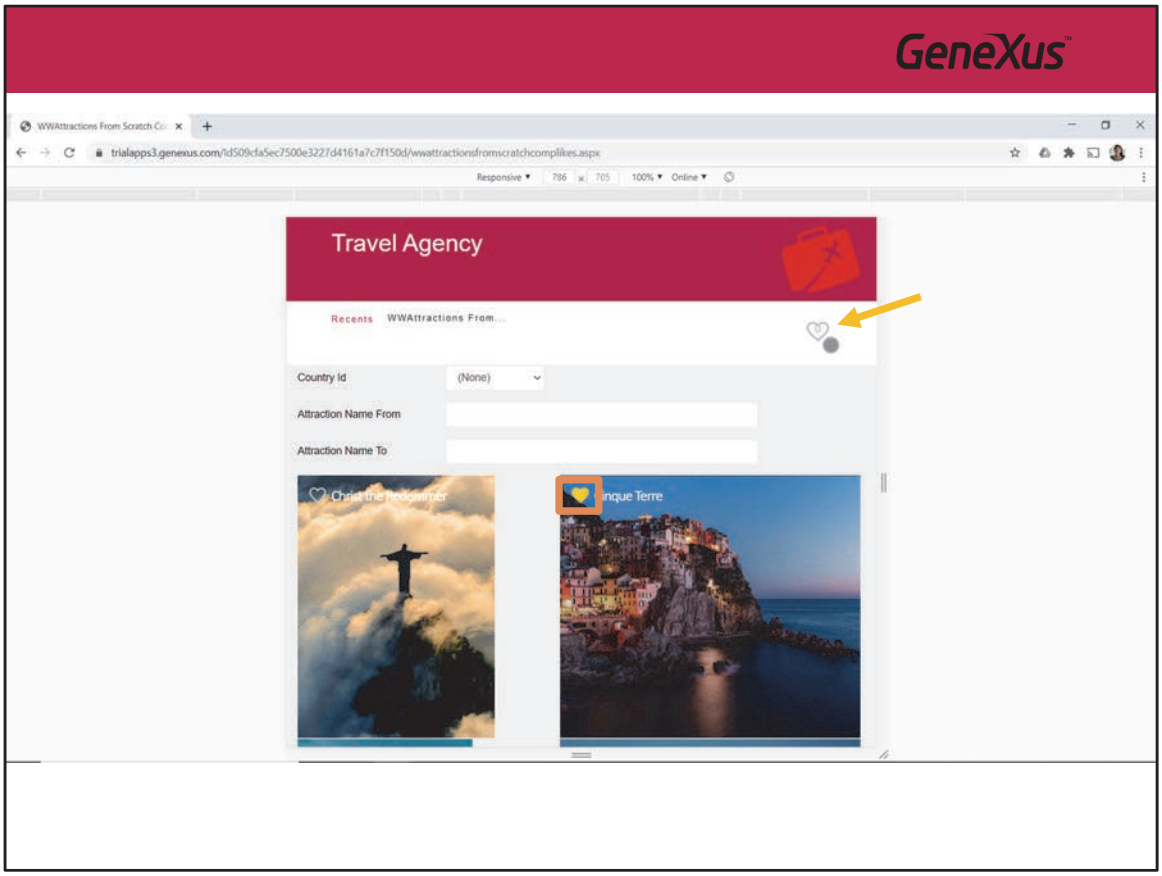


GeneXus の Basic コースで見たものと同様の Web パネルを例として示します。旅行代理店が提案する観光名所のリストです。デザインは大まかなものであり、さらに作業が必要ですが、この点についてここでは扱いません。

ここに示す画面は、それほど複雑ではありません。マスターパネルを踏襲した部分と、独自の部分があります。後者の部分は一部がコンポーネントとして実装されていますが、このことは実行時にはわかりません。



