VMware vFabric AppInsight Developer's Guide

vFabric AppInsight 1.1

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.

EN-000853-01

You can find the most up-to-date technical documentation on the VMware Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

VMware vFabric AppInsight Developer's Guide

The *VMware vFabric AppInsight Developer's Guide* describes an API to enable you to interact with the AppInsight datastore of your monitored applications, outside of the user interface.

Intended Audience

This information is intended for anyone who wants to use automated processes to retrieve data from the AppInsight datastore or perform certain topology actions. It is assumed that users are familiar with the REST concepts and the JSON serialization format.

AppInisght API Overview 1

The vFabric AppInsight API provides you with the means to retrieve data from your AppInsight-monitored applications without launching the AppInsight application. You can also add objects to your application's topology.

The AppInsight API is created using RESTful services. Responses are provided using JSON serialization format.

Using the AppInsight API 2

Use this information to interact with the AppInsight API.

General Requirements

Before you use the AppInsight API, note the following requirements.

- All calls must be authenticated
- You must use the `GET` HTTP method for retrieving data, and `POST` for adding data.
- The path variable is always required. It is used to focus on a single resource in the resource list.

  For example,

  `/applications/+`*`APP_NAME`*`/metrics`

  where *APP_NAME* is the path variable.
- All resources are defined using the plural form, for example applications, metrics, notifications, and so on.
- Query parameters may be required or optional. They are primarily used to provide a filter, or to narrow results to match criteria.

  For example,

  `/applications/+myAPP/metrics/health/average`

  where `indicator` is the query parameter.

  When a query parameter is optional, the default value is used if you do not specify a value.
- When traversing applications, to retrieve data for a specific application, you must indicate the application by using `+`.

  For example,

  `/applications/+myAPP/notifications`
- The URL that you use with the API cannot end with `/`.

Supported HTTP Methods

There are two HTTP methods supported by this API.

Table 2-1. Supported HTTP Methods

Method	Usage
GET	Retreiving data
POST	Adding data

Authentication

This API's authentication is based on basic HTTP or HTTPS authentication.

No session objects are created when you connect to the API. A client must send the basic HTTP headers each time that it connects to the API.

Responses

Responses to API calls differ, depending on whether the response relates to a single object or to a collection.

This is a sample response for calls that return a single object.

```
{ "meta" : {
                    "message" : "" // reserved for API messages
          },
          "result" : {
                           // key-value fields
          }
}
```

This is a sample response for calls that return a collection object.

```
{ "meta" : {
                   "message" : "" // reserved for API messages
         },
         "result" : [
                 {
                           // key-value fields for each result
                 }
         ]
}
```

Error Messages

Errors include a corresponding HTTP status code.

Error Type	HTTP Status Code
Bad user input	BAD_REQUEST (400)
Unauthorized operation	FORBIDDEN (403)
Unrecognized error	SERVER_ERROR (500)

Using Endpoints to Retrieve Data

You use endpoints to retrieve data for a specific object from your AppInsight-monitored applications.

You can append the following endpoints to a URL to retrieve data. You must include an item from the Data to Retrieve column for the request to be answered.

Table 2-2. Endpoints for Retrieving Data

Resource	General	Data to Retrieve
applications	/	■ notifications ■ metrics ■ average ■ overtime
applications/+APP_NAME	/ ■ components ■ groups ■ infrastructure ■ tiers ■ transactions	■ metrics ■ average ■ overtime ■ notifications
infrastructure		notifications

Following are examples of requests using endpoints.

```
https://appinsight IP:8443/am-apm-web/resources/applications/notifications
https://appinsight IP:8443/am-apm-web/resources/applications/+petclinic/notifications?
startTime=123498098&endTime=123534209
https://appinsight IP:8443/am-apm-web/resources/applications/+travel
https://appinsight IP:8443/am-apm-web/resources/applications/+sugar/metrics/cpu/overtime?
startTime=123498098&endTime=123534209
```

API Calls 3

The AppInsight API includes these requests.

- Get All Notifications on Application on page 13

  Retrieves all notifications for an application during a specified timeframe.

- Get All Notifications on page 14

  Retrieves all notifications for applications during a specified timeframe.

- Get the Health of All Applications on page 15

  Retrieves the average health metrics indicator for all applications for the specified time range.

- Get the Health of an Application on page 16

  Retrieves the average health metrics indicator for an application during a specified given timeframe.

