

Web パネルのグリッドでの SDT コレクションのロード



さまざまな状況で、データコレクションのコンテンツを表示する必要があります。
Web パネルはそれを行うための方法の 1 つです。

SDT コレクションを Web パネルのグリッドでロードし、行に対して特定のアクションを実行できるようにする例を確認しましょう。

例：クレジットカードのランキング

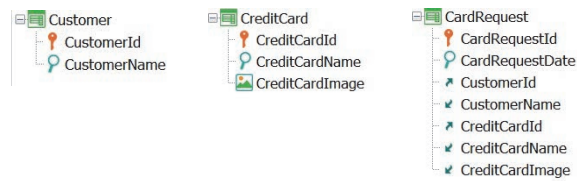


The screenshot shows a web application interface with a title bar 'Travel Agency - Credit Cards'. Below the title bar, there are two tabs: 'Recents' and 'WPCard Requests'. The 'WPCard Requests' tab is active, displaying a table titled 'Ranking'. The table has two columns: 'Card Name' and 'Requests'. The data is as follows:

Card Name	Requests
VISA	4
American Express	3
MasterCard	2
Diners Club	2

旅行代理店が、取り扱っているさまざまなクレジットカードの発行について、顧客からの申請を登録する機能を提供するとします。ここでは、2020年に登録された申請数に基づき、クレジットカードのランキングを実装します。これはランキングであるため、数が多い順に情報を表示する必要があります。

例: クレジットカードのランキング



Name	Type	Description	Is Collection
SDTCardRequests		SDTCard Requests	<input checked="" type="checkbox"/>
SDTCardRequestsItem			
CreditCardId	Attribute:CreditCardId	Credit Card Id	<input type="checkbox"/>
CreditCardImage	Attribute:CreditCardImage		<input type="checkbox"/>
CreditCardName	Attribute:CreditCardName	Card Name	<input type="checkbox"/>
CreditCardRequests	Numeric(4,0)	Requests	<input type="checkbox"/>

ここに示すトランザクションの設計と、すべてのカードのコレクションをロードできる構造化データタイプを考えます。各カードについて、2020年に登録された申請数を取得します。

例: クレジットカードのランキング

```
SDTCARDRequests from CreditCard
{
  SDTCARDRequestsItem
  {
    CreditCardId
    CreditCardImage
    CreditCardName
    CreditCardRequests = count(CardRequestDate, CardRequestDate.Year()=2020)
  }
}
```


Name	Type	Description	Is Collection
SDTCARDRequests		SDTCARD Requests	<input checked="" type="checkbox"/>
SDTCARDRequestsItem			<input type="checkbox"/>
CreditCardId	Attribute:CreditCardId	Credit Card Id	<input type="checkbox"/>
CreditCardImage	Attribute:CreditCardImage		<input type="checkbox"/>
CreditCardName	Attribute:CreditCardName	Card Name	<input type="checkbox"/>
CreditCardRequests	Numeric(4,0)	Requests	<input type="checkbox"/>

GeneXus を開きます。

SDT をロードするにはどうすればよいでしょうか。データプロバイダーを使用します。

このデータプロバイダーには、各クレジットカードと、2020 年に登録された申請数がロードされます。この式に記述されている計算条件はそれを表しています。

例: クレジットカードのランキング

Ranking			
CARD	Credit Card Id	Card Name	Requests
	&CardRequests.item(0).CreditCardId	&CardRequests.item(0).CreditCardName	&CardRequests.item(0).CreditCardRequests

```

Event Start
  &CardRequests = DPRequests()
  &CardRequests.Sort("[CreditCardRequests]")
Endevent

```

Name	Type	Is Collection
Variables		
Standard Variables		
• CardRequests	SDTCARDRequests	<input type="checkbox"/>

次に、Web パネルの Start イベントでこのデータプロバイダーの出力を受け取ります。その際、SDTCARDRequests データタイプに基づいて定義された &CardRequests 変数で受け取ります。

