APPDYNAMICS

JMeter integration with AppDynamics

Purpose

The business use case is to accelerate time to market by understanding quickly, with AppDynamics, the bottleneck of performance and the root cause of issues during load testing with JMeter.

The purpose of JMeter integration with AppDynamics is to:

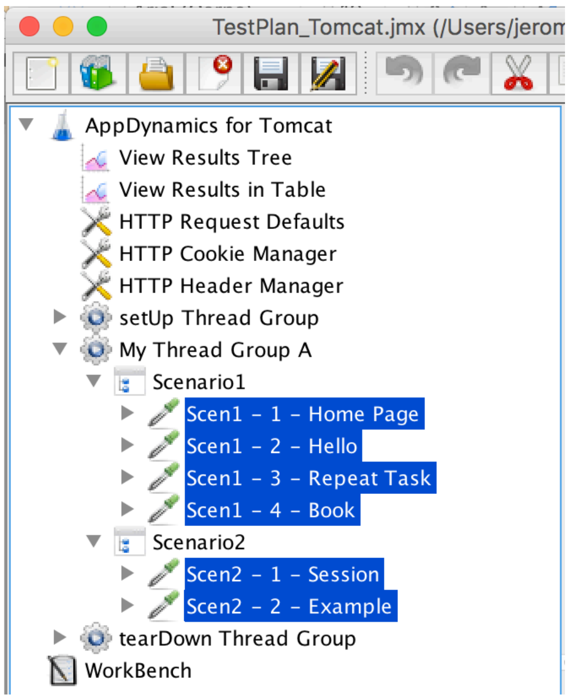
1. Offer a correlated view between JMeter measurements (HTT Request) and AppDynamics measurements (Business Transaction).
    - JMeter simulates end-user activity by launching thread group and by executing HTTP request. AppDynamics retrieves dynamically the name of JMeter HTTP request and stores the metrics in a AppDynamics Business Transaction (BT)
2. Create a custom time range in AppDynamics in relation with the duration of JMeter execution; and compare easily two JMeter executions or analyze specifically the result of one JMeter execution
3. Retrieve the JMeter Thread name within the AppDynamics snapshot (i.e. capture request details and give visibility to call graph which reflects the code-level view); and diagnose easily why JMeter Thread fails

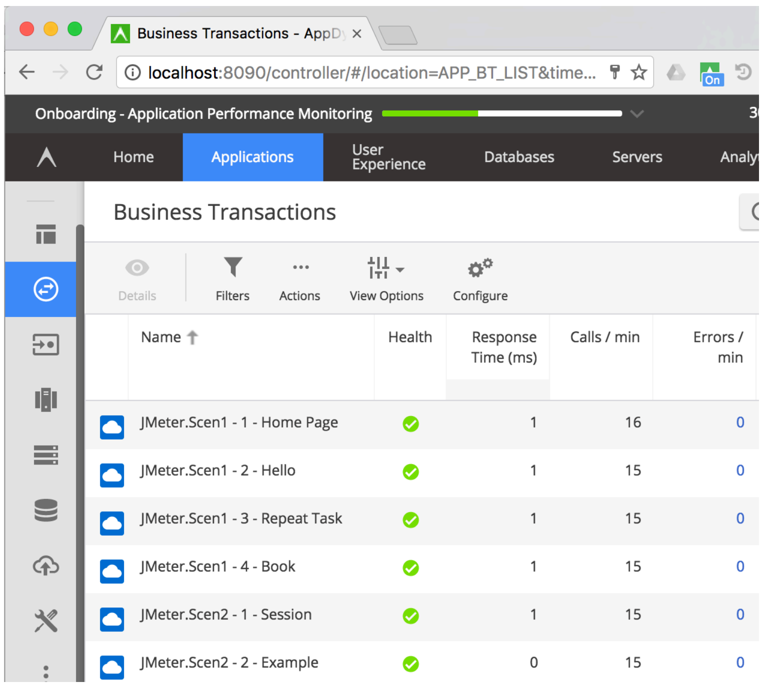
Note: this integration has been done with JMeter 3.1 and AppDynamics 4.2. It should work with other versions.

1) Configure the correlated view between JMeter HTTP Request and AppDynamics BT

JMeter - Test Plan

AppDynamics - Business Transactions

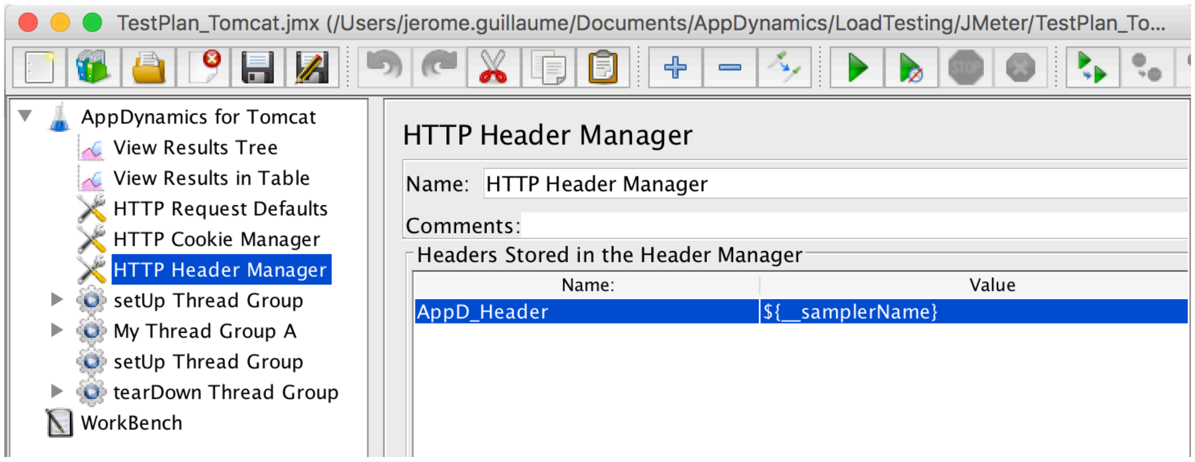



Business Transactions - AppD
localhost:8090/controller/#/location=APP_BT_LIST&time...
Onboarding - Application Performance Monitoring
Home
Applications
User Experience
Databases
Servers
Analy
Business Transactions
Details
Filters
Actions
View Options
Configure

