

HTTP Request Actions and Templates

On this page:

- [Required Permissions](#)
- [Access HTTP Action Templates](#)
- [Create or Modify an HTTP Request Template](#)
- [Test the Template](#)

Related pages:

- [Policies](#)
- [Actions](#)
- [Configuration Import and Export API](#)
- [Alert and Respond](#)
- [Predefined Templating Variables](#)

An HTTP request action sends an HTTP request in response to an event. These types of actions allow you to integrate AppDynamics' policies with third-party HTTP APIs.

You create an HTTP request action using an HTTP request template. The template describes the HTTP request and is reusable by different HTTP request actions within an AppDynamics account.

After you create a template, users in the Controller UI can create actions that use it. The template appears as an option after the user chooses the *Make an HTTP Request* option in the Create Action window and access the configure action screen.

Required Permissions

To create or modify HTTP Request Templates, users need the account-level *Configure HTTP Request Templates* permission.

Access HTTP Action Templates

To access existing templates or to create new ones. Click **Alert & Respond** in the menu bar, and then **HTTP Request Templates**.

From there, you can:

- Click an existing template from the HTTP Request Templates list to view, edit or delete it, or
- Click the **New +** icon to create a template.

Create or Modify an HTTP Request Template

When creating or modifying templates, click the info icons in the template to get information about how to complete the configuration. The following subsections clarify some details in the template configuration that may require additional information.

After you have created a template, click the **Save** button at the bottom to save it.

Custom Templating Variables

You can use variables that replace values in the URL path and payload when the HTTP request is sent. Foreach loops are supported.

The template already knows a set of predefined variables, which are described in [Predefined Templating Variables](#). Check this list before you create any custom templating variables. Chances are the variable you want to use has already been defined.

You can optionally configure custom variables if the predefined variables do not meet all your needs. When a predefined variable and a custom variable are both configured with the same variable name, the template uses the predefined variable.

The template uses Apache Velocity version 1.7 to process the variables. See the [Velocity User Guide](#) for details about usage.

Request URL

Enter the URL for the request in the Raw URL field and select the URL encoding scheme from the pulldown menu. Only UTF-8 and ISO_8858-1 are supported. Verify that the Encoded URL is the exact request to send.

Request URL ⓘ

Method	GET ▼
Raw URL	<input type="text" value="http://myController:8080/controller/rest/applications/\${latestEvent.application.name}/nodes/\${latestEvent"/>
URL Encoding	UTF-8 ▼
Encoded URL	http://mycontroller:8080/controller/rest/applications/\${latestEvent.application.name}/nodes/\${latestEvent.node.name}

Authentication

You can define basic authentication for your HTTP request. Select BASIC authentication type from the Type drop-down list only if the communication is encrypted. If the communication is not encrypted, we recommend not to use any authentication. After selecting the authentication type, specify the authentication username and password.

Custom Request Headers

You can define custom headers for the HTTP requests. Custom Request Headers can contain custom templating variables or predefined templating variables encoded as `$(VARIABLE_NAME)`.

For a complete list of predefined variables, see [Predefined Templating Variables](#). Check this list before you create any custom templating variable. See the [Velocity User Guide](#) for details about usage.

Payload

To include a payload in your HTTP request, define the MIME Type from the drop-down list. You can encode the payload using UTF-8 or ISO-8859-1 encoding.

Response Handling Criteria

Success or Failure

If you do not specify any response-handling criteria for either success or failure, the HTTP response is always success.

Otherwise:

- Failure criteria are evaluated before success criteria. As soon as a match is found on a fail criterion, the response is set to failure and the remaining fail match conditions, if there are any, are not evaluated.
- If no failure criteria are matched, the success criteria are then evaluated.

If no success criteria are configured, the response will always be success as long as no fail criteria were previously matched.

Otherwise:

- If you specify at least one success criterion, there must be a match on a success criterion for the response to be success. In other words, if any success criteria are configured but not matched, the response will be failure.

Both failure and success criteria are considered in the order in which they appear in the template.

Criteria With and Without Payload

You can set separate criteria for the same status code. For example, you can set one criterion on a status with payload as success and another criterion on the same code without payload as failure.

Or, you can set separate criteria for the same status code with payload of different content types. For example, payload of type `application/xml` and `application/xhtml+xml` might be success and payload of other types could be failure.

Content Types for Payload

You can specify the content type for the payload using the dropdown list that appears for requests for which the Expect Payload checkbox is checked.

If you do not know the content type but the request expects payload, use `/*/*` to specify all content types.

The following configuration returns a success response for a 200 status code with an XML payload. Any other code will cause the request to return a failure response.

Response Handling Criteria ⓘ

Failure Criteria

Evaluate the following criteria first. Set action result to "failure" when the HTTP response matches any of the following.

Status Code	Expect Payload	Content Type	
-------------	----------------	--------------	--

[+ Add Failure Criteria](#)

Success Criteria

Evaluate the following only if there is no match in the above failure criteria. Set the action result to "success" if there is a match in the following, else set it to "failure".

Status Code	Expect Payload	Content Type	
200	<input checked="" type="checkbox"/>	application/xml	-

[+ Add Success Criteria](#)

One Request Per Event

Several separate events, or separate occurrences of the same event, could potentially invoke the same HTTP action.

The One Request Per Event setting controls whether the action bundles the HTTP requests.

If this check box is clear, the HTTP request is sent once every minute, no matter how many events triggered the action within that time frame. This is the default behavior.

If this check box is checked, the request is sent every time an event triggers it. In this case, if ten events trigger the action, the HTTP request is sent ten times, even if all those events occurred within a single minute.

You might want to send one request per event, checkbox checked, if you are trying to log tickets and your ticketing system API does not support bulk create. However, if it does, you probably want a single bulk request every minute, checkbox clear.

Event Clamp Limit

If you have not checked the One Request Per Event checkbox and the list of triggering events could be large, you might want to limit the display of the events that triggered the action in the response. The clamp limit is the n most recent triggering events. The default -1 means no limit. This setting is ignored if *One Request Per Event* is checked.

Timeouts and Redirects

- Connect timeout: The maximum number of milliseconds to wait for the request to reach the server.
- Socket timeout: The maximum number of milliseconds to wait to receive the response.
- Max Redirects: The maximum number of times that a single request can redirect.

Test the Template

You can test the template, by clicking *Test* at the bottom of the template configuration.

Configure Test

In the Template Test configuration window, specify the test variables to use for the test. These variables may be different from those that will be used in the real requests that will be automatically sent when the HTTP action is live.

Also, specify the type of events that trigger for the test.

The log level defaults to `INFO`, but you can change this using the dropdown Log Level list.

The following sample test template configuration simulates three `POLICY OPEN CRITICAL` events triggering this action. More triggers could be added.

HTTP Action Template Test



HTTP Request Template http

Log Level **INFO** ▼

Test Variables ⓘ

var1 applications

Event Type Trigger ⓘ

Event	Count
Slow requests - Slow ▼	3 ▲ ▼

+ Add Event Type

Cancel

Save

Run Test

Run Test

When you run the test, the request is sent when it is triggered. You get a transcript of the run.

If the results are not what you expect, you may need to modify the template test configuration, the HTTP action template, or both before you use the template in an action.