

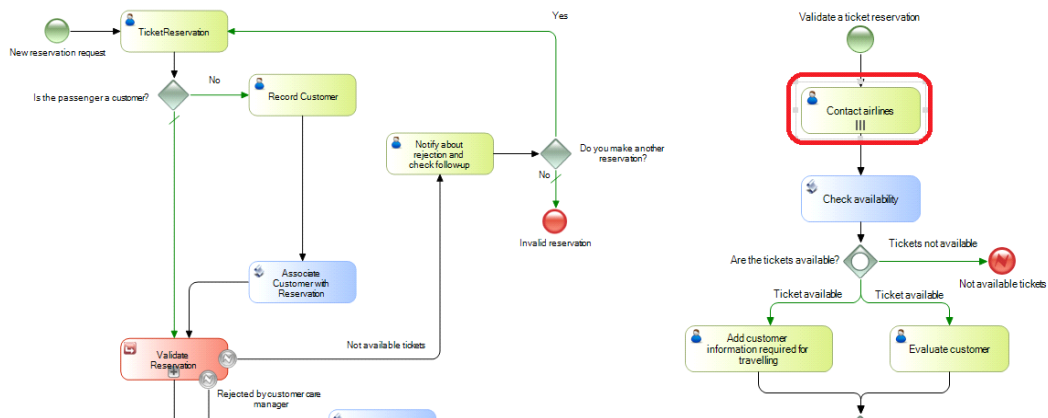
## Início de um processo em um objeto GeneXus usando a API de Workflow

A agência de viagens decidiu modificar seu processo de reserva de passagens e incorporar algumas funcionalidades novas.

Solicitaram que um cliente da agência possa fazer sua própria reserva através do site.

O cliente deverá estar logado no site e fornecer os dados da reserva (data, aeroporto de origem, aeroporto de destino, quantidade de pessoas etc), com os quais o sistema criará uma reserva, iniciará o processo FlightTicketReservation e marcará a reserva TicketReservation como completa.

Como o passageiro já é cliente da empresa, cumprida a tarefa TicketReservation, a tarefa seguinte é a primeira tarefa do subprocesso ValidateReservation: ContactAirlines, uma tarefa multi-instanciada.



O sistema deverá avisar por e-mail a cada uma das pessoas que podem entrar em contato com as companhias aéreas que as instâncias da tarefa ContactAirlines estão disponíveis para serem executadas. Por fim, substituir o tipo de tarefa que notificava o cliente de que a reserva foi autorizada para que o aviso seja pessoal.

Começamos pela reserva por parte do cliente no site.

Para isso, vamos utilizar uma webpanel chamada TravelAgency. Essa web panel têm variáveis na tela por meio das quais o usuário vai inserir os dados da reserva, e um botão de confirmar.

You are logged as: <input type="text" value="&amp;CustomerEmail"/>	
Customer name: <input type="text" value="&amp;CustomerName"/>	
Flight ticket reservation	
Please enter you reservation information:	
Date	<input type="text" value="&amp;ReservationDate"/>
Quantity	<input type="text" value="&amp;ReservationQty"/>
Departure Airport	<input type="text" value="&amp;ReservationDepartureAirportId"/> , <input type="text" value="&amp;ReservationDepartureCityName"/> , <input type="text" value="&amp;ReservationDepartureCountryName"/>
Arrival Airport	<input type="text" value="&amp;ReservationArrivalAirportId"/> , <input type="text" value="&amp;ReservationArrivalCityName"/> , <input type="text" value="&amp;ReservationArrivalCountryName"/>
<input type="button" value="Confirm"/>	

Por razões de simplicidade para a demo, assumiremos que o cliente que insere é o cliente 1 e que ele já está logado, motivo pelo qual não incluiremos os controles de login. Se formos ao evento Start, veremos o código para simular isso.

```

1  Event Start
2      &CustomerId = 1 // Hardcoded only for demo uses
3      &CustomerEmail = GetLoggedUser (&CustomerId)
4      &CustomerName = GetCustomerName (&CustomerId)
5  Endevent