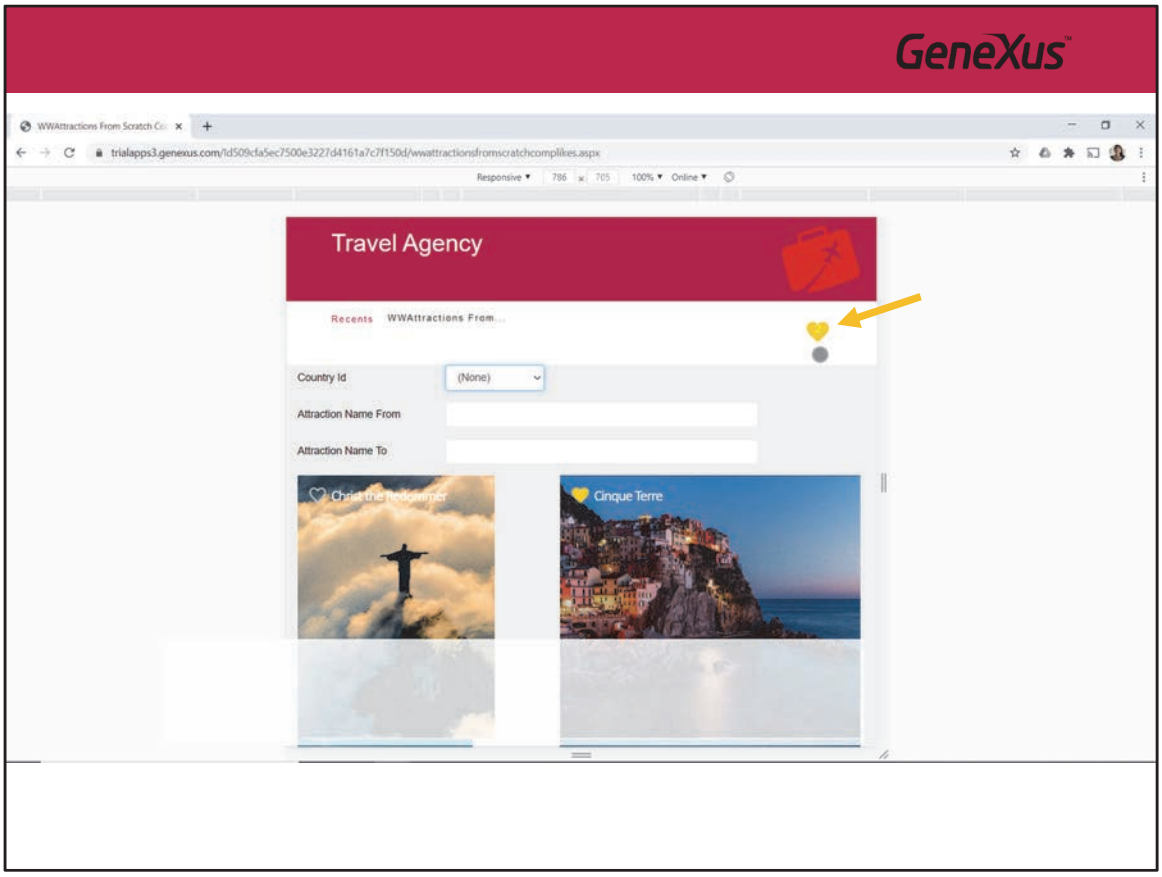
国名や観光名所名によるフィルタリングが可能です。



この章のトピックに関連して注目すべき点は、特定の観光名所をお気に入りとして選択できるようにしたことです。

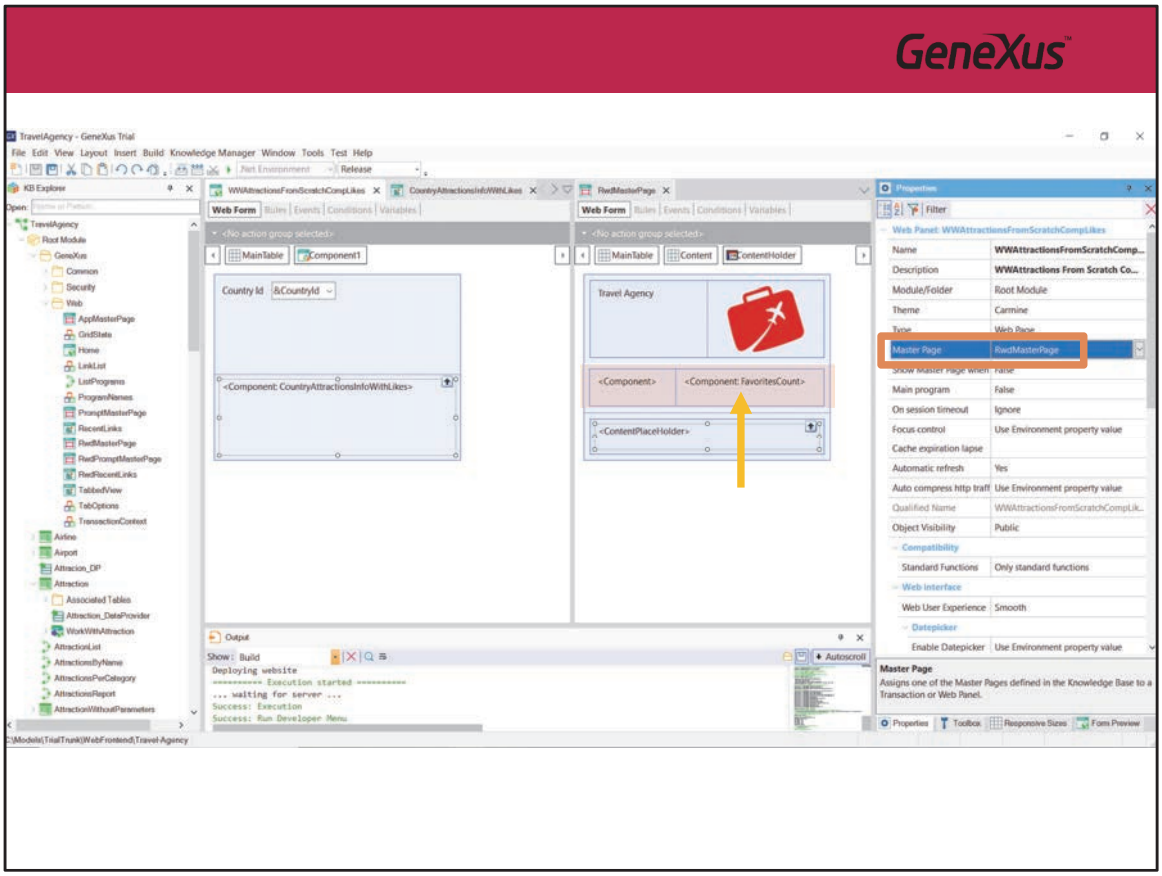
試しにいくつか選択します。これでお気に入りの観光名所を 2 つ選んだことになります。

しかし、ここの表示はゼロであり、選択内容が反映されていません。



画面を再表示すると、数値が 2 に変わりました。お気に入りを 1 つ削除すると、再表示したときにこの数値が 1 になるはずです。

観光名所をお気に入りとして選択するか、選択を解除したときに、値を自動で更新するにはどうすればいいかをこれから考えていきます。

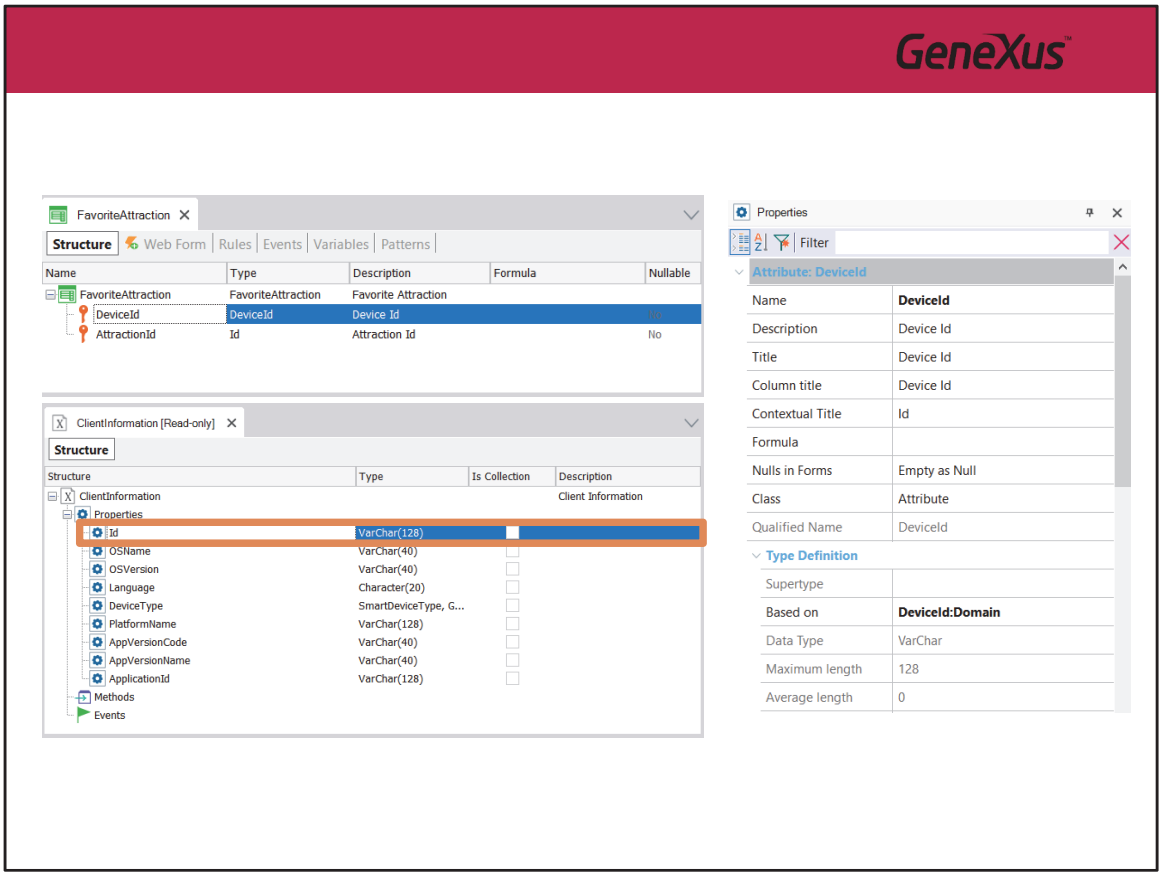


この Web パネルは、GeneXus ではこのようになっています。

最初に、対応するマスターページがあることがわかります。これが既定です。したがって、画面のコンテンツはマスターページの ContentPlaceHolder にロードされます。

そして、その外にほかのものが表示されます。

具体的には 2 つのコンポーネントがあります。最近のリンクを実装する既定のコンポーネントと、ユーザーがお気に入りとして選択した観光名所の数を示すために追加した別のコンポーネントです。