実際のランキングにするには、登録された申請数に基づき多い順に並べ替える必要があります。そのために Sort メソッドを使用します。

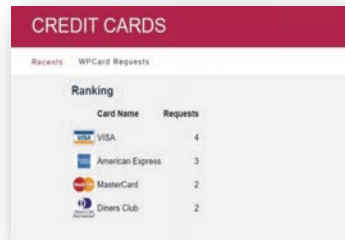
また、このコンテンツをグリッドに効果的に表示する必要があります。

そのために、Web パネルのフォームに &CardRequests 変数を挿入します。これは SDT コレクション変数であるため、GeneXus がその変数とグリッドを自動的に関連付けます。

ただし、これは変数であり、Web パネルでは変数は入力データになります。そのため、コンテンツを変更できないように、読み取り専用指定します。

コンテンツをフリー スタイル グリッドに表示するかどうかを選択することもできます。このオプションをチェックしない場合は通常のグリッドに表示されます。

実行時



The screenshot shows a window titled 'CREDIT CARDS' with a sub-header 'Recently: WPCard Requests'. Below this is a 'Ranking' table with two columns: 'Card Name' and 'Requests'. The table lists four credit cards: VISA (4 requests), American Express (3 requests), MasterCard (2 requests), and Diners Club (2 requests). Each card name is preceded by its respective logo.

Card Name	Requests
VISA	4
American Express	3
MasterCard	2
Diners Club	2

ここでは次の点について考えましょう。

SDT コレクション変数からロードされたこのグリッドと、通常の方法で変数を使用してロードされたグリッドに違いはあるでしょうか。いいえ。どちらもベーステーブルのないグリッドです。

実行環境で確認してみましょう。F5 キーを押します。

これでランキングの準備ができました。ここでは、次の点を解決するとします：これらのクレジットカードのうち 1 つを選択し、登録済みの申請を取得して PDF のリストを表示する必要があります。

GeneXus に戻りましょう。

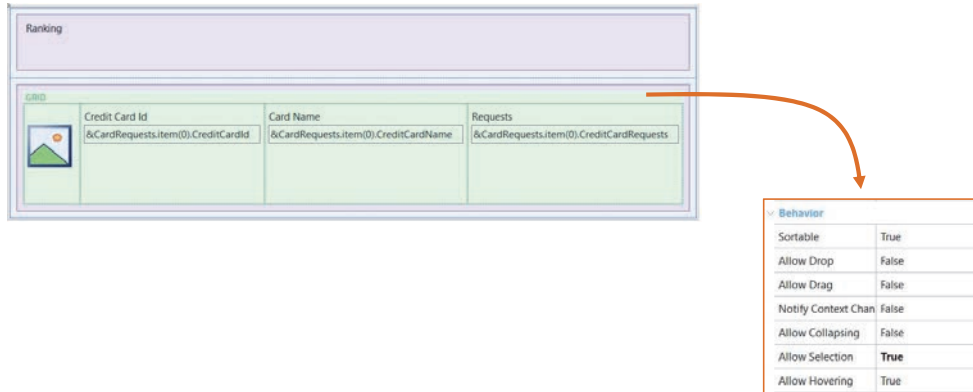
CardRequestList のリストは定義済み

```
▢ Parm(in:CreditCardId, in:&CreditCardName);  
▢ output_file("List", "pdf");
```

```
▢ in printblock1  
.....  
.....&CreditCardName.....  
.....Card requests.....  
.....  
▢ in printblock2  
.....CardReq CustomerName.....  
.....  
1 print printblock1  
2  
3 For each CardRequest  
4   print printblock2  
5 endfor
```

説明を簡単にするために、ここでは CardRequestList プロシーチャーは定義済みであるものとします。このプロシーチャーでは、クレジットカードの ID および名前を Parm ルールで受け取り、登録済みの申請を表示します。