Name	Health	Response Time (ms)	Calls / min	Errors / min
JMeter.Scen1 - 1 - Home Page	✓	1	16	0
JMeter.Scen1 - 2 - Hello	✓	1	15	0
JMeter.Scen1 - 3 - Repeat Task	✓	1	15	0
JMeter.Scen1 - 4 - Book	✓	1	15	0
JMeter.Scen2 - 1 - Session	✓	1	15	0
JMeter.Scen2 - 2 - Example	✓	0	15	0

How to configure JMeter (with AppDynamics)

- Start JMeter and open the Test Plan
- Append a `HTTP Header Manager` (right click on Thread Group, select Add/Config Element/HTTP Header Manager) on Thread Group
- Add a header
  - Set name with `AppD_Header` and value with `${__samplerName}` and click on Save
- See below the expected result


TestPlan_Tomcat.jmx (/Users/jerome.guillaume/Documents/AppDynamics/LoadTesting/JMeter/TestPlan_To...

HTTP Header Manager

Name: HTTP Header Manager

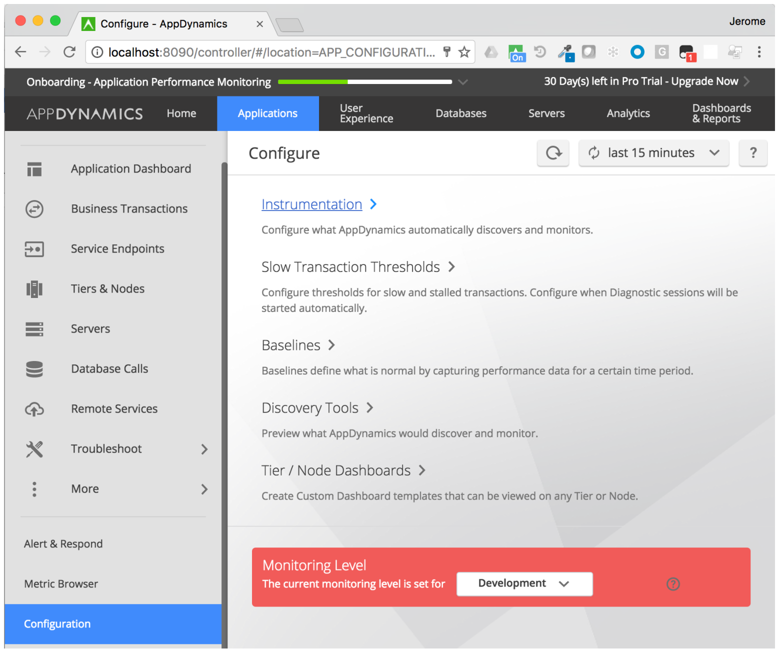
Comments:

Headers Stored in the Header Manager

Name:	Value
AppD_Header	${__samplerName}

How to configure AppDynamics (with JMeter)

- Connect to AppDynamics Controller
- Select the application
- Click on Configuration
- Click on Instrumentation



Instrumentation >
Configure what AppDynamics automatically discovers and monitors.

Slow Transaction Thresholds >
Configure thresholds for slow and stalled transactions. Configure when Diagnostic sessions will be started automatically.

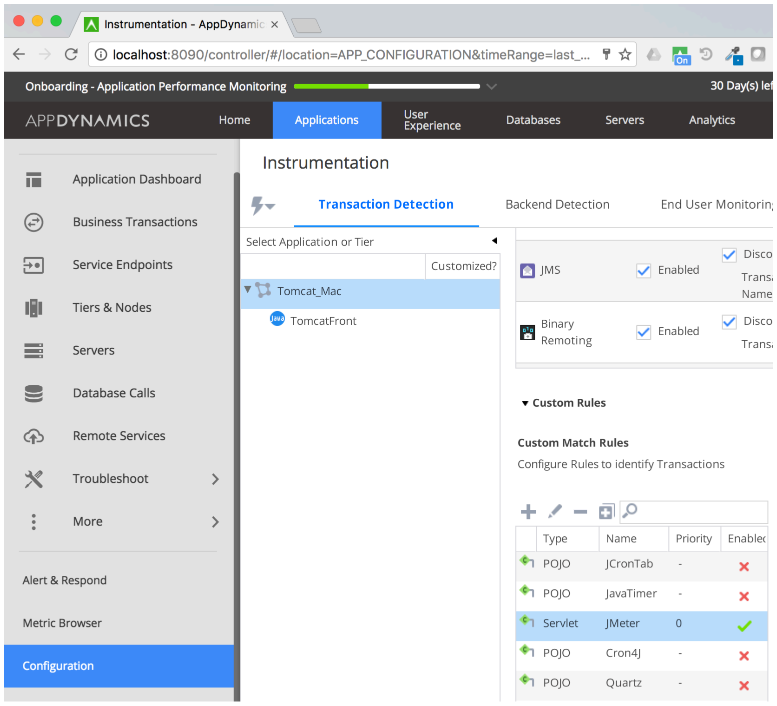
Baselines >
Baselines define what is normal by capturing performance data for a certain time period.