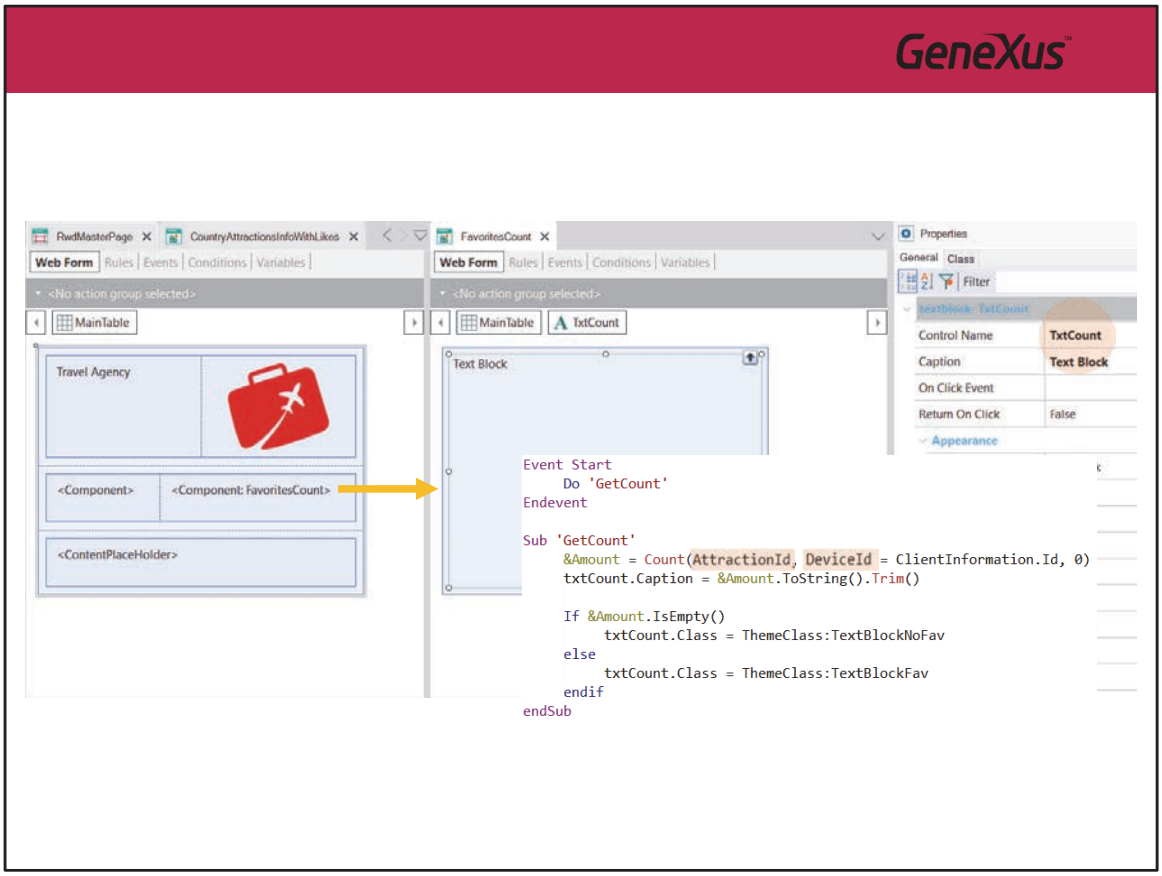


ユーザーがお気に入りとして選択した観光名所を保存するには、各ユーザーのお気に入りの観光名所を示す Users テーブルが必要です。まだユーザーを扱わない場合(たとえば、まだ GAM を適用しておらず、独自の Users テーブルを使用するかどうかを検討中の場合)、お気に入りはブラウザーインスタンスごとに一時的に保持できます。

そのために、このトランザクションを作成しました。項目属性には、新規に作成したこのドメインを指定しています。データタイプはこのプロパティと同じです。

ClientInfomation は GeneXus モジュールの外部オブジェクトであり、その [Id] プロパティにより、ネイティブか Web かを問わず、実行中のクライアントデバイスを特定できます。

Web アプリケーションの場合、このプロパティはユーザー ID を返します。これは同じブラウザーと同じアプリケーションにおけるすべてのセッション間で維持されます。そこで、キーがデバイスと観光名所から構成されるトランザクションを追加しました。このトランザクションのテーブルにお気に入りを保存します。

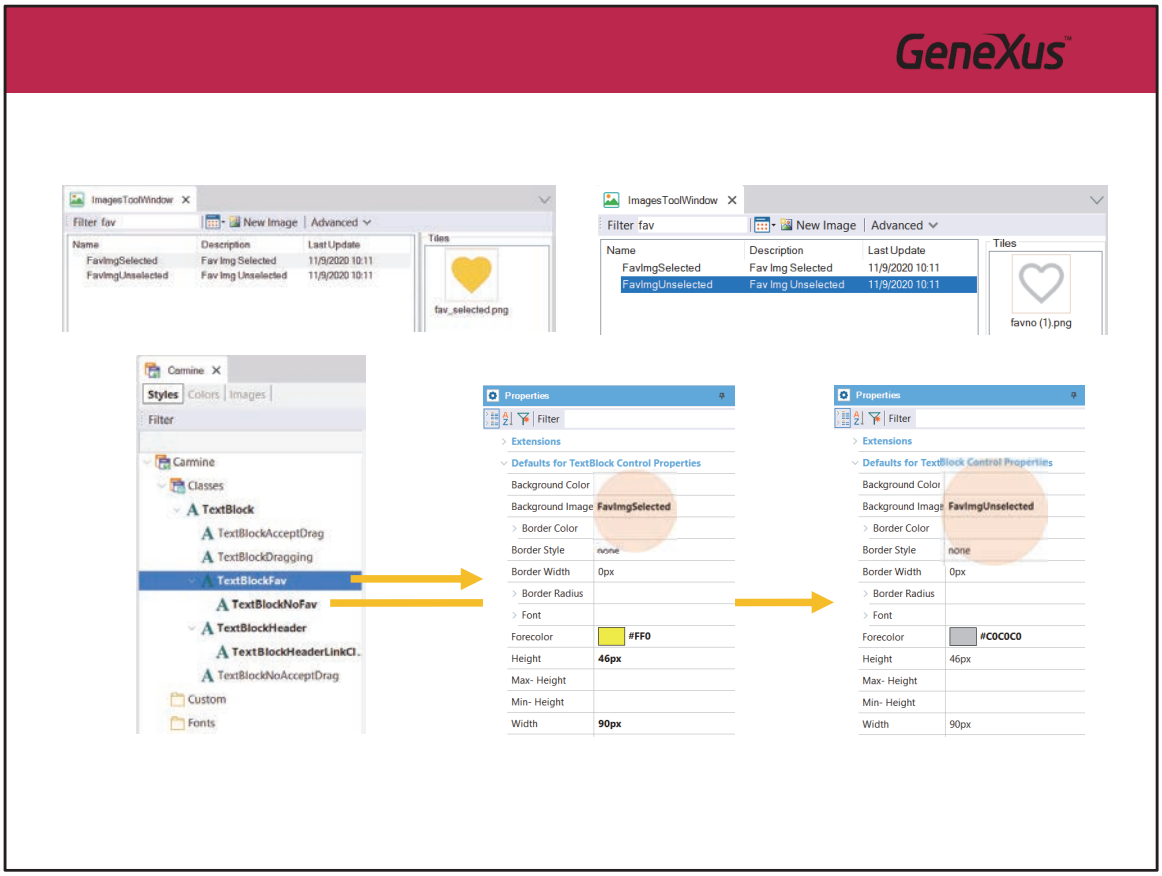


Web のマスターページで、ここにロードされる Web コンポーネントを確認すると、この名前のテキストブロックのみが含まれることがわかります。キャプションは実行時に、最初は Start イベントでのみロードされます。ここで、このサブルーチンを起動し、最初にこのクライアントの観光名所数を計算しています。それは、この 2 つの項目属性が含まれるテーブル (FavoriteAttraction) にあります。

