

Added by Jackie Ju, last edited by Jackie Ju on Dec 20, 2008

TMS

TMS implements the HT operations and expose web service. B4P compliant engine can call TMS as a human task engine. It mainly consist of following parts.

WSDL

For standard B4P HT operatoin, please see [WS-HumanTask specification](#)

Implementations Note: Create a web service extending B4P HT operation WSDL on axis, in project tempo-tms-service.

TMS Service

Tempo projects: Tempo-tms-Service

TMSRequestProcessor: This class implements web service interface, will call underly marshaller/umarseller to unmarshall/umarsall request/response and call underlying layer - TMSSerer, which implements operation logic.

Full Name: org.intalio.tempo.workflow.tms.server.TMSRequestProcessor

Implementations Note: The most big class for this part, will add pretty much interfaces, but not the hard part, since which just call interfaces of TMSServer in most case.

Main Interfaces:

Interface	Function	states change	Permission	Error Handling	Implementation Notes
claim	Claim responsibility for a task,	set the task to status Reserved	Potential Owners Business Administrator		just write
start	Start the execution of the task.	set the task to status InProgress.	Actual Owner Potential Owners (state Ready)		just write
stop	Cancel/stop the processing of the task.	=>reserved	Actual Owner Business Administrator		just write
release	release task	ready	Actual Owner Business Administrator		just write
suspend	suspend the task	no change	Potential Owners (state Ready) Actual Owner Business Administrator		just write
suspendUntil	Suspend the task for a given period of time or until a fixed point in time. The caller has to specify either period of time or a fixed point in time.	no change	Potential Owners (state Ready) Actual Owner Business Administrator		just write
resume	Resume a suspended task	no change	Potential Owners (state Ready) Actual Owner Business Administrator		just write
complete	Execution of the task finished successfully.If no output data is set the operation returns illegalArgumentFault.	complete	Actual Owner		can modify or reuse .
remove	Only applies to notifications	N/A	Notification Recipient		call TMSServer
fail	Actual owner completes the execution of the task raising a fault	N/A	Actual Owner		call TMSServer

	execution of the task raising a fault.				
setPriority	Change the priority of the task.	N/A	Actual Owner Business Administrator		call TMSServer
addAttachment	Add attachment to a task		Actual Owner Business Administrator		
getAttachmentInfos	Get attachment information for all attachments associated with the task		Potential Owners Actual Owner Business Administrator		
getAttachments	Get all attachments of a task with a given name.		Potential Owners Actual Owner Business Administrator		
deleteAttachments	Delete the attachments with the specified name from the task (if multiple attachments with that name exist, all are deleted).		Actual Owner Business Administrator		cannot reuse removeAttachment because of difference semantic
addComment	Add a comment to a task.		Potential Owners Actual Owner Business Administrator		
getComments	Get all comments of a task		Potential Owners Actual Owner Business Administrator		
skip	Skip the task. If the task is not skipable then the fault illegalOperationFault is returned.		Task Initiator Actual Owner Business Administrator		
forward	Forward the task to another organization entity.		Potential Owners Actual Owner Business Administrator		Can not reuse reassign because of difference semantic
delegate	Assign the task to one user	set the task to state Reserved.	Potential Owners (only in Ready state) Actual Owner Business Administrator		
getRendering	Returns the rendering specified by the type parameter		Any		
getRenderingTypes	Returns the rendering types available for the task or notification		Any		
getTaskInfo	Returns a data object of type tTask		Any		
getTaskDescription	Returns the presentation description in the specified mime type.		Any		can not reuse getTask() because task object are different
setOutput	Set the data for the part of the task's output message.		Actual Owner		reuse setOutput()
deleteOutput	Deletes the output data of the task.		Actual Owner		write new
setFault	Set the fault data of the task.		Actual Owner		
deleteFault	Deletes the fault name and fault data of the task.		Actual Owner		
getInput	Get the data for the part of the task's input message.		Potential Owners Actual owner Business Administrator		
getOutput	Get the data for the part of the task's output message.		Actual Owner Business Administrator		
getFault	Get the fault data of the task.		Actual Owner Business		