Discovery Tools >
Preview what AppDynamics would discover and monitor.

Tier / Node Dashboards >
Create Custom Dashboard templates that can be viewed on any Tier or Node.

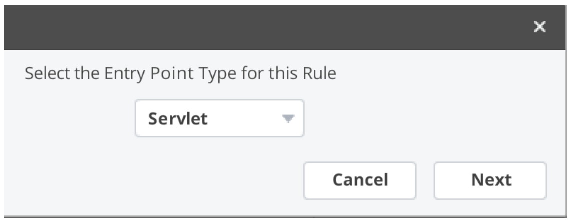
Monitoring Level
The current monitoring level is set for Development

- In "Custom Match Rules" section, click on the + button



Type	Name	Priority	Enablec
POJO	JCronTab	-	✗
POJO	JavaTimer	-	✗
Servlet	JMeter	0	✓
POJO	Cron4J	-	✗
POJO	Quartz	-	✗

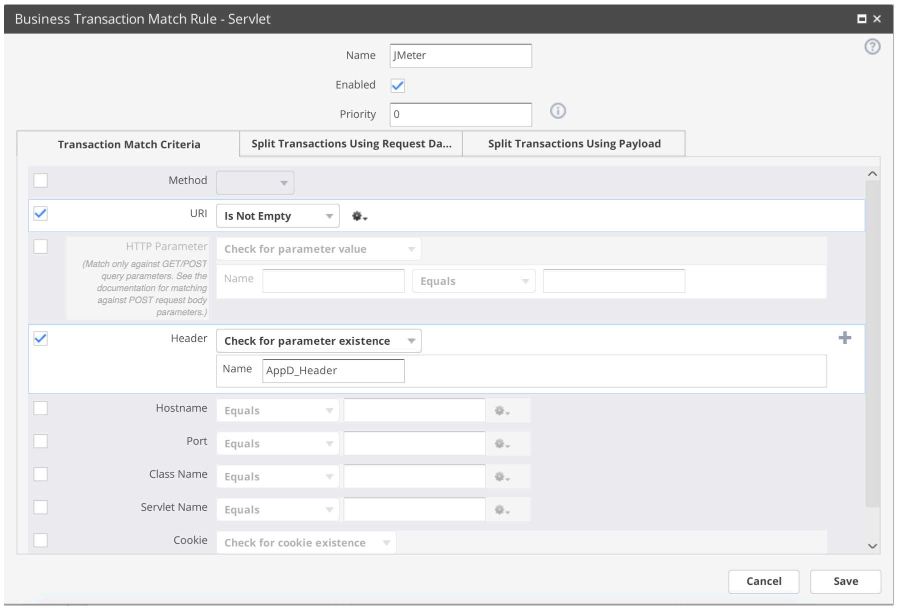
- Select servlet and click on Next



Select the Entry Point Type for this Rule

Servlet

- Set following values:
  - Name = `JMeter`
  - URI = `is Not Empty`
  - Header = Check for parameter existence, name = `AppD_Header`

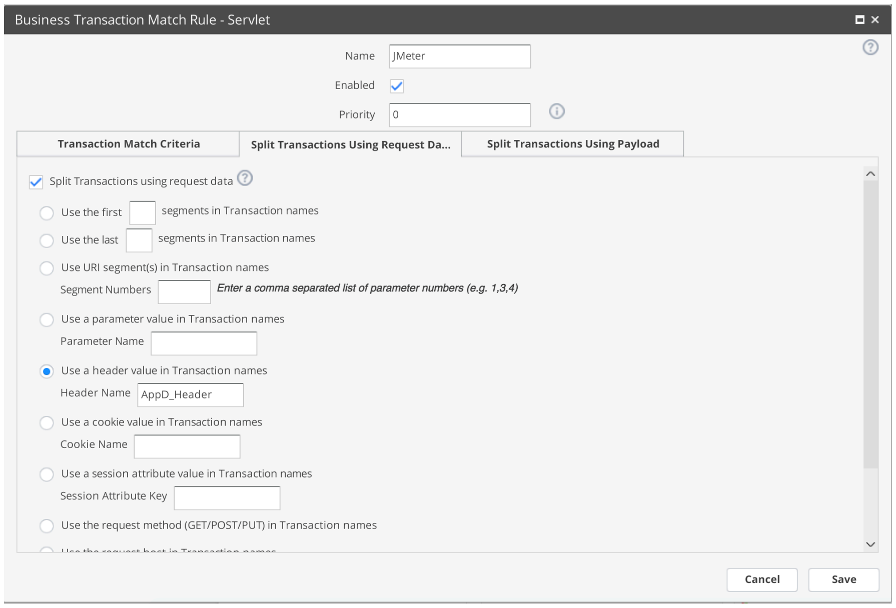

Business Transaction Match Rule - Servlet

Name: JMeter
Enabled: ✓
Priority: 0

Transaction Match Criteria
Split Transactions Using Request Da...
Split Transactions Using Payload

☐	Method	
✓	URI	Is Not Empty
☐	HTTP Parameter (Match only against GET/POST query parameters. See the documentation for matching against POST request body parameters.)	Check for parameter value; Name; Equals
✓	Header	Check for parameter existence; Name: AppD_Header
☐	Hostname	Equals
☐	Port	Equals
☐	Class Name	Equals
☐	Servlet Name	Equals
☐	Cookie	Check for cookie existence

- Click on tab "Split Transactions Using Request Data" and set the following values:
  - Check "Split Transactions Using request Data"
  - Select "Use a header value" in Transaction names and set value = `AppD_Header`