これが必要な値です。この数値をその後、文字列に変換し、表示するテキストブロックに割り当てます。

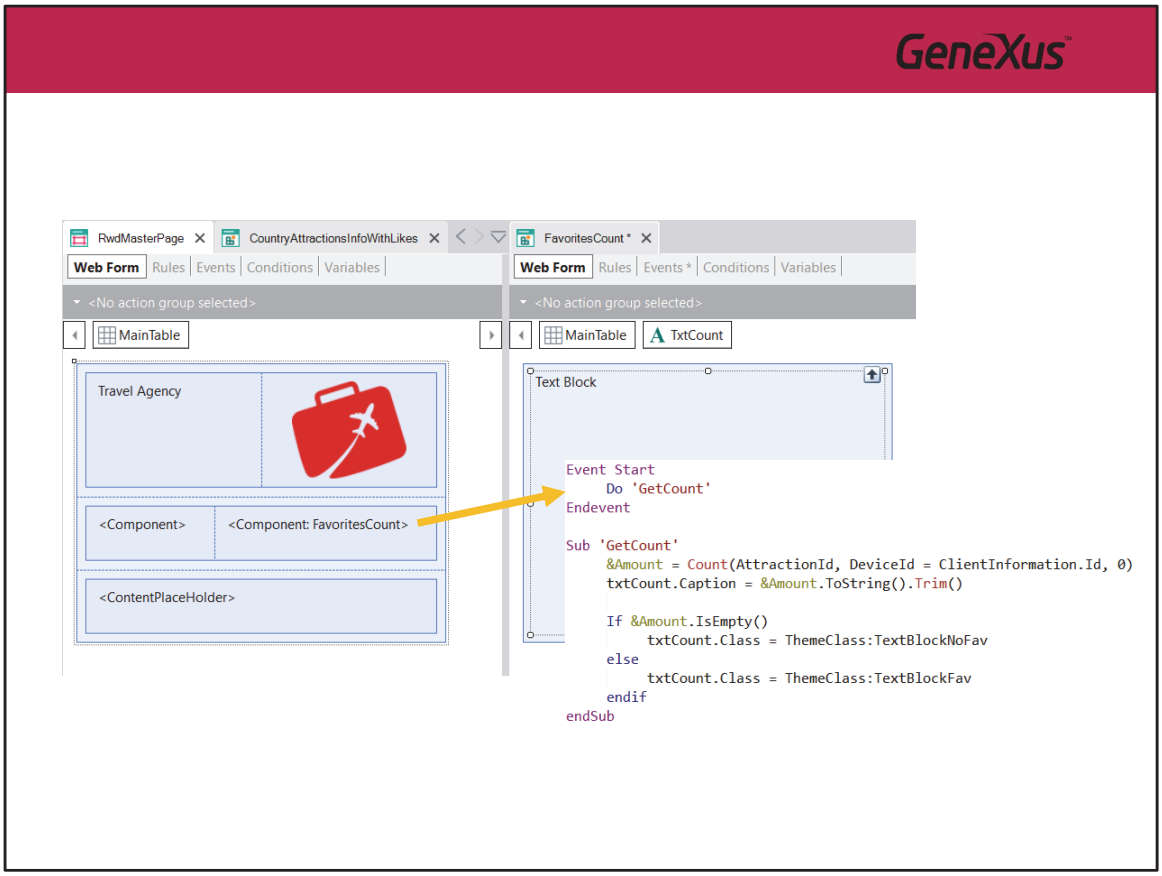
お気に入りの画像と適切な色および書式で表示するために、対応するテーマ、この場合は Carmine に 2 つのクラスを追加しています。

観光名所数がゼロの場合は、お気に入りがない場合のデザインに対応するクラスをテキストブロックに割り当てます。それ以外の場合は、黄色の塗りつぶしのデザインを割り当てます。



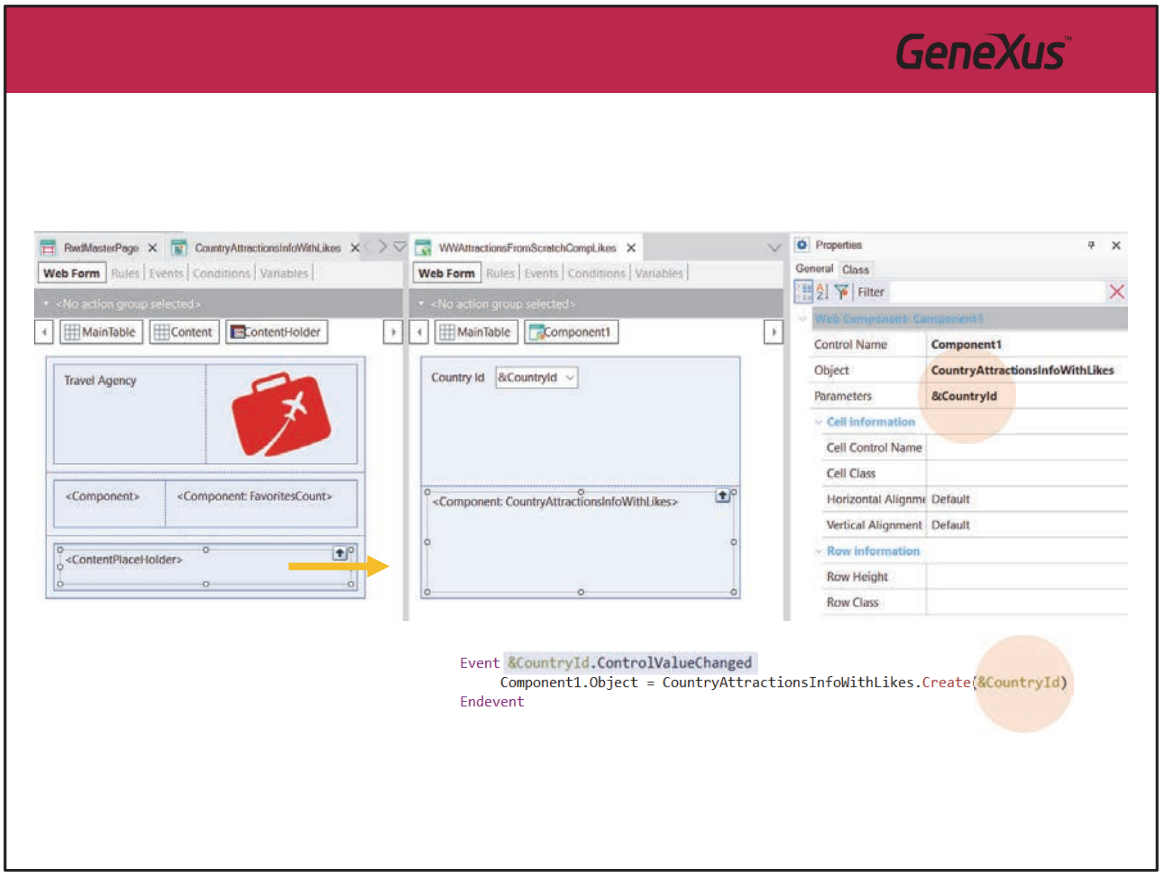
そのために、事前に 2 つの画像をナレッジベースに挿入する必要がありました。

テーマ内のクラスを確認すると、[Background Image] プロパティが黄色の塗りつぶしのハートとグレーの線で描かれたハートになっていることがわかります。

