



# Predefined Templating Variables

## On this page:

- [Forming Variable Names](#)
- [Base Names](#)
- [Info Classes for Templating Variables](#)
- [Examples](#)

## Related pages:

- [HTTP Request Actions and Templates](#)
- [Email Templates](#)

This topic describes the predefined variables that can be used in an HTTP request template or in an Email template.

The policy engine substitutes the value of the variable in the context of the triggering event(s) when it sends the actual request or email.

## Forming Variable Names

To form a predefined variable name, combine the base name of the variable with a field of its corresponding info class.

For example, to use the account name as a variable in a template, combine the base name "account" with the "name" field of the EntityInfo class to form the variable `${account.name}`.

Or, to use the trigger time of an action as a variable, combine the base name "action" with the "triggerTime" field of the ActionInfo class to form the variable `${action.triggerTime}`.

You can chain the segments of the variable name where the info class field is itself the type of a base name. For example, to use the name of the application in which the latest triggering event occurred, combine the "latestEvent" base name with the "application" field of the EventInfo class (from the latestEvent) with the name field of the EntityInfo class (from the application) to form `${latestEvent.application.name}`.

## Base Names

These are the base names for the predefined variables with their corresponding info classes:

Base Name	Description	Info Class
account	Account in which the action was triggered	EntityInfo
policy	Policy that triggered the action	PolicyInfo
action	Triggered action	ActionInfo
topSeverity	INFO, WARN or ERROR	NotificationSeverity
topSeverityImage	Severity image or icon	ImageInfo
notificationConfigText	Email actions only	String
controllerUrl	URL of the controller	String
appDynamicsIcon		ImageInfo
appDynamicsLogo		ImageInfo
latestEvent	Most recent triggering event	EventInfo
fullEventList	List of events that triggered the action	List<EventInfo>
fullEventsByTypeMap	List of events that triggered the action, grouped by type	Map<String, List<EventInfo>>

clampLimit	Optional setting, limit on how many triggering events to display	int
clamped	True if clamp limit is set	boolean
clampedEventList	// most recent triggering events if clamped	List<EventInfo>
clampedEventsByTypeMap	// most recent triggering events if clamped, grouped by type	Map<String, List<EventInfo>>

## Info Classes for Templating Variables

Use the fields in the appropriate class, based on the table above, to form the variable to use in a template.

```

class EventInfo {
    EventType eventType
    long id
    String guid
    String eventTypeKey
    Date eventTime
    String displayName
    String summaryMessage
    String eventMessage
    EntityInfo application
    EntityInfo tier
    EntityInfo node
    List<EntityInfo> affectedEntities
    boolean healthRuleEvent
    EntityInfo healthRule // * Only defined when healthRuleEvent == true
    EntityInfo incident // * Only defined when healthRuleEvent == true
    boolean healthRuleViolationEvent
    NotificationSeverity severity
    ImageInfo severityImage
    boolean btPerformanceEvent // * true when eventType matches one of the BT performance event types
    String deepLink
}

class ImageInfo {
    String name
    String fileName
    String mimeTypeContentRef
    String deepLink
}

class EntityInfo {
    EventType entityType
    String entityTypeDisplayName
    long id
    String name
}

class PolicyInfo {
    EventType entityType
    String entityTypeDisplayName
    long id
    String name
    boolean digest
    digestDurationInMins
}

class ActionInfo {
    EventType entityType
    String entityTypeDisplayName
    long id
    String name
    Date triggerTime
}

enum NotificationSeverity { INFO, WARN, ERROR }

```

## Examples

**HTTP request example for most recent triggering event:**

```
http://myController:8080/controller/rest/applications/${latestEvent.application.name}/nodes
/${latestEvent.node.name}
```

### HTTP request example iterating through a list of triggering events:

```
#foreach(${event} in ${fullEventList})
  http://myController:8080/controller/rest/applications/${event.application.name}/nodes/${event.node.
name}
#end
```

### Email HTML body example:

```
<h1>Summary of events occurring during the ${policy.digestDurationInMins}+ minute(s) prior to
${action.triggerTime}</h1>

<table>

#foreach(${eventList} in ${fullEventsByTypeMap.values()})
  #foreach(${event} in ${eventList})

    <tr>
      <td>
        <!-- Event icon -->
        
      </td>
      <td>
        <!-- Event name with event link -->
        <a href="${event.deepLink}">${event.displayName}</a>
      </td>
      <td>
        <!-- Event message -->
        ${event.eventMessage}
      </td>
    </tr>

  #end
#end

</table>
```

If these predefined variables do not meet all your needs, you can define your own custom variables for use in these templates in the Create Template window. Do not use any of the predefined variable names for your custom variables.