Business Transaction Match Rule - Servlet

Name: JMeter
Enabled: ✓
Priority: 0

Transaction Match Criteria
Split Transactions Using Request Da...
Split Transactions Using Payload

✓ Split Transactions using request data

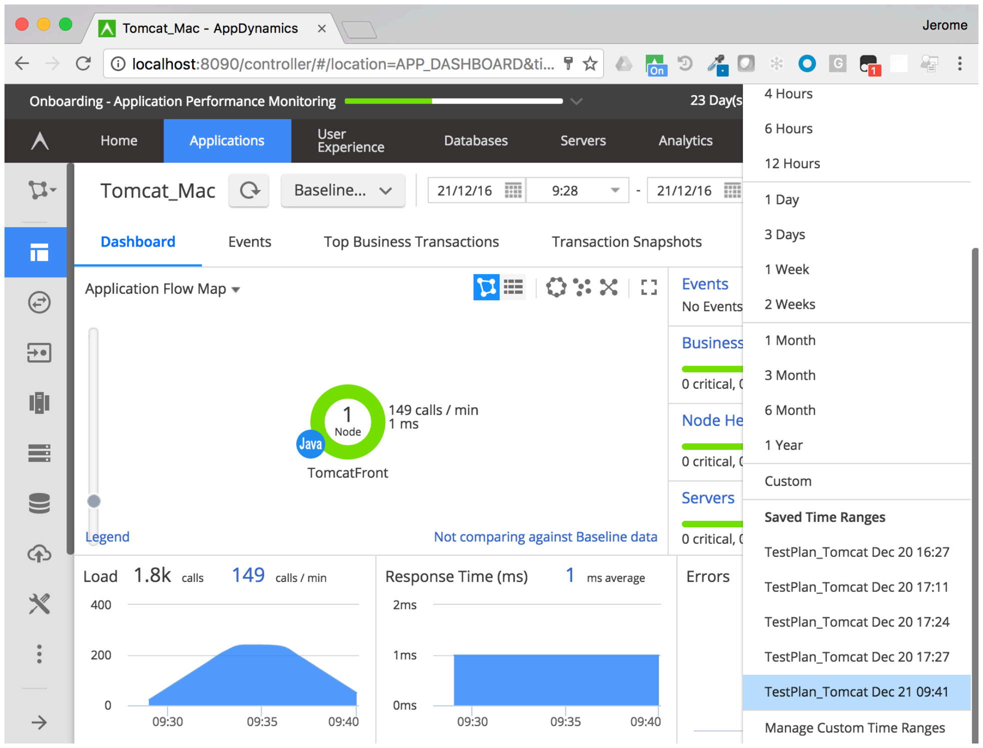
- ○ Use the first ___ segments in Transaction names
- ○ Use the last ___ segments in Transaction names
- ○ Use URI segment(s) in Transaction names
  - Segment Numbers ___ *Enter a comma separated list of parameter numbers (e.g. 1,3,4)*
- ○ Use a parameter value in Transaction names
  - Parameter Name ___
- ● Use a header value in Transaction names
  - Header Name: AppD_Header
- ○ Use a cookie value in Transaction names
  - Cookie Name ___
- ○ Use a session attribute value in Transaction names
  - Session Attribute Key ___
- ○ Use the request method (GET/POST/PUT) in Transaction names
- ○ Use the request host in Transaction names

- Click on Save

2) Create a custom time range in AppDynamics in relation with duration of JMeter execution

See below examples of custom time range created by JMeter execution. The name is based on JMeter Test plan name and date & hour of end of JMeter execution.

- TestPlan_Tomcat Dec 20 16:27
- TestPlan_Tomcat Dec 20 17:11
- TestPlan_Tomcat Dec 20 17:24
- TestPlan_Tomcat Dec 20 17:27
- TestPlan_Tomcat Dec 21 09:41
- …



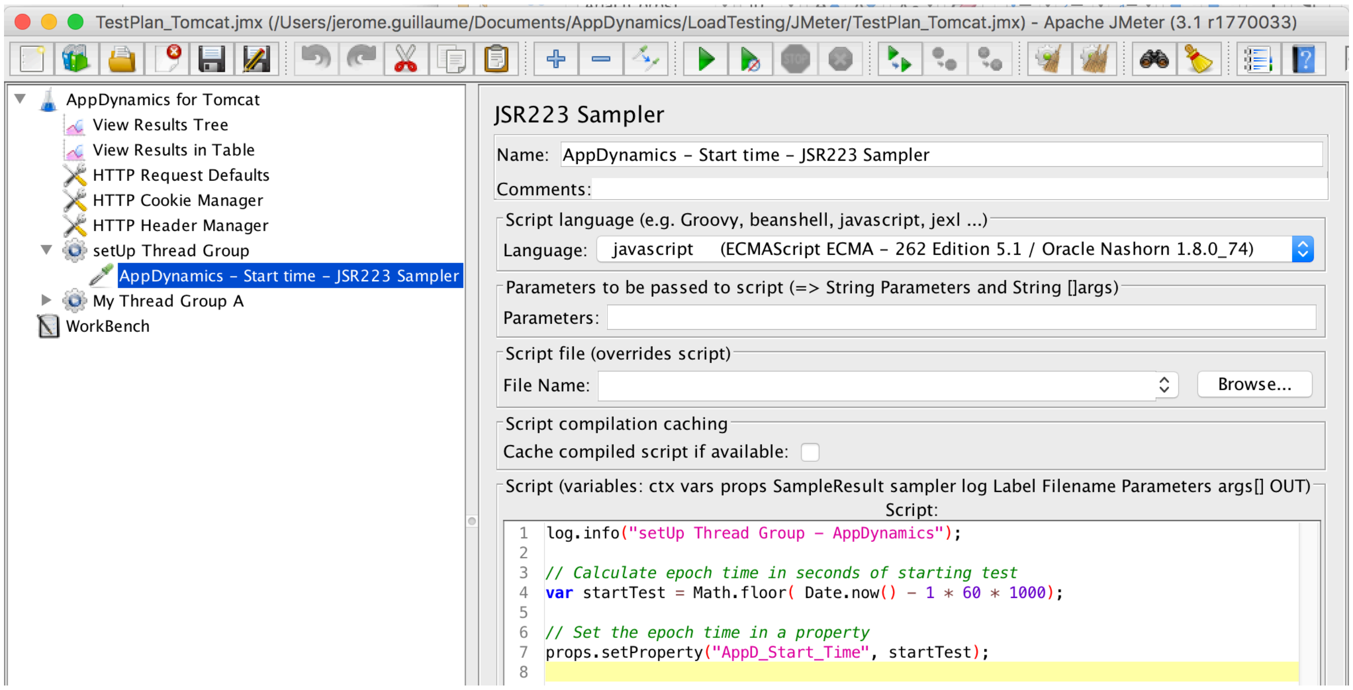
How to configure JMeter (with AppDynamics)