```

Ao pressionar o botão confirmar, o evento Enter - que realiza várias tarefas - é executado.

```

7  Event Enter
8      //Create a new reservation with the entered data
9      &ReservationId = NewReservation (&ReservationDate, &ReservationQty, &CustomerId,
10         &ReservationDepartureAirportId, &ReservationArrivalAirportId )
11
12     //Create a new FlightTicketReservation process instance
13     &WorkflowServer.Connect ("WFADMINISTRATOR", "WFADMINISTRATOR")
14     &WorkflowProcessDefinition = &WorkflowServer.GetProcessDefinitionByName ("FlightTicketReservation")
15     &WorkflowProcessInstance = &WorkflowProcessDefinition.CreateInstance ()
16     &WorkflowProcessInstance.Subject = 'FlightTicketReservation process started from GeneXus Menu'
17     &ReservationIdWorkflowApplicationData = &WorkflowProcessInstance.GetApplicationDataByName ("ReservationId")
18     &ReservationIdWorkflowApplicationData.NumericValue = &ReservationId
19     &WorkflowProcessInstance.Start () // Initiate the FlightTicketReservation process instance
20
21     //Mark the TicketReservation task as completed
22     &WorkflowActivity = &WorkflowProcessDefinition.GetActivityByName ('TicketReservation')
23     &WorkflowWorkitem = &WorkflowProcessInstance.GetWorkitemByActivity (&WorkflowActivity)
24     &WorkflowWorkitem.Complete ()
25     Commit
26 Endevent

```

Primeiramente, o método `NewReservation` é invocado para criar uma reserva no banco de dados. Depois, utilizando **Tipos de dados de workflow**, começa o processo `FlightTicketReservation` e a tarefa `TicketReservation` é marcada como completa.

Esses Tipos de Dados que começam com o prefixo `Workflow` são tipos de dados GeneXus e permitem que a aplicação possa interagir com o motor de Workflow.

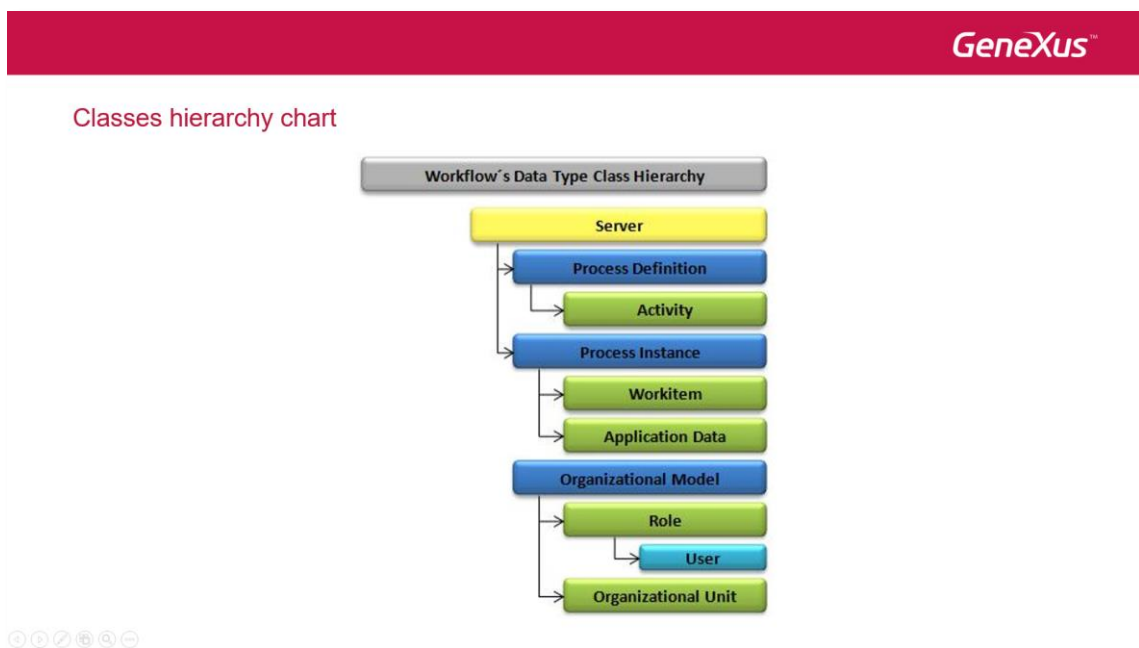
Para poder usar um tipo de dado `Workflow` e interagir com a API do motor, é preciso definir primeiro uma variável com esse tipo de dados. Depois, com a ajuda que a informação contextual nos dá, podemos escolher o método ou propriedade que queremos usar.

```

7 Event Enter
8 //Create a new reservation with the entered data
9 &ReservationId = NewReservation(&ReservationDate,&ReservationQty,&CustomerId,
0     &ReservationDepartureAirportId,&ReservationArrivalAirportId )
1
2 //Create a new FlightTicketReservation process instance
3 &WorkflowServer.Connect("WFADMINISTRATOR","WFADMINISTRATOR")
4 &WorkflowProcessDefinition = &WorkflowServer.GetProcessDefinitionByName("FlightTicketReservation")
5 &WorkflowProcessInstance = &WorkflowProcessDefinition.CreateInstance()
6 &WorkflowProcessInstance.Subject = 'FlightTicketReservation'
7 &ReservationIdWorkflowApplicationData = &WorkflowProcessInstance.ApplicationDataByName("ReservationId")
8 &ReservationIdWorkflowApplicationData.Name = &ReservationId
9 &WorkflowProcessInstance.Start() // Initia
0
1 //Mark the TicketReservation task as complete
2 &WorkflowActivity = &WorkflowProcessDefinition.GetActivity("TicketReservation")
3 &WorkflowWorkitem = &WorkflowProcessInstance.CreateWorkitem(&WorkflowActivity)
4 &WorkflowWorkitem.Complete()
5 Commit
6 Endevent