- Get a Specific KPI State for an Application Over Time on page 16

  Retrieves the metrics for an indicator during a specified timeframe.

- Add an Application on page 17

  Adds a new application to AppInsight.

- Add Tier To Application on page 18

  Adds a tier to the topology of a specific application.

- Add a Component to a Tier on page 18

  Adds a component to a tier in the toplopgy of an application.

- Get Potential Components on page 19

  Retrieves a list of potential components that can be added to the topology of the application.

Get All Notifications on Application

Retrieves all notifications for an application during a specified timeframe.

You must have View permissions for the application to view the notifications.

```
https://appinsight_ip:8443/am-apm-web/resources/applications/notifications
```

HTTP Method

```
GET
```

Parameters

Parameter	Description	Example
`APP_NAME`	Name of application (required.)	travel
`startTime`	Sets the beginning of the time frame for fetched notifications in milliseconds. Default is ten minutes before endTime.	1234098437
`endTime`	Sets the end of the time frame for fetched notifications in milliseconds. Default is current time.	1235098344

Response

```
{ "meta" :{}, "result" : [ {"target": {"title":"travel", "type":"application"},
                                        "source":"appinsight",
                                        "type":"alert",
                                        "message":"At 7:40 AM, the Performance KPI deteriorated
                                               to Warning state",
                                        "eventTime", "1234098500"} ,
                                        {...}
                                  ]
   }
```

Get All Notifications

Retrieves all notifications for applications during a specified timeframe.

You can only see notifications for applications for which you have View permissions.

The notifications are at application level and do not include infrastructure or middleware notifications.

Notifications are sorted by their event time. You can retrieve the name of the application to which a notification refers from the notification message.

`/resources/applications/notifications`

HTTP Method

`GET`

Parameters

Parameter	Description	Example
`startTime`	Sets the beginning of the time frame for fetched notifications in milliseconds. Default is ten minutes before endTime.	1234098437
`endTime`	Sets the end of the time frame for fetched notifications in milliseconds. Default is current time.	1235098344

Response

```
{ "meta" :{}, "result" : [ {"target": {"title":"travel", "dataObjectType ":"application"},
                                        "source":"appinsight",
                                        "type":"alert",
                                        "message":"At 7:40 AM, the Performance KPI deteriorated
                                               to Warning state",
                                        "eventTime", "1234098500"} ,
                                        {...}
                                  ]
   }
```

Get the Health of All Applications

Retrieves the average health metrics indicator for all applications for the specified time range.

You can only see data for applications for which you have View permissions.

https://*appinsight_ip*:8443/am-apm-web/resources/applications/metrics/METRIC/average

HTTP Method

```
GET
```

Parameters

Parameter	Description	Example
METRIC	The metric that is to be fetched for all applications (required.)	health
startTime	Sets the beginning of the time frame for fetched notifications in milliseconds. Default is ten minutes before endTime.	1234098437
endTime	Sets the end of the time frame for fetched notifications in milliseconds. Default is current time.	1235098344

Response

```
{ "meta" :{}, "result" : [ {
                                     "metric":"HEALTH",
                                     "state":"WARNING",
                                     "element":
                                     {
                                       "title":"travel",
                                       "description":"description",
                                       "dataObjectType":"APPLICATION"
                                     }
                                    } ,
                                     {
                                     "metric":"HEALTH",
                                     "state":"EXCELLENT",
                                     "element":
                                     {
                                       "title":"petclinic",
                                               "description":"description",
                                               "dataObjectType":"APPLICATION"
                                             }
                                           },
                                           {...}
                                             ]
}
```

Get the Health of an Application

Retrieves the average health metrics indicator for an application during a specified given timeframe.

You must have View permissions for the application to view the data.

`https://appinsight_ip:8443/am-apm-web/resources/applications/+APP_NAME/metrics/INDICATOR/average`

HTTP Method

GET

Parameters

Parameter	Description	Example
APP_NAME	Name of application (required.)	travel
INDICATOR	The indicator metric that is to be fetched for the specified APP_NAME (required.)	health
startTime	Sets the beginning of the time frame for fetched notifications in milliseconds. Default is ten minutes before endTime.	1234098437
endTime	Sets the end of the time frame for fetched notifications in milliseconds. Default is current time.	1235098344

Response

```
{ "meta" :{}, "result" : [ {
                                              "metric":"HEALTH",
                                              "state":"WARNING",
                                              "element":
                                            {
                                              "title":"travel",
                                              "description":"description",
                                              "dataObjectType":"APPLICATION"
                                            }
                                          }
                                       ]
}
```

Get a Specific KPI State for an Application Over Time

Retrieves the metrics for an indicator during a specified timeframe.