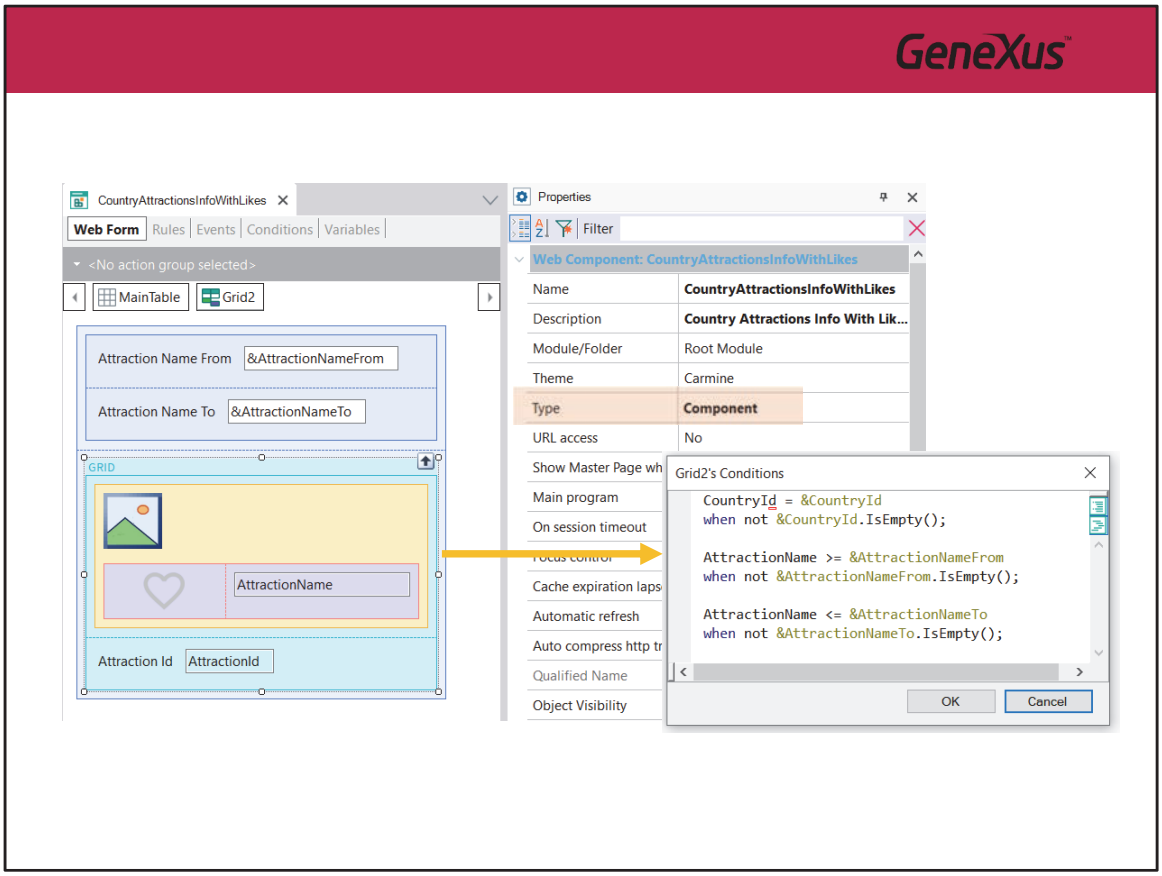


このように、このテキストブロックでは数を計算し、適切な背景画像および色で表示します。

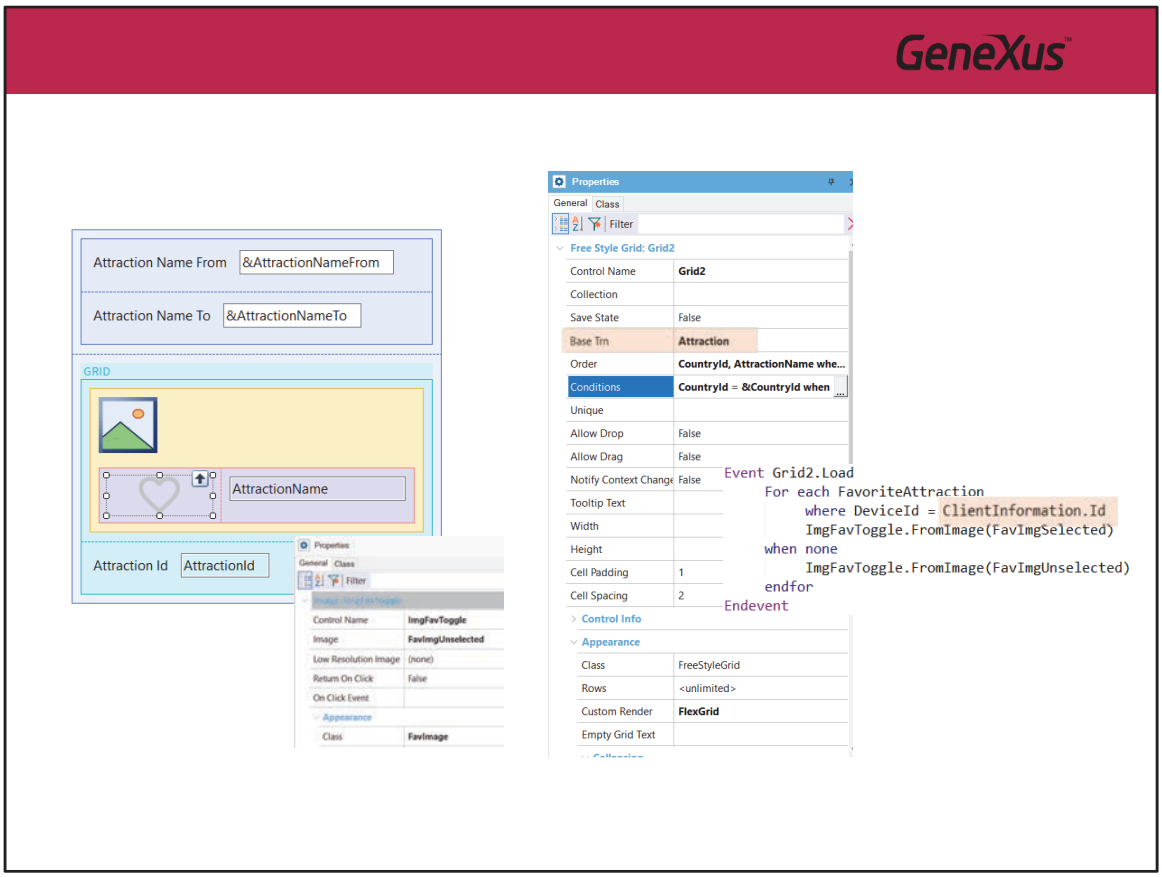
この Start イベントは、マスターページが開くときにトリガーされます。



一方で、Web パネルの ContentPlaceHolder にロードされる内容を確認すると、国を選択するためのコンボボックスがある画面になっています。その下には別のコンポーネントがあり、国の値が変更されるたびにここに送信されるようになっています。

