

Activiti

Demo Activiti

Demo

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:activiti="http://activiti.org/bpmn"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:omgdc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:omgdi="http://www.omg.org/spec/DD/20100524/DI"
typeLanguage="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.w3.org/1999/XPath"
targetNamespace="http://www.activiti.org/test">
  <process id="hello" name="hello" isExecutable="true">
    <!-- -->
    <startEvent id="start" name="Start" ></startEvent>
    <!-- serviceTask me.likeyao.activiti.demo.HelloWorld execute hello world -->
    <serviceTask id="helloworld" name="helloworld"
activiti:class="me.likeyao.activiti.demo.HelloWorld"/>
    <!-- -->
    <endEvent id="end" name="End"></endEvent>
    <!-- serviceTask -->
    <sequenceFlow id="sid-1" sourceRef="start"
targetRef="helloworld"></sequenceFlow>
    <!-- serviceTask -->
    <sequenceFlow id="sid-3" sourceRef="helloworld" targetRef="end"></sequenceFlow>
  </process>
</definitions>
```



demo

```
public class App {
    public static void main(String[] args) {
        //
        ProcessEngine processEngine = ProcessEngines.getDefaultProcessEngine();
        //

        processEngine.getRepositoryService().createDeployment().addClasspathResource("hello.bpmn20.xml");
        //
        processEngine.getRuntimeService().startProcessInstanceByKey("hello");
    }
}

public class HelloWorld implements JavaDelegate{
    public void execute(DelegateExecution execution) throws Exception {
        System.out.println("Hello world!");
    }
}
```

Demo

serviceTask

Hello world

5.22.0

2242271-4becaf2056d3c51f.webp

Image not found or type unknown

ProcessEngine

ProcessEngine

ProcessEngine Activiti

ProcessEngine

getXXXService

Service

Demo

- RepositoryService
- RuntimeService

ProcessEngineConfiguration

```
public ProcessEngine buildProcessEngine() {  
    init();  
    return new ProcessEngineImpl(this);  
}
```

```
protected void init() {  
    initConfigurators();  
    configuratorsBeforeInit();  
    initProcessDiagramGenerator();  
    initHistoryLevel();  
    initExpressionManager();  
    initDataSource();  
    initVariableTypes();  
    initBeans();  
    initFormEngines();  
    initFormTypes();  
    initScriptingEngines();  
    initClock();  
    initBusinessCalendarManager();  
    initCommandContextFactory();  
    initTransactionContextFactory();  
    initCommandExecutors();  
    initServices();  
    initIdGenerator();  
    initDeployers();  
    initJobHandlers();  
    initJobExecutor();  
    initAsyncExecutor();  
    initTransactionFactory();  
    initSqlSessionFactory();  
    initSessionFactory();  
    initJpa();  
    initDelegateInterceptor();  
    initEventHandlers();  
    initFailedJobCommandFactory();  
    initEventDispatcher();  
    initProcessValidator();  
    initDatabaseEventLogging();  
}
```

```
configuratorsAfterInit();  
}
```

ProcessEngineConfigurator

```
public interface ProcessEngineConfigurator {  
    //  
    void beforeInit(ProcessEngineConfigurationImpl processEngineConfiguration);  
    //  
    void configure(ProcessEngineConfigurationImpl processEngineConfiguration);  
    //  
    int getPriority();  
}
```

init initConfigurators ServiceLoader ProcessEngineConfigurator configuratorsBeforeIn

xml Activiti

RepositoryService

2242271-bc6f111998288acc.webp

Image not found for RepositoryService

Demo

```
processEngine.getRepositoryService().createDeployment().addClasspathResource("hello.bpmn20.xml");
```

createDeployment DeploymentBuilder DeploymentBuilder xml byte