You must have View permissions for the application to see the state.

`https://appinsight_ip:8443/am-apm-web/resources/applications/+APP_NAME/metrics/METRIC/overtime`

HTTP Method

GET

Parameters

Parameter	Description	Example
APP_NAME	Name of application (required.)	travel
METRIC	The metric that is to be fetched for the specified APP_NAME (required.)	health
startTime	Sets the beginning of the time frame for fetched notifications in milliseconds. Default is ten minutes before endTime.	1234098437
endTime	Sets the end of the time frame for fetched notifications in milliseconds. Default is current time.	1235098344

Response

```
{ "meta" :{}, "result" : [ {
                                    "Health KPI": "EXCELLENT",
                                    "occurred at": "1324566000000" ,
                                 },
                                 {
                                    "Health KPI": "CRITICAL",
                                    "occurred at": "1324566120000" ,
                                 },
                                 {...}
                                ]
}
```

Add an Application

Adds a new application to AppInsight.

You must have the Administrator role to use this request.

The application name must be unique.

`https://appinsight_ip:8443/am-apm-web/resources/applications/+APP_NAME`

HTTP Method

POST

Parameters

Parameter	Description	Example
APP_NAME	Name of application (required.)	travel
description	Description of the application	online travel agency application

Response

```
{"meta":{"message":""},"result":[{"title":"test",
                                           "description":"This was created by API",
                                           "dataObjectType":"APPLICATION"}
                                         ]
}
```

Add Tier To Application

Adds a tier to the topology of a specific application.

You must have Edit permissions to use this request.

`https://`*`appinsight_ip`*`:8443/am-apm-web/resources/applications/+APP_NAME/metrics/METRIC/overtime`

HTTP Method

```
POST
```

Parameters

Parameter	Description	Example
`APP_NAME`	Name of application (required.)	travel
`TIER_NAME`	Name of tier (required.)	load balancer
`position`	Position of the new tier. Possible values are `left` for left-most tier, and `right` for right-most tier. Default is `left`.	left

Response

```
{ "meta" :{}, "result" : [{"title":"load balancer","dataObjectType":"TIER"}
                                    ]
    }
```

Add a Component to a Tier

Adds a component to a tier in the toplopgy of an application.

You must have Edit permissions to use this request.

`https://`*`appinsight_ip`*`:8443/am-apm-`
`web/resources/applications/+APP_NAME/tiers/+TIER_NAME/components/+COMPONENT_NAME`

HTTP Method

```
POST
```

Parameters

Parameter	Description	Example
`APP_NAME`	Name of application (required.)	travel
`TIER_NAME`	Name of tier (required.)	load balancer

Parameter	Description	Example
COMPONENT_NAME	Name of component (required.)	travel
type	Type of component (required.) Possible values are APPSPEED_ATOM, BCI_COMPONENT, and BCI_DB_COMPONENT	BCI_COMPONENT
ip	IP of component (required.)	172.16.1.12
port	Port number of component (required.)	80
protocol	Type of protocol (required for network components.) Possible values are HTTP, MSSQL, MYSQL, ORACLE, and HSQLDB	HTTP
agentId	ID of the BCI agent (required if adding a BCI component.)	169331274

Response

```
{ "meta" :{}, "result" : [{}
                                   ]
    }
```

Get Potential Components

Retrieves a list of potential components that can be added to the topology of the application.

You must have the Administrator role to see the potential components.

https://*appinsight_ip*:8443/am-apm-web/resources/applications/+APP_NAME

HTTP Method

GET

Parameters

Parameter	Description	Example
APP_NAME	Name of application (required.)	travel

Response

```
{ "meta" :{}, "result" : [{"agentId":null,"name":"172.16.1.12:80",
                                   "mac":"00:50:56:90:00:62",
                                   "type":"APPSPEED_ATOM","port":80,
                                   "ip":"172.16.1.12",
                                   "componentProtocol":"HTTP"},

                                       {
                                   "agentId":169331274,"name":"travel",
                                   "mac":null,"type":"BCI_COMPONENT",
                                   "port":80,"ip":"10.23.202.74",
                                   "componentProtocol":null},...
                               ]
}
```

Core Plug-in Reference 4

Most metrics are gathered using plug-ins to AppInsight. You can develop your own plug-ins to gather additional metrics.

