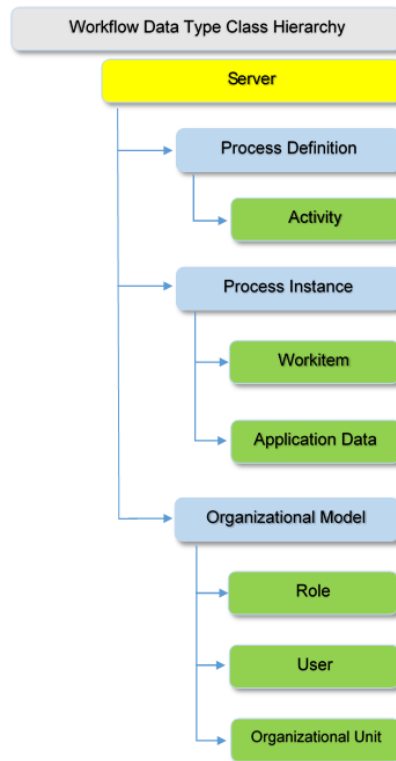


API Workflow avançado

A maior parte da API está organizada em uma hierarquia, com o Server na parte superior. Isso significa que o Server representa o ponto de entrada para todas as funções que vimos até agora, incluindo os modelos, a execução desses modelos e as informações de usuário das pessoas que os executam. Neste vídeo em particular, nos concentraremos neste objeto e nas características que nos permite incluir em nosso código GeneXus



WorkflowServer possui uma série de propriedades que nos permitem obter informações sobre o ambiente de execução. Estas incluem:

- A sessão de trabalho atual do usuário registrado
- A informação do usuário e
- A configuração global para todo o servidor.

Além dessas propriedades, o objeto Workflow Server nos dá acesso a uma série de métodos e funcionalidades, que são:

- **Connect and disconnect:** Conectar e desconectar em uma sessão de GXflow, fornecendo o nome de usuário e senha. Esta sessão é aquela que usará o resto do código em GeneXus para executar e o usuário será aquele que retornará em `Server.CurrentUser`, portanto algumas funcionalidades descritas abaixo podem falhar se o usuário atual não tiver os privilégios adequados.
- **ListProcessDefinitions:** Lista todas as definições de processo com uma série de filtros, em uma ordem específica ou acessando diretamente uma única definição de processo usando seu ID ou nome.
- **ListActivities:** Lista todas as atividades em um processo, usando alguns filtros adicionais ou acessando diretamente um usando sua identificação ou nome.
- **ListProcessInstances:** Lista todas as instâncias (ou execuções) de processos de fluxo de trabalho usando alguns filtros ou acessando apenas um usando seu ID.
- **ListWorkitem:** Lista todos os workitems utilizando alguns filtros ou acessando apenas um.
- **GetCalendar:** Permite acessar todos os calendários definidos num Servidor, com suas definições de feriados e dias úteis, bem como acessar apenas um se soubermos seu id ou nome
- **GetNode:** Permite o acesso a todos os nós de processo, ou seja, obter informações sobre cada atividade, cada condição, nó de evento e os vínculos entre eles.
- **GetOrganizationalModel:** Permite acessar o modelo organizacional definido no Servidor
- **GetEventsRepository:** Permite acessar o repositório de eventos, para acessar as informações de registro do Servidor.
- **GetDocumentsrepository:** Permite acessar o repositório de conteúdo dos documentos no servidor.



Para vários recursos, o tipo de dados WorkflowFilter é uma parte importante do uso adequado da funcionalidade. O objeto é um superconjunto de todos os filtros possíveis que podem ser interessantes para nós incluímos ao acessar uma série de instâncias, workitems ou alguns objetos. Sem ele, seríamos forçados a filtrar a aplicação na coleção completa, o que pode ser extremamente ineficaz em ambientes de produção que geralmente funcionam com centenas de processos e milhões de workitems. Nem todos os filtros se aplicam a todas as funções, mas é útil saber quais se aplicam em cada lugar para maximizar nosso desempenho no uso da API. No entanto, também é importante saber que o uso de um filtro que não se aplica a uma função não dispara um erro.

- O filtro **User** : Aplica-se a todas as entidades reguladas pela presença de um usuário, como workitems atribuídos, eventos ativados ou documentos atribuídos.
- O filtro **Role**: Aplica-se às entidades reguladas por roles, como candidatos para uma atividade ou usuários com alguma role atribuída.
- O filtro **OrganizationalUnit**: Aplica-se às entidades reguladas por elas, como usuários ou roles.
- O filtro **OrganizationalUnitDefinition** : Aplica-se à lista de organizações.
- O filtro **ProcessDefinition**: Aplica-se a todas as entidades que incluem ProcessDefinition. Estes incluem workitems, instâncias de processo, atividades e roles de candidatos.
- O filtro **ProcessInstance**: Aplica-se às entidades definidas por uma instância, que incluem documentos, workitems e usuários atribuídos.
- O filtro **Subject**: Aplica-se ao tema das instâncias e dos workitems. Este filtro diferencia entre maiúsculas e minúsculas.
- O filtro **Activity**: Aplica-se a workitems e roles candidatos dessa atividade.
- O filtro **From/To**: Definem um limite inferior e superior para a data de criação de documentos, eventos, instâncias de processo e workitems.

