Integration with Web Load Testing and Monitoring Tools

Overview

Dynatrace integrates well with load testing and monitoring products, especially on the transactional / request level.

Dynatrace automatically starts a PurePath on a servlet, ASP.NET or web service when the Dynatrace HTTP trace tag is present.

The Dynatrace Community Webinars show how to use Dynatrace in load testing and also show how it uses tagged web requests to integrate with any HTTP-based load testing solution.

Integration with Web Load Testing

Web Load Testing is integrated directly with Dynatrace. For details, see Web Load Testing Integration.

Request Header

To use other load testing and monitoring tools for diagnosing web services in Dynatrace, add the following information to the HTTP headers generated by the load testing tool:

```
x-dynaTrace: VU=1;PC=.1;ID=4;NA=SearchPage
```

Below is a sample HTTP header. The final line of the example shows the added code.

```
POST /onca/soap?Service=AWSECommerceService
HTTP/1.1
Content-Type: text/xml; charset=utf-8
Host: soap.amazon.com
Content-Length: 2566
Connection: Keep-Alive
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5;
Windows NT)
x-dynaTrace: VU=1;PC=.1;ID=4;NA=ListResults
```

The following table describes the HTTP header string for Dynatrace diagnostics.

Key	Description	APMaaS Name	Usage

ID	The unique request **ID** (serial number). This string should be unique for one web request or a set of web requests that together make up a step/transaction execution. `ID=<Id>`	Transaction GUID	optional
PC	The **P**age **C**ontext contains information about what document in the currently processed page is loaded. The following syntax is recommended, though not required: • If it is a named frame, then the value starts with the frame name. • The document number, unique for the page, is appended after a period. If embedded documents are cached, this number need not be progressive. `PC=<FrameName>.<DocId>`		optional

VU	The unique number of the **V**irtual **U**ser that sends the request. `VU=<Id>`	Virtual User ID	optional
NA	The timer **NA**me of the request. This can be the timer or transaction name used in the load test script to identify the response time measure, or the document/page title, or any other human-readable URL encoded identifier for that document. `NA=<timername e>`	Transaction Name (SYM), Step Name (SaaS)	**required**
SI	The **S**ource **I**D can be used to identify the product that triggered the request: For example **WLT** (Web Load Testing), SYM (Dynatrace Synthetic Monitoring), BB (Backbone), or LM (Last Mile).	Source ID	optional
GR	**G**eographic **Re**gion, useful only for the Synthetic Monitoring solution. This contains arbitrary text.	Location	optional

AN	**A**gent **N**ame: the logical name of the Agent from which the request originated. This information is used by Synthetic Monitoring.	Agent Name	optional
SN	**S**cript **N**ame. This groups a set of requests that make up a multi-step transaction, for example making an online purchase.	Script Name	optional
TE	**TE**st Name is the name of the entire load test. It uniquely identifies a test run.	Test Name	optional

Response Header

```
HTTP/1.1 200 OK
x-dynaTrace:
RS=session20060620091846;PT=31;PA=1;PS=76562898
Content-Type: text/html
Transfer-Encoding: chunked
Date: Tue, 20 Jun 2006 07:19:16 GMT
Server: Apache-Coyote/1.1
```

All the attributes together uniquely define a single object request, i.e. a single PurePath.

Key	Description	Usage
RS	Name of the Dynatrace **R**ecorded **S**ession that is currently being recorded.	only set when session recording is active

PT	**P**urePath identifier – The **T**race number.	required
PA	**P**urePath identifier – The entry point **A**gent number.	required
PS	**P**urePath **S**erver identifier – The Dynatrace Server that captured this PurePath.	required
SP	**S**ystem **P**rofile – Set to the active System Profile if manual session recording is not active; i.e., if either continuous session recording or no session recording is active.	available

Viewing Tagged Web Requests in Dynatrace

Dynatrace groups the resulting web page requests specifically for load tests. See Tagged Web Requests for details.

Tagged Web Requests

Synchronizing Session Recording with Load Test Runs

Dynatrace Server offers both a web service and RESTful service API that can be used to record a Dynatrace session for the time of the load test run.
In earlier versions of Dynatrace the command line interface could be used via the `dtcmd` utility. You can still use this command line utility, but we recommend using the web-based APIs.

To get a list of all web-based APIs, browse to the Dynatrace Server web frontend, which by default is http://dtservername:8020.

From there, you have the option to obtain the SOAP/HTTP web service

WSDL (http://dtservername:8020/wsdl/ManagementServerService.wsdl) or interact with the RESTful interface via an HTML interface (http://dtserve rname:8020/rest/html/management/server) or via an XML-based interface (http://dtservername:8020/rest/management/server).

The HTML interface enables you to explore all options the RESTful interface offers. Clicking the XML link on each HTML page shows you the links to the RESTful service that should be used when accessing the service from an external tool. The response of the XML-based interface is also XML, which is easier to parse and analyze.

The RESTful interface makes it easy to interact with the Dynatrace Server. For synchronizing session recording with load test runs, you can use the RESTful calls to start and stop session recording for a specific System Profile. If your load testing tool offers a way to execute web requests at the beginning and at the end of the load test run, you can add the necessary web request calls to the Dynatrace Server.

See Start Session Recording and Stop Session Recording for details on how to start and stop session recording using the REST Interfaces.

WebAPI Tests

Error rendering macro 'excerpt-include' : User 'null' does not have permission to view the page 'Web API Tests'.