This appendix provides a list of core plug-ins that you can use to gather metrics that are not included as a default in AppInsight.

For information about how to integrate these plug-ins to AppInsight, see http://pubs.vmware.com/vfabric5/index.jsp?topic=/com.vmware.vfabric.tc-server.2.ł/devedition/about.html.

You can also access a plug-in development tutorial. See http://pubs.vmware.com/vfabric5/index.jsp?topic=/com.vmware.vfabric.tc-server.2.6/devedition/tutorial-plugin.html.

Core Plug-in Reference

Most metrics are contributed by plug-ins. Several plug-ins ship with vFabric AppInsight. Developers can create their own plug-ins. Only core plug-ins are documented here.

annotation

Use of the `@Insight*` annotations are optional. They make it easy for end users to define custom operation frames and end points without needing to create a plug-in. Because end user code modification is required to use the annotations, they are an option for users who cannot or do not wish to write aspects.

For more information on using the `@Insight*` annotations, see Using Annotations To Customize Insight Data For Your Application.

Operations of annotation plug-in

Collection Point	Summary
@InsightOperation	Any methods annotated with @InsightOperation will create an operation, including: basic source code location, method parameters and the return value.
@InsightEndPoint	Similar to @InsightOperation; also indicates the operation should be treated as an end point.
@InsightObscure	Used to annotate a value which should be obscured from Trace transmission or storage.
@InsightSensitive	Used to mark a method which processes sensitive data.

Endpoint Analysis of annotation plug-in

Analyzer	AnnotatedMethodEndPointAnalyzer
Core Operation	AnnotatedMethod Operation
Score	High by default, customizable in annotation.
Summary	Finds operations created from @InsightEndPoint, thus creating the end point.

apache-http-client3

Traces calls to external HTTP resources using the Apache HTTP client version 3.

Operations of Apache HTTP Client 3 plug-in

Collection Point	Summary
HttpClient.executeMethod(...)	Execution of HTTP calls

apache-http-client4

Traces calls to external HTTP resources using the Apache HTTP client version 4.

Operations of Apache HTTP Client 4 plug-in

Collection Point	Summary
HttpClient.execute(...)	Execution of HTTP calls

blazeds

Tracing of calls to BlazeDS. Supports the message broker APIs as well as command and message based communication.

Operations of BlazeDS plug-in

Collection Point	Summary
MessageBroker.route*(..)	Message broker operations
Service.serviceMessage(..)	Remote object and message based operations
Service.serviceCommand(..)	Command based operations

Endpoint Analysis of BalzeDS plug-in

Analyzer	BlazeDSEndPointAnalyzer
Score	Varies.
Summary	Creates an end point for HTTP requests handled by BlazeDS.

ehcache

Tracing of calls to Ehcache

Operations of ehcache plug-in

Collection Point	Summary
Get	Get value from Ehcache
Put	Put value in Ehcache
Remove	Remove a value from Ehcache
Replace	Replace a value in Ehcache

ejb3

Tracing of calls to EJB3 beans

Operations of EJB3 plug-in

Collection Point	Summary
@Stateful	Execution of stateful beans
@Stateless	Execution of stateful beans

files-tracker

Tracing of file open and close operations

Operations of files tracker plug-in

Collection Point	Summary
Open	A file was opened
Close	A file was closed

gemfire

Tracing of calls to GemFire

Operations of gemfire plug-in

Collection Point	Summary
Region	Writing and reading values using the GemFire Region APIs
Query	Querying GemFire using the Query APIs

grails

Grails-specific support for controller methods.

Operations of grails plug-in

Collection Point	Summary
Grails Controller Method	Method on a Grails Controller to which the request is mapped.

Endpoint Analysis of grails plug-in

Analyzer	GrailsControllerMethodEndPointAnalyzer
Core Operation	GrailsControllerMethodOperation
Score	Varies.
Summary	Creates an end point for HTTP requests handled by Grails.

hibernate

Support for tracing Hibernate persistence operations, such as get, save, update, and delete. A Hibernate session represents one transaction, which may include several database transactions. The hibernate plug-in collects the method name (flush, save, update, delete, etc.), entity count (the number of entity instances associated with the Hibernate session), and collection count (the number of collection instances associated with the session).