```

Os tipos de dados `Workflow` estão organizados em uma hierarquia de classes.



A classe de nível hierárquico superior é a **Server**, da qual dependem diretamente 3 classes: **Process Denifition**, que nos permite acessar os componentes de um diagrama de processos, **Process Instance**, que nos permite acessar uma instância de um processo em execução e **Organizational Model**, que nos permite acessar a informação da estrutura organizacional da empresa, como papéis e usuários.

A classe Server é o ponto de entrada na hierarquia de tipos e seus métodos nos permitem acessar qualquer tipo de dados workflow.

Os tipos de dados mais usados são os seguintes:

**GeneXus™**

**Most used Workflow Data Types**

- WorkflowProcessDefinition
  
- WorkflowProcessInstance
  
- WorkflowWorkItem
  
- WorkflowContext
  
- WorkflowApplicationData

Por meio do **WorkflowProcessDefinition**, podemos acessar várias propriedades do diagrama, como nome, versão, tarefas que o incluem (às quais chamamos de atividades) e também é possível por meio do método CreateInstance, criar uma instância em execução do processo, baseada nessa definição.

**GeneXus™**

**Most used Workflow Data Types**

- WorkflowProcessDefinition {
  - ✓ Diagram name
  - ✓ Diagram version
  - ✓ Activities (tasks)
  - ✓ Create Instance
  
- WorkflowProcessInstance
  
- WorkflowWorkItem
  
- WorkflowContext
  
- WorkflowApplicationData

Utilizando WorkflowProcessInstance, podemos conhecer a definição do processo na qual a instância de baseia, o assunto tratado pela instância, a coleção dos workitems que fazem parte da instância e, através do método GetApplicationByName, podemos recuperar um dado relevante a partir do seu nome.

### Most used Workflow Data Types

- WorkflowProcessDefinition
- WorkflowProcessInstance
  - ✓ Process Definition
  - ✓ Subject
  - ✓ Workitems
  - ✓ GetApplicationByName
- WorkflowWorkItem
- WorkflowContext
- WorkflowApplicationData

A classe WorkflowWorkItem nos permite conhecer o trabalho que será realizado por um participante no contexto de uma atividade dentro da instância do processo.

Sua propriedade ProcessInstance nos dá informação da instância do processo ao qual o workitem pertence. A propriedade Activity devolve a atividade que gerou o workitem.

O método Assign nos permite atribuir um workitem a um usuário determinado e o método Complete permite dar por finalizada a execução do workitem.

### Most used Workflow Data Types

- WorkflowProcessDefinition
- WorkflowProcessInstance
- WorkflowWorkItem
  - ✓ Process instance
  - ✓ Activity
  - ✓ Assign
  - ✓ Complete
- WorkflowContext
- WorkflowApplicationData

O tipo de dados WorkflowContext nos dá informação de contexto na execução de um aplicativo associado a uma atividade. Este contexto se instancia automaticamente se a aplicação associada à atividade (ou seja, à tarefa) é um objeto GeneXus que faz parte da mesma KB onde o diagrama do processo está.

Essa instanciação automática do contexto nos permite conhecer os valores da definição do processo, a instância do processo e do workitem associado à atividade.