- O filtro **Name**: Aplica-se ao nome descritivo dos tipos de documentos, o título dos documentos e os nomes das roles, restrições, usuários, tarefas e definições de processos. Todos estes filtros diferenciam entre maiúsculas e minúsculas.
- **Start/limit**: São utilizados para melhorar o desempenho de obtenção de vários elementos, para utilizá-los em funcionalidades como relatórios ou paginação. Start especifica o deslocamento do conjunto de dados recuperado, enquanto Limit especifica a quantidade de objetos retornados. A maioria dos DBMS modernos oferece suporte a esta funcionalidade no nível do servidor, em vez do controlador, melhorando a largura de banda e o tempo de transferência com a base de dados, embora possa haver algumas restrições para seu uso. Por exemplo, SQL Server oferece suporte a esta funcionalidade desde 2012, mas requer que seja especificado um pedido.
- **State**: Inclui objetos que são afetados pelo motor de workflow, portanto, pode ser utilizado para filtrar definições de processos, instâncias, workitems e documentos. Este filtro é bastante complexo, pois tem uma série de valores possíveis e nem todos se aplicam aos diferentes tipos de objetos, então vamos nos aprofundar um pouco mais:

Quando são utilizados com definições de processo, podem ser habilitados (ou seja, podem ser criadas novas instâncias) ou desabilitados (para que não possam ser criadas novas instâncias)

WorkflowProcessDefinitionState
ENABLED
DISABLED

Quando são utilizados com instâncias de processos, podem ser:

OPEN_RUNNING: Quando uma instância é criada e executa normalmente.

OPEN_NOTRUNING_SUSPENDED: Quando é executada a funcionalidade de Suspende na instância, o estado muda para NOTRUNING_SUSPENDED. Além disso, todos os subprocessos e workitems ativos são alterados para suspensos para garantir que nenhuma parte da instância permaneça ativa.

CLOSED_COMPLETED: este é o estado utilizado quando uma instância é concluída normalmente.

CLOSED_ABORTED: Similar a suspenso, quando é executada a funcionalidade de Abortar a instância, esta instância e todos os seus componentes são definidos para este estado.

CLOSED_TERMINATED: este estado é utilizado quando a instância termina de forma inesperada, especialmente em uma data limite

Quando usado em um workitem, o filtro de estado pode ser:

OPEN_ACTIVE_READY: Quando é criado o workitem

OPEN_ACTIVE_ASSIGNED: Quando foi atribuído ou tomado por um usuário

OPEN_ACTIVE_INPROCESS: Enquanto o workitem está em execução, ou seja, depois de ter sido tomado, mas antes de ser enviado.

CLOSED_COMPLETED: Quando o elemento de trabalho foi devolvido e foi considerado concluído.

Voltando aos filtros:

Os filtros **CreatedFrom** e **CreatedBy** funcionam de maneira similar a From/To. Na verdade, From/To são mantidos por razões de compatibilidade.

Os filtros **EndedFrom** e **EndedBy** funcionam de maneira similar aos anteriores, mas para objetos que têm uma data de término, como workitems e instâncias. É importante detalhar que as instâncias que estão abertas atualmente serão reduzidas por estes filtros, se for especificado. No caso de estar interessado nas instâncias abertas atualmente, deve usar o filtro State em seu lugar.

Search, pode ser utilizado para filtrar uma combinação de diferentes atributos em diferentes entidades: para workitems, filtra por subject, seu nome, o nome de seu processo, o nome da tarefa ou o nome do usuário; para os usuários, filtra por código, nome ou e-mail; para restrições, filtra por código ou valor; para processos, filtra por nome e descrição; para instâncias, filtra por subject, o nome de seu processo ou o nome de seu usuário e para os documentos, seu título ou seu responsável.