[Allow Selection] プロパティおよび [Allow Hovering] プロパティ



Web パネルに戻しましょう。

グリッドの行を選択できる必要があります。グリッドのプロパティを確認しましょう。[Allow Selection] プロパティの値を True にして有効にします。

これで [Allow Hovering] プロパティも有効になります。これにより、行を選択した場合に色付きで強調表示されます。

既定の色は、適用されているテーマを通じて変更できます。その場合、Grid クラスレベルで、[Selected Row Class] プロパティおよび [Hover Row Class] プロパティに割り当てられているクラスを編集します。

選択した行を取得し、アクションを実行できるようにするイベントも必要です。このイベントは、グリッドに関連付けられた OnlineActivate イベントです。

[Events] エLEMENTで、「Event」の後にグリッド名 (この場合は Grid1) とピリオドを入力し、OnlineActivate イベントを選択します。

ここでは、先ほど確認したリストを呼び出し、行で選択したカードの ID および名前をパラメーターで渡します。

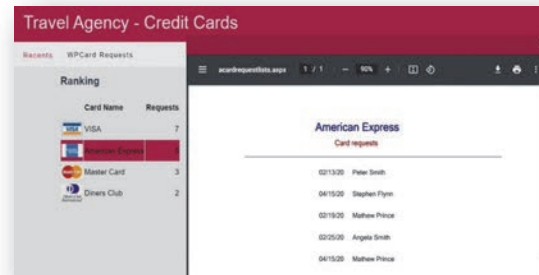
[CurrentItem] プロパティ

```

Event Start
    &CardRequests = DPREquests()
    &CardRequests.Sort("[CreditCardRequests]")
EndEvent

Event Grid1.OnLineActivate
    &CreditCardId = &CardRequests.CurrentItem.CreditCardId
    &CreditCardName = &CardRequests.CurrentItem.CreditCardName
    CardRequestLists.Popup(&CreditCardId, &CreditCardName)
EndEvent

```



それを行うにはどうすればよいでしょうか。

グリッドがロードしたコレクションを表示する際に、ID および名前を取得するために、コレクションの選択されたアイテムを参照する必要があります。そのために、[CurrentItem] プロパティを使用します。

次に、&CreditCardId 変数を定義します。同じ名前を持つ項目属性に基づき、関連付けが自動的に行われます。コレクション変数 &CardRequests の後に「.CurrentItem.CreditCardId」と入力し、行で選択されたカードの ID を取得して割り当てます。

名前を取得するために同じ処理を行います。

&CreditCardName 変数を定義し、同じ名前の項目属性に基づき関連付けが行われていることを確認し、対応する割り当てを定義します。

最後に、定義済みのリストを呼び出してポップアップウィンドウとして開き、必要なパラメーターである &CreditCardId 変数および &CreditCardName 変数を渡します。

実行環境で確認してみましょう。F5 キーを押します。

Panel オブジェクトのグリッドの SDT コレクション

- Refresh イベントでコレクションを受け取ります。
- フォームに変数を挿入すると、選択済みのフィールドを含むグリッドを GeneXus が自動で作成します。
- 選択した行に対して任意のアクションを実行するには、イベントを定義し、[CurrentItem] プロパティを使用する必要があります。
- このイベントは、グリッドの既定のアクションとして定義する必要があります ([Default Action] プロパティ)。

最後に、Panel タイプのオブジェクトを操作する場合は、次のようなプロセスになります。

- Refresh イベントで、コレクションを変数で受け取ります。
- この変数がフォームに挿入され、選択したフィールドに関連付けられたグリッドを GeneXus が自動的に作成します。
- 選択した行に対して特定のアクションを実行する場合は、イベントを定義して [CurrentItem] プロパティを使用する必要があります。
- 最後に、このイベントは、[Default Action] プロパティを使用してグリッドの既定のアクションとして定義する必要があります。