**GeneXus™**

**Most used Workflow Data Types**

- WorkflowProcessDefinition
  
- WorkflowProcessInstance
  
- WorkflowWorkItem
  
- WorkflowContext {
  - ✓ Process Definition
  - ✓ Process Instance
  - ✓ Workitem
  
- WorkflowApplicationData

Por último, o tipo de dados WorkflowApplicationData é o tipo de dado que usamos quando queremos trabalhar com dados relevantes, por exemplo, quando armazenamos um dado relevante que obtemos através do método GetApplicationDataByName.

**GeneXus™**

**Most used Workflow Data Types**

- WorkflowProcessDefinition
  
- WorkflowProcessInstance
  
- WorkflowWorkItem
  
- WorkflowContext
  
- WorkflowApplicationData {
  - ✓ Relevant datas

Voltando ao evento da webpanel que invoca o processo FlightTicketReservation, vemos que temos definida uma variável &WorkflowServer do tipo WorkflowServer. Como prática, utilizamos sempre nomes de variáveis que coincidem com os tipos de dados Workflow, de forma que nos resulte mais simples sua identificação.

A primeira operação que realizamos com o tipo de dados WorkflowServer é conectarmos ao motor de workflow, utilizando o usuário e a senha do administrador.

```

//Create a new FlightTicketReservation process instance
&WorkflowServer.Connect("WFADMINISTRATOR","WFADMINISTRATOR")
&WorkflowProcessDefinition = &WorkflowServer.GetProcessDefinitionByName("FlightTicketReservation")
&WorkflowProcessInstance = &WorkflowProcessDefinition.CreateInstance()
&WorkflowProcessInstance.Subject = 'FlightTicketReservation process started from GeneXus Menu'
&ReservationIdWorkflowApplicationData = &WorkflowProcessInstance.GetApplicationDataByName("ReservationId")
&ReservationIdWorkflowApplicationData.NumericValue = &ReservationId
&WorkflowProcessInstance.Start() // Initiate the FlightTicketReservation process instance

```

Depois, obtemos a definição do processo FlightTicketReservation a partir de seu nome e o guardamos em uma variável do tipo WorkflowProcessDefinition.

Assim que tivermos a definição, criamos uma instância desse processo utilizando o método CreateInstance. Depois, mudamos seu assunto para poder reconhecer o processo rapidamente na caixa de entrada.

Em seguida, carregamos o dado relevante ReservationId com o identificador da reserva que criamos anteriormente e finalmente iniciamos a instância com o método Start().

As linhas de código abaixo servem para marcar a tarefa TicketReservation como completa.

```

//Mark the TicketReservation task as completed
&WorkflowActivity = &WorkflowProcessDefinition.GetActivityByName('TicketReservation')
&WorkflowWorkitem = &WorkflowProcessInstance.GetWorkitemByActivity(&WorkflowActivity)
&WorkflowWorkitem.Complete()
Commit

```

A partir da definição do processo, obtemos a atividade TicketReservation e, com essa atividade, obtemos o workitem correspondente à tarefa na execução no processo. Depois, marcamos o workitem (ou seja, a tarefa) como completo.

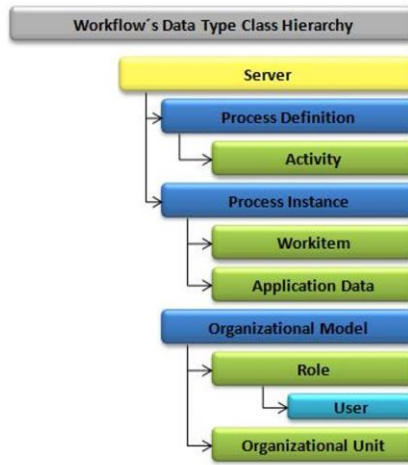
Note que depois do método Complete(), há um Commit. As alterações realizadas utilizando os tipos de dados de workflow estão compreendidas dentro da **Unidade de Trabalho Lógica** da aplicação.

No entanto, **as operações de workflow não fazem Commit**, por isso devemos nos assegurar de definir bem a UTL na aplicação e fazer o Commit ao final. Neste caso, fizemos as operações de workflow em uma webpanel, por isso é preciso adicionar o Commit ao finalizá-las.

Esses tipos de dados workflow que vimos é um subconjunto de todos os disponíveis, por isso podemos realizar muitas tarefas por código, através da API do motor de workflow.

Você poderá encontrar mais informações sobre este tema no seguinte link.

Classes hierarchy chart



[http://wiki.gxtechnical.com/commwiki/servlet/hwiki?Category%3AWorkflow+Data+Types.](http://wiki.gxtechnical.com/commwiki/servlet/hwiki?Category%3AWorkflow+Data+Types)