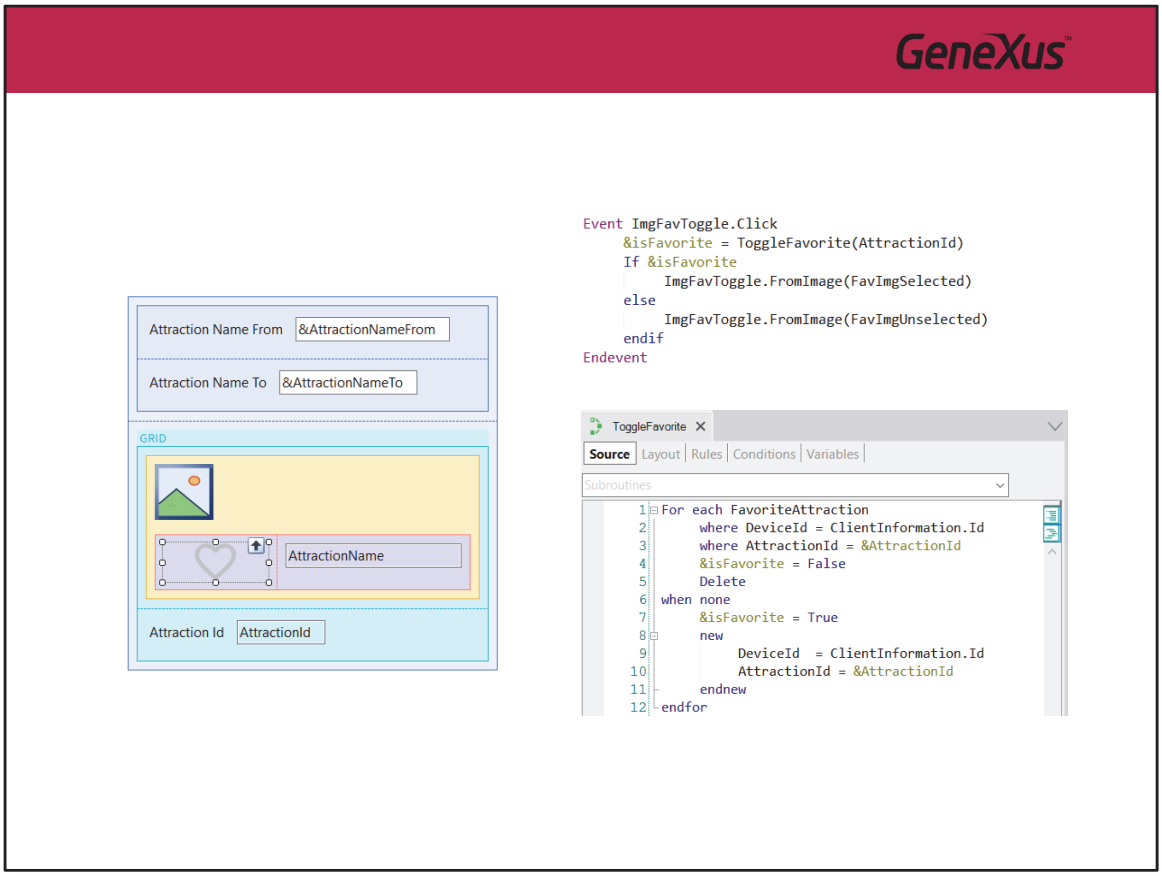


この Web コンポーネントにロードされるのは、コンポーネントタイプの別の Web オブジェクトです。ここには、パラメーターとして受け取る国と画面上のフィルタに従い表示される観光名所がフィルタリングされるグリッドがあります。



キャンバステーブルを含む Flex グリッドを選択し、観光名所の写真に、その名前とユーザーがその名所をお気に入りとして選択または選択解除するための画像を重ねて表示できるようにしました。
デザインに関する詳しい説明は省きます。

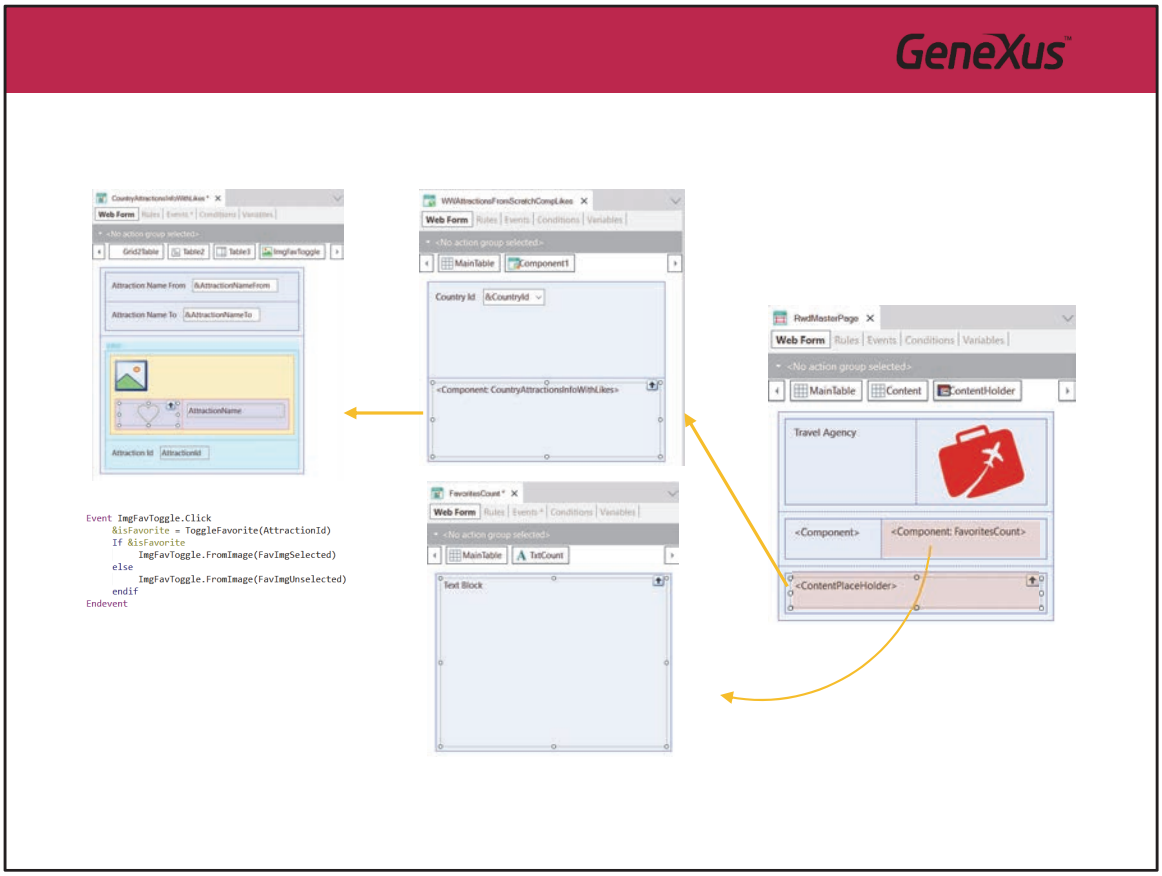
Attraction をベーステーブルとするグリッドの Load イベントで、ロードする観光名所ごとにこの For Each コマンドが実行され、このコマンドにより従属テーブル FavoriteAttraction で、実行中のデバイスのレコードが存在するかどうかが検索されます。存在する場合、その観光名所はお気に入りであるため、塗りつぶしのハートの画像がロードされます。存在しない場合は、塗りつぶしのないハートがロードされます。