```
public DeploymentBuilder addInputStream(String resourceName, InputStream inputStream) {  
    ...  
    byte[] bytes = IoUtil.readInputStream(inputStream, resourceName);  
    ResourceEntity resource = new ResourceEntity();  
    resource.setName(resourceName);  
    resource.setBytes(bytes);  
    deployment.addResource(resource);  
    return this;  
}
```

DeploymentBuilder deploy RepositoryService deploy

```
public Deployment deploy() {
    return repositoryService.deploy( this);
}
```

CommandExecutor

RepositoryService deploy CommandExecutor

```
public Deployment deploy(DeploymentBuilderImpl deploymentBuilder) {
    return commandExecutor.execute(new DeployCmd<Deployment>( deploymentBuilder));
}
```

2242271-e6a84ab68b549bf4.webp

Image not found in CommandExecutor

Activiti Command DeployCmd CommandExecutor CommandInterceptor

CommandInterceptor

CommandContextInterceptor	CommandContext
LogInterceptor	Debug
JtaTransactionInterceptor	Jta

“ activiti-spring SpringTransactionInterceptor Spring

CommandExecutor ProcessEngineConfigurationImpl initCommandExecutors

```
protected void initCommandExecutors() {
    initDefaultCommandConfig();
    initSchemaCommandConfig();
    initCommandInvoker();
    initCommandInterceptors();
    initCommandExecutor();
}
```

ProcessEngineConfigurationImpl customPreCommandInterceptors customPostCommandInterce

```
protected void initCommandInterceptors() {
```

```
if (commandInterceptors==null) {
    commandInterceptors = new ArrayList<CommandInterceptor>();
    if (customPreCommandInterceptors!=null) {
        commandInterceptors.addAll( customPreCommandInterceptors);
    }
    commandInterceptors.addAll( getDefaultCommandInterceptors());
    if ( customPostCommandInterceptors!=null) {
        commandInterceptors.addAll( customPostCommandInterceptors);
    }
    commandInterceptors.add( commandInvoker);
}
}
```

“ pre post Activiti getDefaultCommandInterceptors()

CommandInvoker CommandInterceptor Command

```
public class CommandInvoker extends AbstractCommandInterceptor {
    @Override
    public <T> T execute( CommandConfig config, Command<T> command) {
        return command.execute( Context. getCommandContext());
    }
}
```

CommandContext

2242271-dbf9c4a66c79d3c9.webp

Image not found CommandContext

CommandContext Activiti Command SessionFactory

```
sessionFactories = processEngineConfiguration.getSessionFactories();
```

SessionFactory Session Session Activiti

ProcessDefinitionEntityManager	
ExecutionEntityManager	

DefaultHistoryManager	

CommandContext

CommandConext CommandContextInterceptor finally

```

public <T> T execute(CommandConfig config, Command<T> command) {
    //          CommandContext
    CommandContext context = Context.getCommandContext();
    boolean contextReused = false;
    //          CommandContext
    //1 CommandConfig          CommandContext
    //2          CommandConext
    //3          CommandConext
    if (!config.isContextReusePossible() || context == null || context.getException() !=
null) {
        context = commandContextFactory.createCommandContext(command);      }
    else {
        contextReused = true;
    }

    try {
        //          CommandContext
        Context.setCommandContext(context);
        Context.setProcessEngineConfiguration(processEngineConfiguration);
        //    interceptor  CommandInvoker      Context.getCommandContext()      CommandContext
        return next.execute(config, command);
    } catch (Exception e) {
        //
        context.exception(e);
    } finally {
        try {
            // CommandContext
            if (!contextReused) {
                context.close();
            }
        } finally {
            //
            Context.removeCommandContext();
        }
    }
}

```

```
Context.removeProcessEngineConfiguration();
Context.removeBpmnOverrideContext();
}
}

return null;
}
```

Activiti Command Command CommandContext true

DeployCmd DeploymentEntityManager DeploymentEntity

```
commandContext.getDeploymentEntityManager().insertDeployment(deployment);
```

DeploymentManager