Operations of hibernate plug-in

Collection Point	Summary
org.hibernate.Session	CRUD operation on persistent entities.

javax-mail

Tracing of mail sending using java mail

Operations of javax-mail plug-in

Collection Point	Summary
javax.mail.Transport.sendMessage(..)	Sending an email message.

jax-rs

JAX-RS trace support.

Operations of jax-rs plug-in

Collection Point	Summary
@DELETE	Collect DELETE operations.
@GET	Collect GET operations.
@HEAD	Collect HEAD operations.
@POST	Collect POST operations.
@PUT	Collect PUT operations.

Endpoint Analysis of grails plug-in

Analyzer	JaxrsEndPointAnalyzer
Score	Varies.
Summary	Creates an end point for HTTP requests handled by JAX-RS.

jdbc

Low level support for tracing raw SQL queries.

Operations of jdbc plug-in

Collection Point	Summary
java.sql.Statement	Operation with the raw SQL statement.

Collection Point	Summary
java.sql.PreparedStatement	Operation with the raw SQL statement and parameters.

jms

jms trace support.

Operations of jms plug-in

Collection Point	Summary
MessageConsumer	Tracing for the javax.jms.MessageConsumer APIs
MessageListener	Tracing for the javax.jms.MessageListener APIs
MessageProducer	Tracing for the javax.jms.MessageProducer APIs

Endpoint Analysis of jms plug-in

Analyzer	JMSConsumerEndPointAnalyzer and JMSMessageListenerEndPointAnalyzer
Score	Varies.
Summary	Creates an end point for messages that are received on a queue or topic.

jws

jws trace support.

Operations of jws plug-in

Collection Point	Summary
@WebService	Tracing for the javax.jws.WebService APIs

Endpoint Analysis of jms plug-in

Analyzer	JwsEndPointAnalyzer
Score	Varies.
Summary	Creates an end point for HTTP requests handled by JAX-RS.

logging

Logging frameworks support. Currently supports Log4J, SLF4J and Commons logging. This plugin will create error traces when an error or fatal message is logged.

Operations of logging plug-in

Collection Point	**Summary**
Log.error and Log.fatal	Logging using commons logging
Category.error and Category.fatal	Logging using Log4J
Logger.error	Logging using SLF4J

method-endpoint

Creates an end point for a root-level operation within a trace. When a web request invokes a method asynchronously (background threads, scheduled threads, etc.), this plug-in creates an end point for it.

Endpoint Analysis of method-endpoint plug-in

Analyzer	TopLevelMethodEndPointAnalyzer
Core Operation	MethodOperation
Score	Low.
Summary	Creates an end point for a root-level operation within a Trace.

mongodb

Support for tracing MongoDB queries.

Operations of mongodb plug-in

Collection Point	**Summary**
com.mongodb.DBCollection	Operations with the DBCollection APIs. insert, update, remove, save, find, count, group, distinct, map reduce and index operations are supported
com.mongodb.DBCursor	Operations with the DBCursor APIs. next, skip, limit,

Collection Point	Summary
	toArray and sort operations are supported

quartz-scheduler

Support for tracing Quartz scheduler queries.

Operations of quartz-scheduler plug-in

Collection Point	Summary
Job.execute	Quartz job execution

Endpoint Analysis of quartz-scheduler plug-in

Analyzer	QuartzSchedulerEndPointAnalyzer
Score	Low.
Summary	Creates an end point for Quartz triggered jobs.

rabbitmq-client

Support for tracing RabbitMQ queries.

Operations of rabbitmq-client plug-in

Collection Point	Summary
com.rabbitmq.client.Consumer and com.rabbitmq.client.Channel	Consumer and Channel based message receivers
com.rabbitmq.client.Channel	Channel based message publishing

Endpoint Analysis of rabbitmq-client plug-in

Analyzer	AbstractRabbitMQResourceAnalyzer
Score	Low.
Summary	Creates an end point for messages that are received on an exchange or routing key.

redis

Support for tracing Redis queries.

Operations of redis plug-in

Collection Point	Summary
Jedis	Collects operations executed using the Jedis APIs
AbstractRedisCollection	Collects operations exucted using the Spring Data abstraction for Redis

servlet

Creates end points for servlets and application lifecycle events such as start and stop. Support for servlet Listeners and Filters operation collection. For tc Runtime, the collection is done by the `com.springsource.insight.collection.tcserver` packages.