			Administrator		
getMyTaskAbstracts			Any		write new
getMyTasks			Any		can not reuse getAvalaibleTasks () because task object are different
query			N/A		write new
activate	Activate the task,	set the task to status Ready.	Business Administrator		write new
nominate	Nominate an organization entity to process the task.	If it is nominated to one person then the new state of the ask is Reserved. If it is nominated to several people then the new state of the task is Ready. This can only be performed when the task is in the state Created.	Business Administrator		write new
setGenericHumanRole	Replace the organizational assignment to the task in one generic human role.		Business Administrator		write new

TMS Server

Tempo projects: Tempo-tms-Service

TMSRequestServer: This class implements task operation logic, will call underlying persistence layer to access data and security service to do authentication/authorization.

Full Name: org.intalio.tempo.workflow.tms.server.TMSServer

Implementations Note: The most big and hardest class for this part, will add pretty much interfaces. Will reuse some old code if operation logic is similar

Main Interfaces:

Please see operations tabels in previous diagram.

Marshaller/Unmarshaller

Task Marshaller/ummarshaller is responsible for marshalling/unmarshalling webservice response/request message.

Task Marshaller/ummarshaller

TaskMarshaller: Mashall the Task Object, call underlying xmlbean marshaller

Tempo projects: Tempo-tms-common

Full Name: org.intalio.tempo.workflow.task.xml.TaskMarshaller

Implementations Note: Since web service xmlbean marshaller will change a lot, so this class won't reuse much current code.

Main Interfaces:

Interface	Description	Implementation
marshalFullTask	Marsall task object	call xmlbean marshaller
marshalTaskInput	Marsall task input	call xmlbean marshaller
marshalTaskOutput	Marsall task output	call xmlbean marshaller
marshalTaskMetadata	Marsall task metadata	call xmlbean marshaller

TaskUnmarshaller: unmarshall the task request

Tempo projects: Tempo-tms-common

Full Name: org.intalio.tempo.workflow.task.xml.TaskMarshaller

Implementations Note: Since web service and xmlbean unmarshaller change a lot, so this class won't reuse much current

Main Interfaces:

Interface	Description	Implementation
unmarshalFullTask	unMarsall task object	call xmlbean unmarshaller
unmarshalTaskInput	unMarsall task input	call xmlbean marshaller
unmarshalTaskOutput	unMarsall task output	call xmlbean marshaller
unmarshalTaskMetadata	unMarsall task metadata	call xmlbean marshaller

OMMarshaller: marshall the task request

Tempo projects: Tempo-tms-common

Full Name: org.intalio.tempo.workflow.util.xml

Implementations Note: This class serve as util call,so won't change a lot.

Main Interfaces:

OMUnmarshaller: unmarshall the task response

Tempo projects: Tempo-tms-common

Full Name: org.intalio.tempo.workflow.util.xml

Implementations Note: This class serve as util class,so won't change a lot.

Main Interfaces:

TMSRequestProcessor: implementation of tms,also unmarshall the task response

Tempo projects: Tempo-tms-service

Full Name: org.intalio.tempo.workflow.tms.server

Main Interfaces:

see above interface talbe

TMSResponseMarshaller: marshall the service response

Tempo projects: Tempo-tms-service

Full Name: org.intalio.tempo.workflow.tms.server

Implementations Note: A very small class, just make a instantable marshaler callser

Main Interfaces:

XmlBean unmarshaller

Tempo projects: Tempo-tms-axis

classes: will be created from web service definition.

Full Name: com.intalio.bpms.workflow.taskManagementServices2008xxxx.xxxx

Implementations Note: create from new task web service

Main Interfaces: Generated from web service definition.

Comments

Site powered by a free **Open Source Project / Non-profit License** ([more](#)) of [Confluence - the Enterprise wiki](#).

[Learn more](#) or [evaluate Confluence for your organisation](#).

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 1.4.1 Build:#212 Jun 02, 2005) - [Bug/feature request](#) - [Contact Administrators](#)