```
commandContext.getProcessEngineConfiguration().getDeploymentManager().deploy(deployment, deploymentSettings);
```

DeploymentEntityManager

[2242271-eb4810b9b4a64a04.webp](#)

Image not found for DeploymentEntityManager

DeploymentEntityManager deploy Deployer deploy Activiti Deployer BpmnDeployer

DeploymentEntityManager Bpmn

“ Activiti

xml

“deploy”

ProcessDefinitionEntity

BpmnDeployer

[2242271-68d44c88944fc225.webp](#)

Image not found for BpmnDeployer

BpmnDeployer deploy

```
public void deploy(DeploymentEntity deployment, Map<String, Object> deploymentSettings) {
```



```

...
BpmnParse bpmnParse =
bpmnParser.createParse().sourceInputStream(inputStream).setSourceSystemId(resourceName).deployments
bpmnParse.execute();
for (ProcessDefinitionEntity processDefinition: bpmnParse.getProcessDefinitions())
{
    if (deployment.isNew()) {
        ProcessDefinitionEntity latestProcessDefinition = ...
        if (latestProcessDefinition != null) {
            processDefinitionVersion = latestProcessDefinition.getVersion() +
1;
        }else{
            processDefinitionVersion = 1;
        }
        processDefinition.setId(idGenerator.getNextId());
        dbSqlSession.insert(processDefinition);
    }
    ...
}
}

```

- BpmnParser BpmnParse
- BpmnParse execute inputStream ProcessDefinitionEntity
- ProcessDefinitionEntity

BpmnParse

[2242271-c0d8c16d9fbf42fc.webp](#)

Image not found or is unknown

BpmnParse execute xml ProcessDefinitionEntity

```

public BpmnParse execute() {
    //xml->bpmnModel
    bpmnModel = converter.convertToBpmnModel(streamSource, validateSchema, enableSafeBpmnXml,
encoding);
    //bpmnModel-> ProcessDefinitionEntity
    transformProcessDefinitions();
}

```

```
protected void transformProcessDefinitions() {
    for (Process process : bpmnModel.getProcesses()) {
        bpmnParserHandlers.parseElement(this, process);
    }
}
```

- Bpmn BpmnXMLConverter
- PVM BpmnParseHandlers

Bpmn

[2242271-0d7054f46acfb7dd.webp](#)

Image not found or type unknown

PVM

[2242271-ce5026e930258c97.webp](#)

Image not found or type unknown

Bpmn xml PVM Bpmn ServiceTask StartEvent PVM ActivityImpl Ser

demo RuntimeService

```
processEngine.getRuntimeService().startProcessInstanceByKey("hello");
```

startProcessInstanceByKey StartProcessInstanceCmd

```
public class StartProcessInstanceCmd<T> implements Command<ProcessInstance>, Serializable
{
    ...
    public ProcessInstance execute(CommandContext commandContext) {
        //
        ProcessDefinitionEntity processDefinition = ...
        //
        ExecutionEntity processInstance =
processDefinition.createProcessInstance(businessKey);
        //
        processInstance.start();
        return processInstance;
    }
    ...
}
```

```
}
```

StartProcessInstanceCmd

ProcessDefinitionEntity

ExecutionEntity

2242271-5a8bc87dc9fd0f46.webp

Image not found: ExecutionEntity

ExecutionEntity

- PVM ExecutionEntity single start
- VariableScope ExecutionEntity
- ProcessInstance ExecutionEntity ProcessDefinitionEntity
- PersistentObject ExecutionEntity

ExecutionEntity activity activity

ExecutionEntity activity

StartProcessInstanceCmd ExecutionEntity start

```
public void start() {  
    performOperation(AtomicOperation.PROCESS_START);  
}
```

Activiti

PVM

PVM

ActivityImpl TransitionImpl ActivityBehavior

- ActivityImpl ActivityImpl ActivityImpl
- TransitionImpl TransitionImpl source target
- ActivityBehavior ActivityImpl ActivityBehavior ActivityBehavior

ActivityImpl TransitionImpl ActivityBehavior

ExecutionEntity

AtomicC