Operations of servlet plug-in

Collection Point	Summary
javax.servlet.Filter	Operation with the filter name and init params.
javax.servlet.ServletContextListener	Operations for context initialization and destruction events with context params.

Endpoint Analysis of servlet plug-in

Analyzer	ServletEndPointAnalyzer
Core Operation	HttpOperation
Score	Low.
Summary	Creates an end point for HTTP requests, grouped by the matching servlet, that are otherwise not accounted for by an end point.

socket

Support for tracking opening and closing of sockets

Operations of socket plug-in

Collection Point	Summary
HttpURLConnection	Creating a new connection
ServerSocketChannel, ServerSocket	Accepting a connection
SocketChannel	Creating a new connection or openning a connection
Socket	Creating a connection

spring-core

Basic support for calls into Spring managed beans containing user business logic. Currently only calls into objects annotated with @Service or @Repository stereotype annotations are supported. Even if component scanning is not enabled, using these annotations allows the agent to pick up the operations.

Operations of spring-core plug-in

Collection Point	Summary
@Repository	Creates an operation for calls into classes annotated with @Repository
@Service	Creates an operation for calls into classes annotated with @Service

spring-integration

Support for Spring Integration.

Operations of spring-integration plug-in

Collection Point	Summary
MessageChannel	Sending messages using the MessageChannel APIs
MessageHandler	Handling of messages using the MessageChannel APIs
Transformer	Tansforming of messages using the Transformer APIs

Endpoint Analysis of spring-integration plug-in

Analyzer	IntegrationEndPointAnalyzer

Score	Low.
Summary	Creates an end point for operations that were triggered in Spring Integration.

spring-security

Tracing support for Spring Security.

Operations of spring-security plug-in

Collection Point	Summary
AuthenticationManager	Calls to the authenticate operation
AuthenticationProvider	Calls to the authenticate operation
UserDetailsService	Calls to the loadUserByUsername, createUser, updateUser, deleteUser, changePassword and userExists operations

spring-tx

Detects the creation, commit, and rollback of a transaction using Spring's PlatformTransactionManager. Transaction boundaries defined with @Transactional or with standard Spring config are detected.

Operations of spring-tx plug-in

Collection Point	Summary
PlatformTransactionManager	Defines the boundaries of a transaction. Attributes such as the propagation, isolation, timeout and readonly states are collected.

spring-web

Full support for events in the Spring MVC request life cycle. Both annotated @Controller and the legacy Controller interface class hierarchy flavors of MVC are supported.

Operations of spring-web plug-in

Collection Point	**Summary**
@InitBinder	Data binder configuration including target name and type, allowed, required, and disallowed fields.
@ModelAttribute	Methods returning objects that are added directly to the model, including the attribute name and value.
@RequestMapping	Spring MVC request handlers.
Validator	Validation logic.
View	View rendering including view name and content type.
ViewResolver	View resolution with the view name requested and the matching View object returned.
Controller	Legacy Spring MVC Controller interface.
DispatcherServlet	Handling of requests by Spring's DispatcherServlet.

Endpoint Analysis of spring-web plug-in

Analyzer	ControllerEndPointAnalyzer
Core Operation	ControllerMethodOperation
Score	Varies.
Summary	Creates an end point for HTTP requests handled by Spring MVC.

tomcat

Provides deep hook into Tomcat and vFabric tc Server internals that are not exposed by the public Servlet API

Operations of tomcat plug-in

Collection Point	**Summary**
JSP Compiler	Operation showing time spent in JSP compilation (typically the first request to a JSP) and the compiler implementation.

Server-Based Instrumentation

The following instrumentation is provided by default to the tc Runtime; it is not provided as a plug-in.

Operations of Server-Based Instrumentation

Collection Point	**Summary**
AbstractHttpRequestOperationSupport	Creates an HTTP operation for every HTTP request to a web application. Full request and response headers are collected as provided by Tomcat and the Servlet API. Request parameters are provided if available (user parsing of the request body prevents parameters from being available). Request headers are collected before the application is involved, and response headers are collected after the application has returned.
ApplicationLifecycleCollectionSupport	Creates traces for application life cycle events such as start and stop. Not all web applications have a start event when the server is first starting, if the agent has not fully initialized.

Index