- Prerequisite: **have a test plan name with no space**, the name will be retrieved dynamically and it will be used to setup the name of custom range. Prefer a name like `TestPlan_Tomcat.jmx` instead of `TestPlan Tomcat.jmx`
- Start JMeter and open the Test Plan
- Add a `setup Thread Group`
- Add a `JSR223 Sampler` below `setup Thread Group` and call it `AppDynamics – Start time – JSR223 Sampler`
- Configure `AppDynamics – Start time – JSR223 Sampler` with a JavaScript language and append the following code

```
log.info("setUp Thread Group - AppDynamics");
```

```
// Calculate epoch time in seconds of starting test
var startTest = Math.floor( Date.now() - 1 * 60 * 1000);

// Set the epoch time in a property
props.setProperty("AppD_Start_Time", startTest);
```

The result is as follows:



TestPlan_Tomcat.jmx (/Users/jerome.guillaume/Documents/AppDynamics/LoadTesting/JMeter/TestPlan_Tomcat.jmx) - Apache JMeter (3.1 r1770033)

- AppDynamics for Tomcat
  - View Results Tree
  - View Results in Table
  - HTTP Request Defaults
  - HTTP Cookie Manager
  - HTTP Header Manager
  - setUp Thread Group
    - AppDynamics – Start time – JSR223 Sampler
  - My Thread Group A
- WorkBench

JSR223 Sampler

Name: AppDynamics – Start time – JSR223 Sampler

Comments:

Script language (e.g. Groovy, beanshell, javascript, jexl ...)

Language: javascript (ECMAScript ECMA – 262 Edition 5.1 / Oracle Nashorn 1.8.0_74)

Parameters to be passed to script (=> String Parameters and String []args)

Parameters:

Script file (overrides script)

File Name:

Script compilation caching

Cache compiled script if available: ☐

Script (variables: ctx vars props SampleResult sampler log Label Filename Parameters args[] OUT)

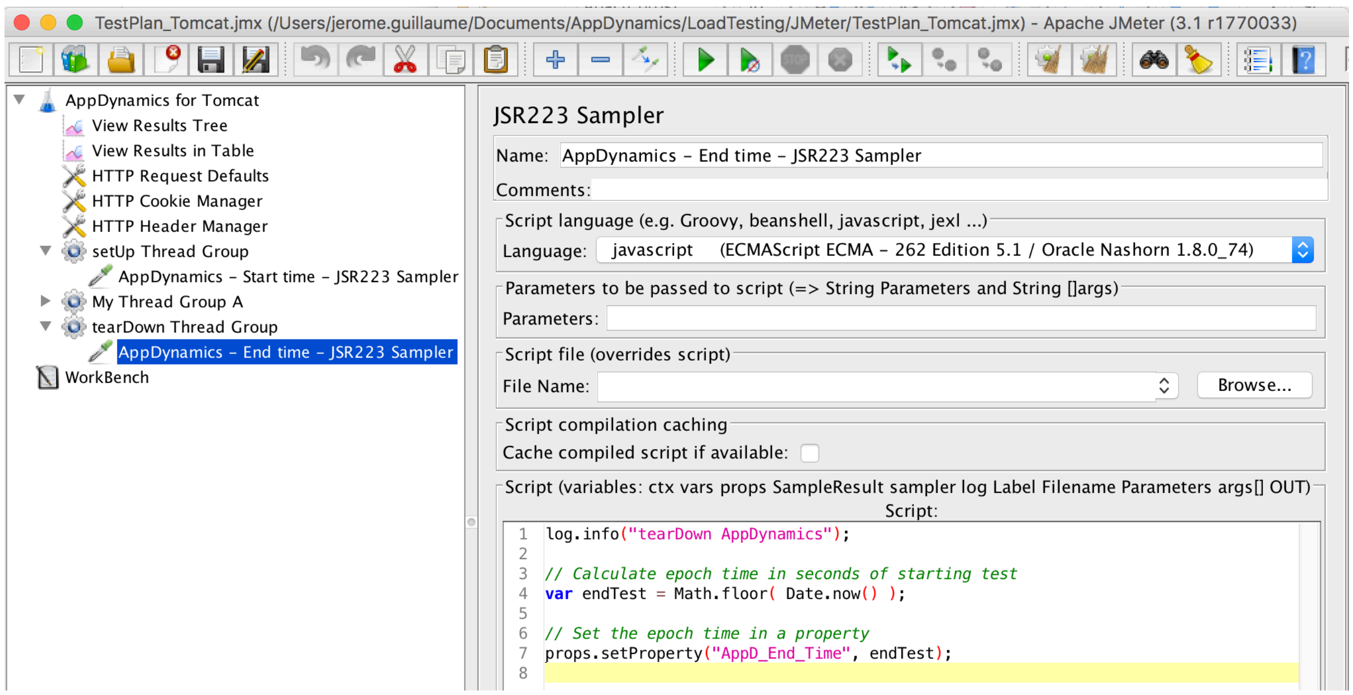
Script:

```
log.info("setUp Thread Group – AppDynamics");

// Calculate epoch time in seconds of starting test
var startTest = Math.floor( Date.now() – 1 * 60 * 1000);

// Set the epoch time in a property
props.setProperty("AppD_Start_Time", startTest);