WorkflowPriority, pode ser utilizado para filtrar workitems e instâncias por sua prioridade estabelecida, as opções são:

- **High priority**, prioridade alta.
- **Normal priority**, normal e
- **Low priority** baixa

Outro objeto que devemos considerar ao programar com GeneXus na API é **WorkflowContext**. Geralmente, os objetos que devem ser usados como Gateways requerem os seguintes parâmetros: Process Definition, Process Instance e Workitem, para identificar em que parte do fluxo de trabalho nos encontramos.

parm(in:&WorkflowProcessDefinition, in:&WorkflowProcess, in:&WorkflowWorkitem, out:&ConditionalCode);

Desde as últimas versões de GeneXus, introduzimos WorkflowContext, que se define automaticamente em cada transação e webpanel que se chama a partir do workflow e inclui:

ProcessDefinition: A definição de processo que está em execução atualmente

ProcessInstance: A instância de processo que está em execução atualmente

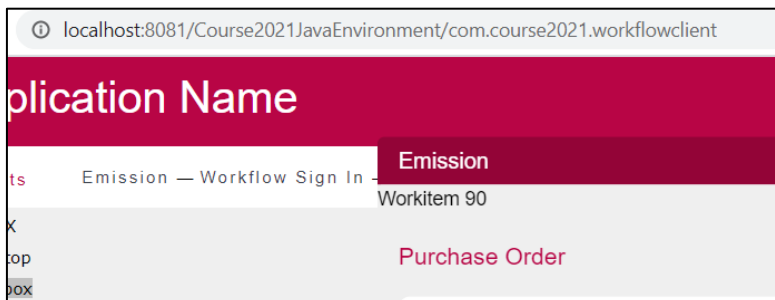
Workitem: O elemento de trabalho atual

E também a funcionalidade para verificar os direitos de um usuário com uma atividade em particular.

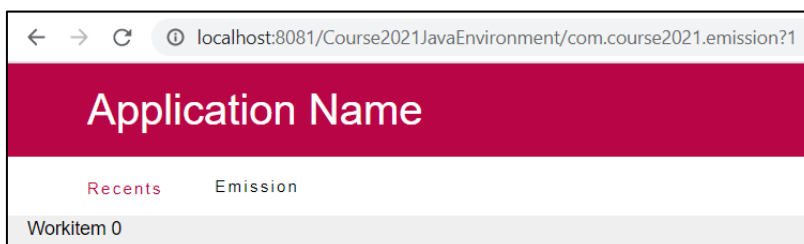
Isto significa que o mesmo código no mesmo objeto se comportará de maneira diferente se for executado dentro da caixa de entrada ou como uma URL externa

```
Event Start
    msg(!"Workitem "+&WorkflowContext.Workitem.Id)
EndEvent
```

Caixa de entrada



URL externa



O tipo de dados WorkflowUser amplia as propriedades que definem um usuário no Workflow, que incluem:

- **UserName:** Deve ser único.
- **FullName**
- **Email** (que é opcional e não único).
- **AccessLevel**, que define se o usuário pode executar esta API sobre outros usuários ou apenas sobre si mesmo.
- Um conjunto de roles, OrganizationalUnits, Restrictions, ExternalCode, que se utiliza para interagir com outros módulos, como GeneXus Access Manager e se o usuário se encontra atualmente conectado, bloqueado ou "Fora do escritório". Caso se encontre fora do escritório, as propriedades também incluem seu substituto.

Além dessas propriedades, o tipo de dados inclui muitas funções.

Pode obter a worklist atual do usuário, **GetWorklist** ou a lista de processos que o usuário tem direito de iniciar, utilizando os filtros descritos anteriormente. Sempre que seu usuário tenha suficiente nível de acesso, também pode obter as roles definidas dentro da unidade organizacional que tem o usuário. Pode atribuir ou remover roles, unidades organizacionais ou restrições do usuário.

Como administrador, também pode bloquear ou habilitar um usuário, bem como configurá-lo como "Fora do escritório", caso em que também pode selecionar outro usuário como substituto.

Finalmente, pode forçar a reconstrução da worklist do usuário (**RebuildWorklist**). Ao fazer isso, voltarão a ser aplicadas todas as regras relacionadas com as roles candidatas para os workitems que possa ter o usuário, considerando as restrições e unidades organizacionais adequadas para, em essência, atualizar a lista de workitems que um usuário pode assumir e executar.

A instalação do servidor está configurada de forma predeterminada para realizar estes cálculos para qualquer alteração nos direitos dos usuários, como podem consumir muitos recursos e tempo em ambientes com dezenas de roles, centenas de usuários. e milhões de workitems, tem a opção de desabilitar a reconstrução automática e ativá-la manualmente sob certas condições (por exemplo, quando um usuário inicia sessão)

-WorkflowRole	}	✓ Name/Value
-WorkflowRestriction		✓ AddUser
-WorkflowOrganizationalUnit		✓ RemoveUser

O último conjunto de tipos de dados que gostaríamos de discutir são Workflowrole, WorkflowRestriction e WorkflowOrganizationalUnit. Mesmo quando são conceitualmente diferentes e têm diferentes casos de uso, compartilham a característica de que todos são modificadores para definir o que um usuário pode ou não fazer.

Como tal, todos eles têm um nome ou valor e também as funcionalidades para adicionar e remover usuários de seus metadados