ただし、この中で本当に重要なのは別のことです。それは、この画像の Click イベントのプログラミングです。

観光名所のグリッドがロードされたあと、ユーザーが観光名所をお気に入りとしてクリックすると、その名所が既にお気に入りだったかどうかを確認するプロシージャーが呼び出されます。既にお気に入りだった場合は、テーブルから削除し、画像を塗りつぶしのないほうに置き換えるよう指定します。逆の場合は、レコードを挿入します。

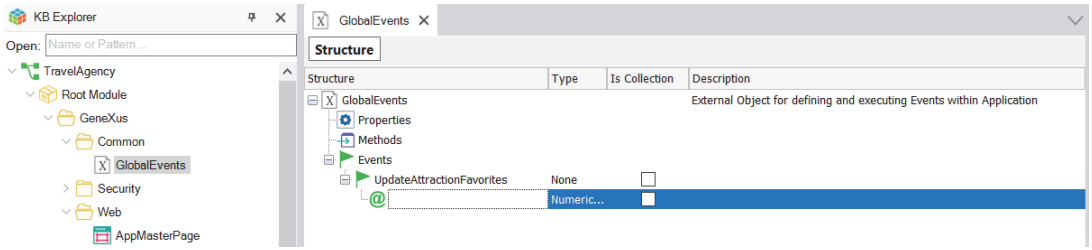
この処理により、お気に入りの合計数が更新されるはずですが、実際には更新されません。



簡単に言えば、このイベントはこの Web パネル内の Web コンポーネントでトリガーされます。そしてこの Web パネルは、この Web マスターパネル内で実行されます。そして、再表示するコンポーネントもここに作成しています。

言い換えると、このパネルにロードされるコンポーネントのユーザーイベントに、別のコンポーネントに対するアクションを実行させる必要があります。そのコンポーネントとは、同じ画面上で間接的に関わりがあること以外に関係はありません。

グローバルイベント: 既定の外部オブジェクト



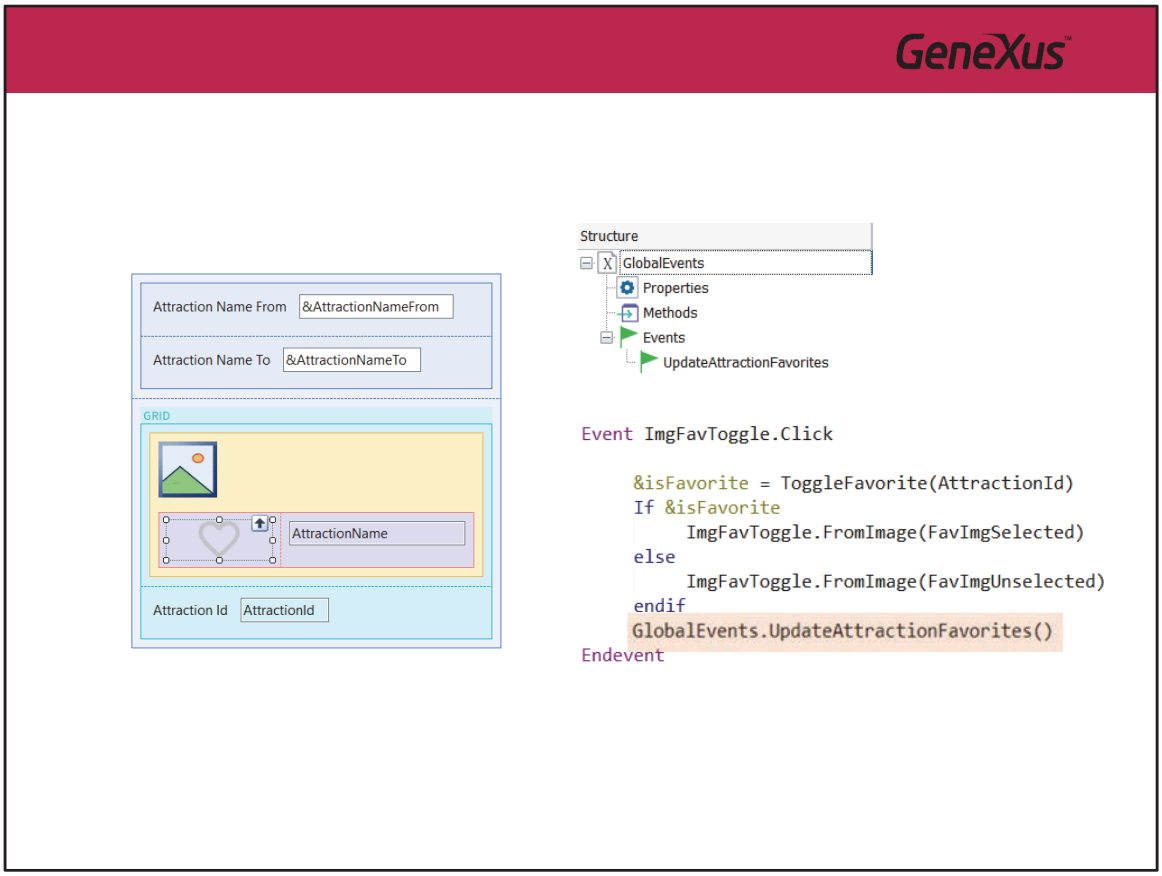
このやり取りを可能にするため、すべてのナレッジベースには**グローバルイベント**という事前定義のオブジェクトがあり、変更できるようになっています。