```

- Add a `tearDown Thread Group`
- Add a `JSR223 Sampler` below `tearDown Thread Group` and call it `AppDynamics – End time – JSR223 Sampler`
- Configure `AppDynamics – End time – JSR223 Sampler` with a JavaScript language and append the following code

```
log.info("tearDown AppDynamics");

// Calculate epoch time in seconds of starting test
var endTest = Math.floor( Date.now() );

// Set the epoch time in a property
props.setProperty("AppD_End_Time", endTest);
```

The result is as follows:


TestPlan_Tomcat.jmx (/Users/jerome.guillaume/Documents/AppDynamics/LoadTesting/JMeter/TestPlan_Tomcat.jmx) - Apache JMeter (3.1 r1770033)
AppDynamics for Tomcat
View Results Tree
View Results in Table
HTTP Request Defaults
HTTP Cookie Manager
HTTP Header Manager
setUp Thread Group
AppDynamics – Start time – JSR223 Sampler
My Thread Group A
tearDown Thread Group
AppDynamics – End time – JSR223 Sampler
WorkBench
Browse...

JSR223 Sampler

Name: AppDynamics – End time – JSR223 Sampler

Comments:

Script language (e.g. Groovy, beanshell, javascript, jexl ...)

Language: javascript (ECMAScript ECMA – 262 Edition 5.1 / Oracle Nashorn 1.8.0_74)

Parameters to be passed to script (=> String Parameters and String []args)

Parameters:

Script file (overrides script)

File Name:

Script compilation caching

Cache compiled script if available: ☐

Script (variables: ctx vars props SampleResult sampler log Label Filename Parameters args[] OUT)

Script:

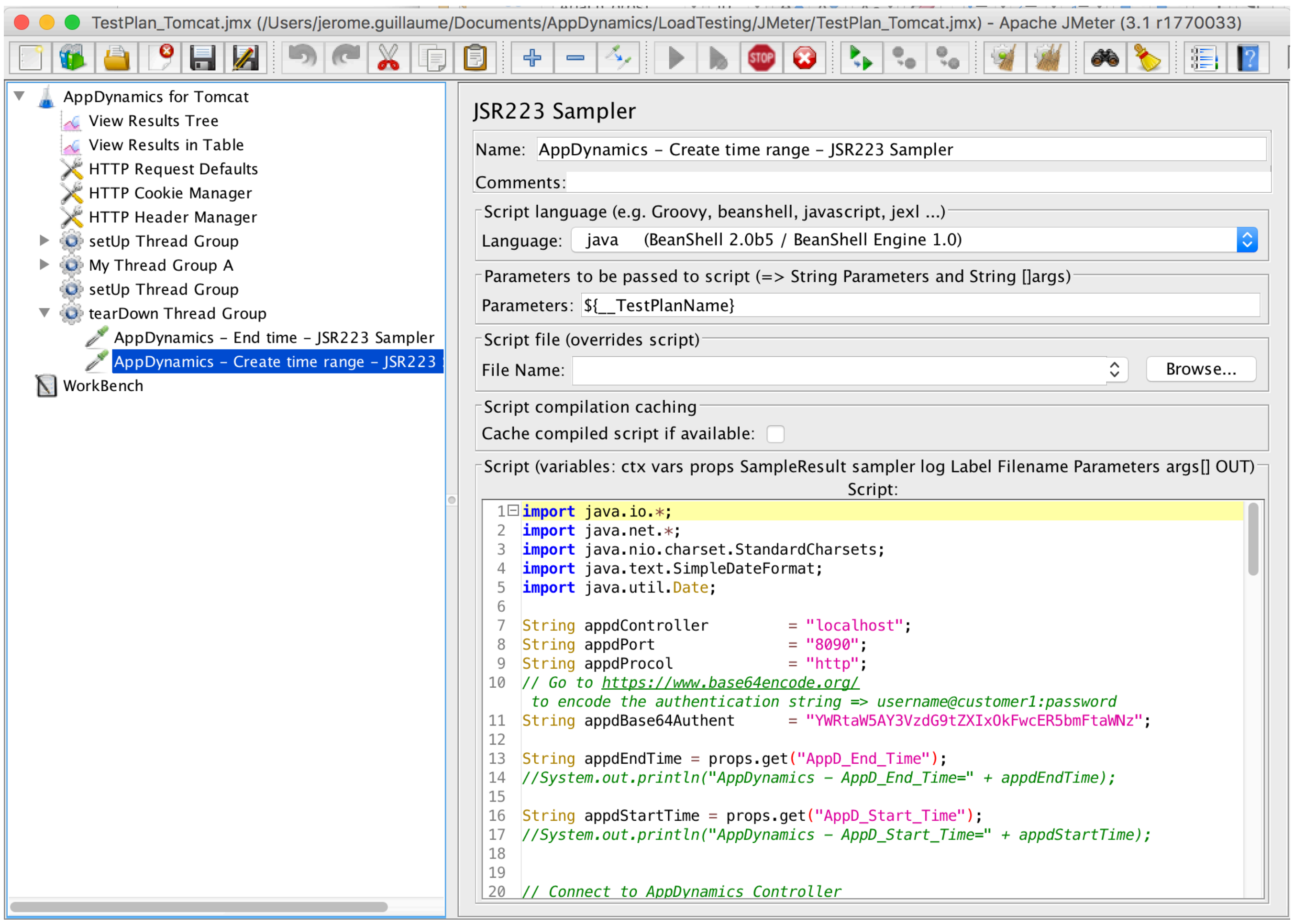
```
log.info("tearDown AppDynamics");

// Calculate epoch time in seconds of starting test
var endTest = Math.floor( Date.now() );

// Set the epoch time in a property
props.setProperty("AppD_End_Time", endTest);