```
AtomicOperation PROCESS_START = new AtomicOperationProcessStart();  
AtomicOperation PROCESS_START_INITIAL = new AtomicOperationProcessStartInitial();  
AtomicOperation PROCESS_END = new AtomicOperationProcessEnd();  
AtomicOperation ACTIVITY_START = new AtomicOperationActivityStart();  
AtomicOperation ACTIVITY_EXECUTE = new AtomicOperationActivityExecute();  
AtomicOperation ACTIVITY_END = new AtomicOperationActivityEnd();  
AtomicOperation TRANSITION_NOTIFY_LISTENER_END = new  
AtomicOperationTransitionNotifyListenerEnd();  
AtomicOperation TRANSITION_DESTROY_SCOPE = new AtomicOperationTransitionDestroyScope();  
AtomicOperation TRANSITION_NOTIFY_LISTENER_TAKE = new  
AtomicOperationTransitionNotifyListenerTake();  
AtomicOperation TRANSITION_CREATE_SCOPE = new AtomicOperationTransitionCreateScope();
```

```
AtomicOperation TRANSITION_NOTIFY_LISTENER_START = new
AtomicOperationTransitionNotifyListenerStart();

AtomicOperation DELETE_CASCADE = new AtomicOperationDeleteCascade();
AtomicOperation DELETE_CASCADE_FIRE_ACTIVITY_END = new
AtomicOperationDeleteCascadeFireActivityEnd();
```

ExecutionEntity start PROCESS_START PROCESS_START

- start ExecutionListener notify
- ActivitiEntityWithVariablesEvent ActivitiProcessStartedEvent
- PROCESS_START_INITIAL

PROCESS_START_INITIAL

- start ExecutionListener notify
- ACTIVITY_EXECUTE

Demo AtomicOperation

- ACTIVITY_EXECUTE activity behavior
- TRANSITION_NOTIFY_LISTENER_END activity end ExecutionListener
- TRANSITION_NOTIFY_LISTENER_TAKE ExecutionListener
- TRANSITION_NOTIFY_LISTENER_START activity start ExecutionListener

Demo ServiceTask helloworld ACTIVITY_EXECUTE activity behavior

```
public class AtomicOperationActivityExecute implements AtomicOperation {
    public void execute(InterpretableExecution execution) {
        ...
        ActivityImpl activity = (ActivityImpl) execution.getActivity();
        ActivityBehavior activityBehavior = activity.getActivityBehavior();
        activityBehavior.execute(execution);
        ...
    }
}
```

ServiceTask ServiceTaskJavaDelegateActivityBehavior execution

```
public void execute(ActivityExecution execution) throws Exception {
    //execution    me.likyao.activiti.demo.HelloWorld
    execute((DelegateExecution) execution);
    //
```

```
leave(execution);
}
```

leave

```
bpmnActivityBehavior.performDefaultOutgoingBehavior(execution);
```

performDefaultOutgoingBehavior

activity

Demo

```
protected void performOutgoingBehavior( ActivityExecution execution,
        boolean checkConditions, boolean throwExceptionIfExecutionStuck,
        List<ActivityExecution> reusableExecutions) {

    if ( transitionsToTake.size() == 1) {
        execution.take( transitionsToTake.get(0));
    }

}
```

take

AtomicOperation

```
public class ExecutionEntity{
    ...
    public void take(PvmTransition transition, boolean fireActivityCompletionEvent) {
        ...
        setActivity(( ActivityImpl) transition.getSource());
        setTransition(( TransitionImpl) transition);
        performOperation( AtomicOperation. TRANSITION_NOTIFY_LISTENER_END);
        ...
    }
    ...
}
```

AtomicOperation

AtomicOperation

UserTask

single

RuntimeService

ExecutionListener

AtomicOperation

Activiti

BPMN

Activit

Activiti

Revision #2

Created 16 March 2020 14:27:05 by

Updated 27 March 2020 08:24:26 by