グローバルイベントは別名で保存でき、必要な数だけ固有のオブジェクトを作成できます。

この例では、事前定義のものを変更します。最初は空の状態なので、このように呼び出すイベントを追加します。

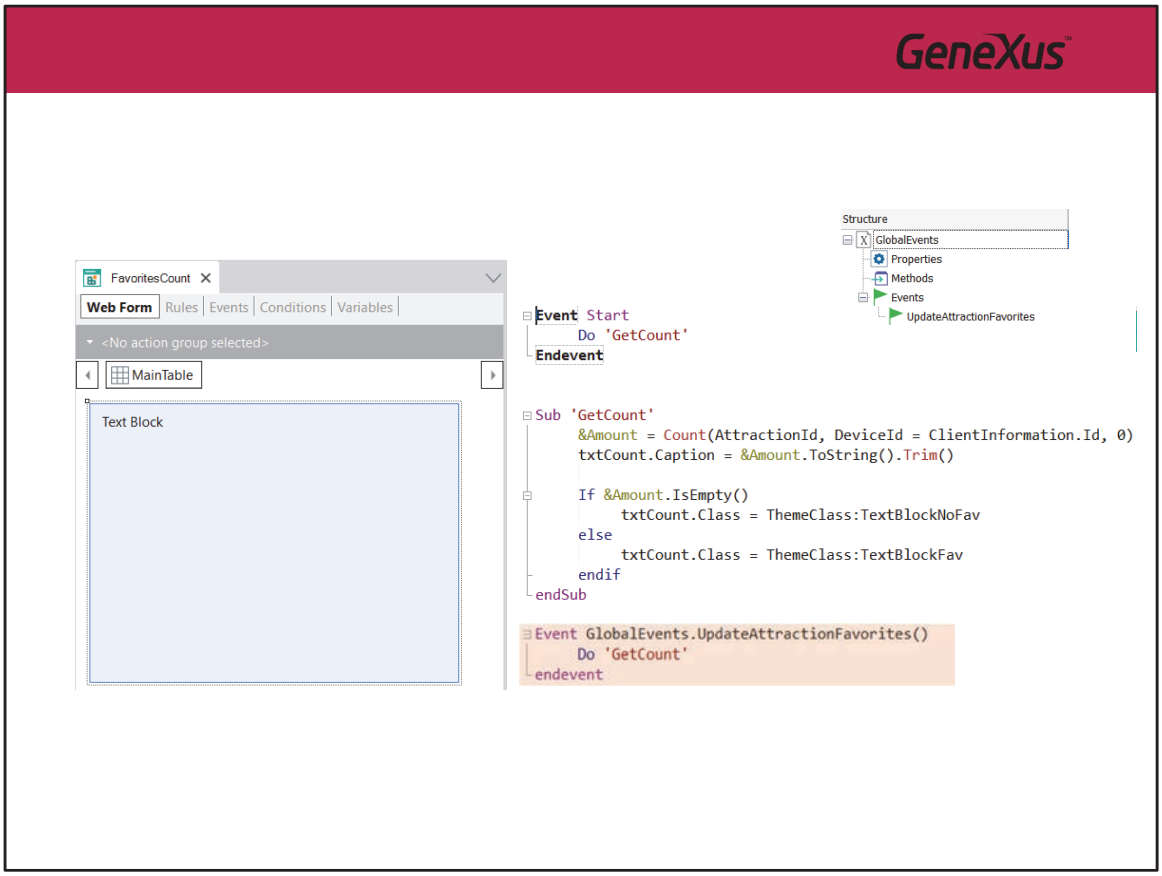
コンポーネント間のやり取りに必要な場合は、追加するイベントでパラメーターを扱うことができます。

ここでは、別のイベントからトリガーされたことがわかれば十分です。パラメーターは必要ありません。



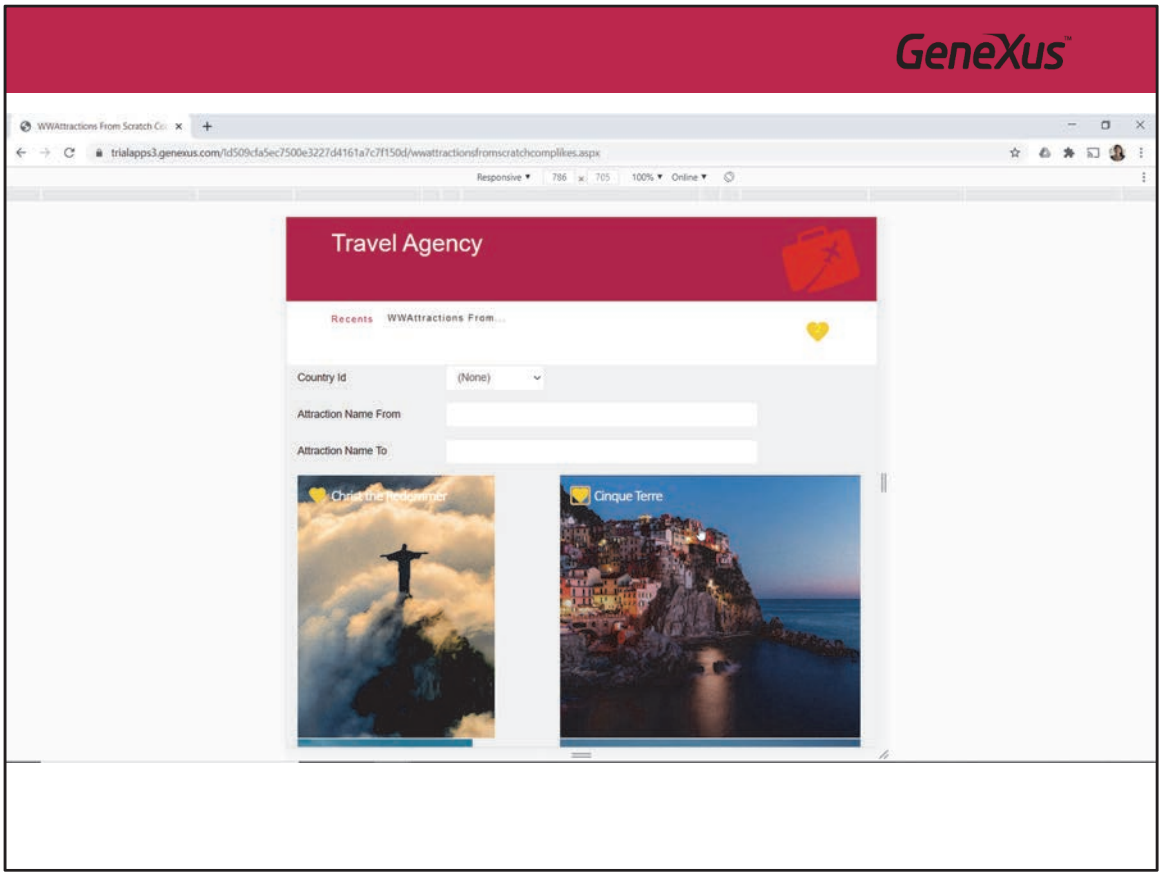
このグローバルなやり取りの手段を実際に使用してみましょう。

イベントをトリガーするパネルに移ります。トリガーは、画像がクリックされるたびに発生します。そのときに、GlobalEvents 外部オブジェクトを呼び出します。この段階では、GlobalEvents 外部オブジェクトは 1 つしかありません。この方法で、イベントのトリガーが通知されます。

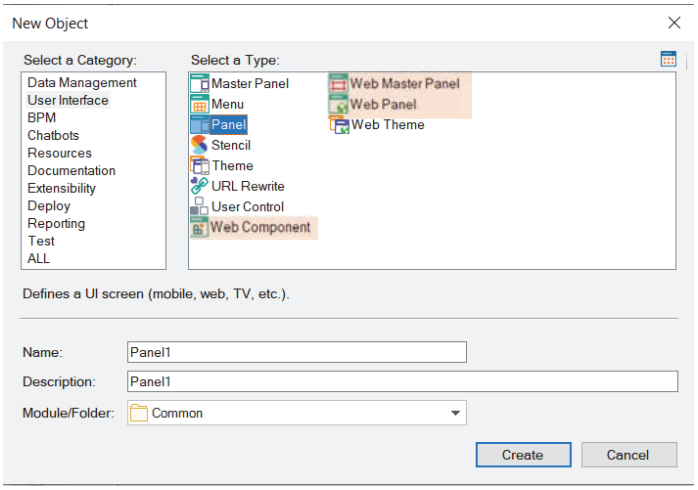


次に、観光名所を数えて表示するコンポーネントにこのイベントをサブスクライブさせます。こうすれば、リスナーの画面でイベントが発生するたびにリッスンできます。

これについては、単にイベントをリッスンするよう設定します。そしてここで、イベント発生時に実行する処理をプログラミングします。この例では、観光名所数を計算し直します。



試してみましょう。選択を解除し、再表示されることを確認します。クリックします。やり取りは正常に行われています。



ここでは、コンポーネントを含む Web パネル間のやり取りの例を見てきましたが、コンポーネントを含むマルチエクスペリエンスパネル (ネイティブアプリケーションの場合など) のやり取りも同様です。

このトピックの詳細については、Wiki をご覧ください。