```

- Add a `JSR223 Sampler` below `tearDown Thread Group` and call it `AppDynamics – Create time range – JSR223 Sampler`
- Configure `AppDynamics – End time – JSR223 Sampler` with a Java language (not JavaScript); set **Parameters value to `${  TestPlanName}`**
- Append the following code:
  - The values in regards of AppDynamics Controller should be configured correctly: `host, port, protocol` and `base64 authentication string`.
  - Go to https://www.base64encode.org/ to encode the authentication string in base64 format ; the format of authentication string is `username@customer1:password`

```
import java.io.*;
import java.net.*;
import java.nio.charset.StandardCharsets;
import java.text.SimpleDateFormat;
import java.util.Date;

String appdController           = "localhost";
String appdPort                 = "8090";
String appdProcol               = "http";
// Go to https://www.base64encode.org/ to encode the authentication string =>
username@customer1:password
String appdBase64Authent        = "YWRtaW5AY3VzdG9tZXIxOkFwcER5bmFtaWNz";

String appdEndTime = props.get("AppD_End_Time");
//System.out.println("AppDynamics - AppD_End_Time=" + appdEndTime);

String appdStartTime = props.get("AppD_Start_Time");
//System.out.println("AppDynamics - AppD_Start_Time=" + appdStartTime);


// Connect to AppDynamics Controller
StringBuilder result = new StringBuilder();
URL url = new URL(appdProcol + "://" + appdController + ":" + appdPort + "/controller/auth?action=login");
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestProperty("Authorization", "Basic " + appdBase64Authent);
conn.setRequestMethod("GET");
```

```
BufferedReader brLogin = new BufferedReader(new InputStreamReader(conn.getInputStream()));

// If we are correctly connected to AppDynamics Controller
if (conn.getResponseCode() == 200)
{
          String headerName = null;
          String appdCookie = "";

          for (int i=1; (headerName = conn.getHeaderFieldKey(i))!=null; i++) {
                    if (headerName.equals("Set-Cookie"))
                    {
                              appdCookie += conn.getHeaderField(i) + " ;";
                    }
          }

          //System.out.println("appdCookie=" + appdCookie);


          URL urlCustomRange = new URL(appdProcol + "://" + appdController + ":" + appdPort + "/controller/restui/user/createCustomRange");

          HttpURLConnection connCustomRange = (HttpURLConnection) urlCustomRange.openConnection();
          connCustomRange.setRequestMethod("POST");
          connCustomRange.setDoOutput(true);
          connCustomRange.setRequestProperty("Authorization", "Basic " + appdBase64Authent);
          connCustomRange.setRequestProperty("Content-Type", "application/json");
          connCustomRange.setRequestProperty("charset", "utf-8");
          connCustomRange.setRequestProperty("Accept-Encoding", "gzip, deflate");
          connCustomRange.setRequestProperty("Accept", "application/json, text/plain");
          connCustomRange.setRequestProperty("Cookie", appdCookie);

          // If the Test Plan name has been setup correctly in Parameters of this script
          if (args.length > 0)
          {
                    String appdTestPlan = args[0];
                    // Remove the extenson .jmx from file name
                    if (appdTestPlan.indexOf(".jmx") != -1)
                    {
                              appdTestPlan = appdTestPlan.substring(0, appdTestPlan.indexOf(".jmx"));
                    }
                    Date date = new Date();
                    SimpleDateFormat sdf = new SimpleDateFormat("MMM dd HH:mm");
                    appdTestPlan += " " + sdf.format(date);

                    String postJsonData = "{\"name\":\"" + appdTestPlan + "\",\"description\":\"JMeter
execution\",\"shared\":true,\"timeRange\":{\"type\":\"BETWEEN_TIMES\",\"durationInMinutes\":0,\"startTime\":" + appdStartTime +
",\"endTime\":" + appdEndTime +"}}";

                    DataOutputStream wr = new DataOutputStream(connCustomRange.getOutputStream());
                    wr.writeBytes(postJsonData);
                    wr.flush();
                    wr.close();
                    if (connCustomRange.getResponseCode() != 200)
                    {
                              System.out.println("AppDynamics - Unable to create time range" + urlCustomRange + " HTTP Code=" +
connCustomRange.getResponseCode());
                    }
          }
          // Else the Test Plan name has been not setup correctly in Parameters of this script
          else
          {
                    System.out.println("AppDynamics - the test plan name has not been setup in parameters");
          }
}
else
{
          System.out.println("AppDynamics - Unable to connect to Controller" + url + " HTTP Code=" + conn.getResponseCode());
}

brLogin.close();
```

The result is as follows:



TestPlan_Tomcat.jmx (/Users/jerome.guillaume/Documents/AppDynamics/LoadTesting/JMeter/TestPlan_Tomcat.jmx) - Apache JMeter (3.1 r1770033)

- AppDynamics for Tomcat
  - View Results Tree
  - View Results in Table
  - HTTP Request Defaults
  - HTTP Cookie Manager
  - HTTP Header Manager
  - setUp Thread Group
  - My Thread Group A
  - setUp Thread Group
  - tearDown Thread Group
    - AppDynamics – End time – JSR223 Sampler
    - AppDynamics – Create time range – JSR223
- WorkBench

JSR223 Sampler

Name: AppDynamics – Create time range – JSR223 Sampler

Comments:

Script language (e.g. Groovy, beanshell, javascript, jexl ...)

