# **APPROVAL PROCESSING**

An approval process automates how records are approved in Salesforce. An approval process specifies each step of approval, including from whom to request approval and what to do at each point of the process.

We use the Apex process classes to create approval requests and process the results of those requests:

* **ProcessRequest Class**
* **ProcessResult Class**
* **ProcessSubmitRequest Class**
* **ProcessWorkItemRequest Class**

**ProcessRequest Class:**

The ProcessRequest class is the parent class for the ProcessSubmitRequest and ProcessWorkitemRequest classes. Use the ProcessRequest class to write generic Apex that can process objects from either class.

**Usage:** The request must be instantiated via the child classes, ProcessSubmitRequest and ProcessWorkItemRequest.

**ProcessRequest Methods:**

The following are methods for ProcessRequest. All are instance methods.

* **getComments():** Returns the comments that have been added previously to the approval request.
* **getNextApproverIds():** Returns the list of user IDs of user specified as approvers.
* **setComments(comments):** Sets the comments to be added to the approval request.
* **setNextApproverIds(nextApproverIds):** If the next step in your approval process is another Apex approval process, you specify exactly one user ID as the next approver. If not, you cannot specify a user ID and this method must be null.

**ProcessResult Class:**

After you submit a record for approval, use the ProcessResult class to process the results of an approval process.

**Usage:** A ProcessResult object is returned by the process method. You must specify the Approval namespace when creating an instance of this class.

For example:

Approval.ProcessResult result = Approval.process(req1);

**ProcessResult Methods:**

The following are methods for ProcessResult. All are instance methods.

* **getEntityId():** The ID of the record being processed.
* **getErrors():** If an error occurred, returns an array of one or more database error objects including the error code and description.
* **getInstanceId():** The ID of the approval process that has been submitted for approval.
* **getInstanceStatus():** The status of the current approval process. Valid values are: Approved, Rejected, Removed or Pending.
* **getNewWorkitemIds():** The IDs of the new items submitted to the approval process. There can be 0 or 1 approval processes.
* **isSuccess():** A Boolean value that is set to true if the approval process completed successfully; otherwise, it is set to false.

**ProcessSubmitRequest Class:**

Use the ProcessSubmitRequest class to submit a record for approval.

**Usage:** You must specify the Approval namespace when creating an instance of this class. The constructor for this class takes no arguments.

For example:

Approval.ProcessSubmitRequest psr = new Approval.ProcessSubmitRequest();

**ProcessSubmitRequest Methods:**

The following are methods for ProcessSubmitRequest. All are instance methods.

* **getObjectId():** Returns the ID of the record that has been submitted for approval. For example, it can return an account, contact, or custom object record.
* **getProcessDefinitionNameOrId():** Returns the developer name or ID of the process definition.
* **getSkipEntryCriteria():** If getProcessDefinitionNameOrId() returns a value other than null, getSkipEntryCriteria() determines whether to evaluate the entry criteria for the process (true) or not (false).
* **getSubmitterId():** Returns the user ID of the submitter requesting the approval record. The user must be one of the allowed submitters in the process definition setup.
* **setObjectId(recordId):** Sets the ID of the record to be submitted for approval. For example, it can specify an account, contact, or custom object record.
* **setProcessDefinitionNameOrId(nameOrId):** Sets the developer name or ID of the process definition to be evaluated.
* **setSkipEntryCriteria(skipEntryCriteria):** If the process definition name or ID is not null, setSkipEntryCriteria() determines whether to evaluate the entry criteria for the process (true) or not (false).
* **setSubmitterId(userID):** Sets the user ID of the submitter requesting the approval record. The user must be one of the allowed submitters in the process definition setup. If you don’t set a submitter ID, the process uses the current user as the submitter.

**ProcessWorkitemRequest Class:**

Use the ProcessWorkitemRequest class for processing an approval request after it is submitted.

**Usage:** You must specify the Approval namespace when creating an instance of this class. The constructor for this class takes no arguments.

For example:

Approval.ProcessWorkitemRequest pwr = new Approval.ProcessWorkitemRequest();

**ProcessWorkitemRequest Methods:**

The following are methods for ProcessWorkitemRequest. All are instance methods.

* **getAction():** Returns the type of action already associated with the approval request. Valid values are: Approve, Reject, or Removed.
* **getWorkitemId():** Returns the ID of the approval request that is in the process of being approved, rejected, or removed.
* **setAction(actionType):** Sets the type of action to take for processing an approval request.
* **setWorkitemId(id):** Sets the ID of the approval request that is being approved, rejected, or removed.