Language: java (BeanShell 2.0b5 / BeanShell Engine 1.0)

Parameters to be passed to script (=> String Parameters and String []args)

Parameters: ${__TestPlanName}

Script file (overrides script)

File Name:

Script compilation caching

Cache compiled script if available:

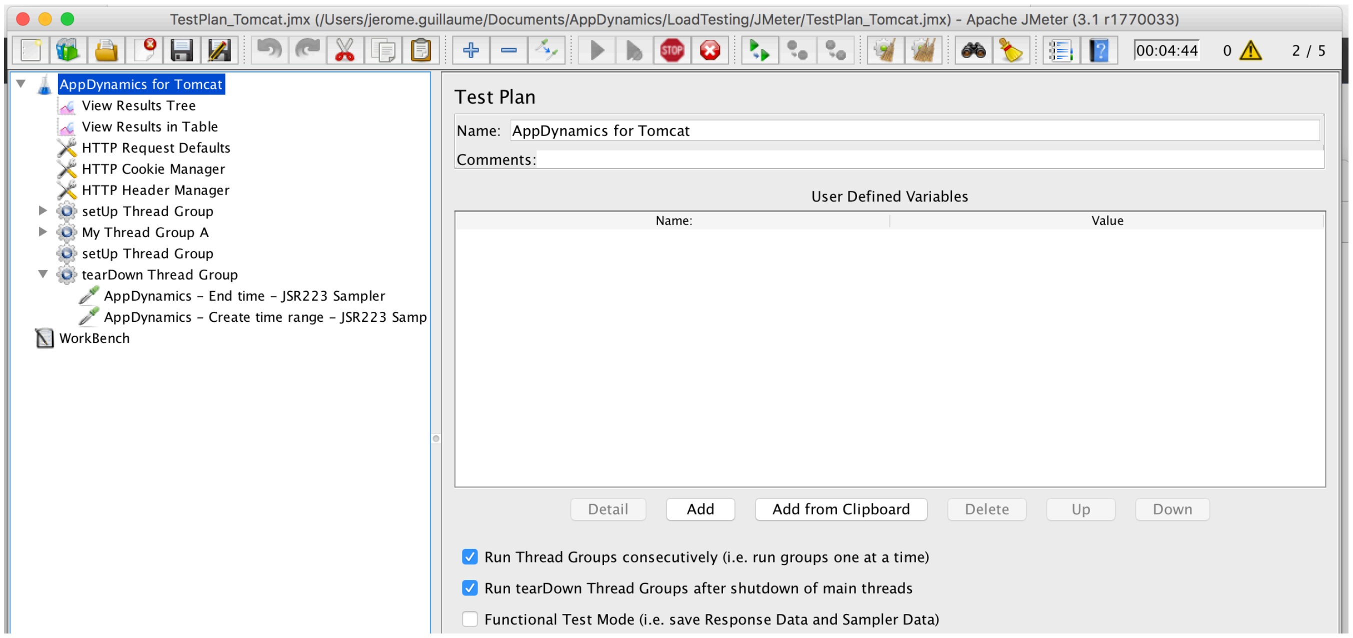
Script (variables: ctx vars props SampleResult sampler log Label Filename Parameters args[] OUT)

Script:

```
import java.io.*;
import java.net.*;
import java.nio.charset.StandardCharsets;
import java.text.SimpleDateFormat;
import java.util.Date;

String appdController         = "localhost";
String appdPort               = "8090";
String appdProcol             = "http";
// Go to https://www.base64encode.org/
 to encode the authentication string => username@customer1:password
String appdBase64Authent      = "YWRtaW5AY3VzdG9tZXIxOkFwcER5bmFtaWNz";

String appdEndTime = props.get("AppD_End_Time");
//System.out.println("AppDynamics - AppD_End_Time=" + appdEndTime);

String appdStartTime = props.get("AppD_Start_Time");
//System.out.println("AppDynamics - AppD_Start_Time=" + appdStartTime);


// Connect to AppDynamics Controller
```

- Configure the Test plan by checking `run tearDown Thread Groups after shutdown of main threads`. The tearDown threads won't be run if the test is forcibly stopped and it will avoid useless creation of custom time range in AppDynamics.
  The result is as follows:

APPDYNAMICS


TestPlan_Tomcat.jmx (/Users/jerome.guillaume/Documents/AppDynamics/LoadTesting/JMeter/TestPlan_Tomcat.jmx) - Apache JMeter (3.1 r1770033)

00:04:44 0 2 / 5

- AppDynamics for Tomcat
  - View Results Tree
  - View Results in Table
  - HTTP Request Defaults
  - HTTP Cookie Manager
  - HTTP Header Manager
  - setUp Thread Group
  - My Thread Group A
  - setUp Thread Group
  - tearDown Thread Group
    - AppDynamics – End time – JSR223 Sampler
    - AppDynamics – Create time range – JSR223 Samp
- WorkBench

Test Plan

Name: AppDynamics for Tomcat

Comments:

User Defined Variables

Name:	Value

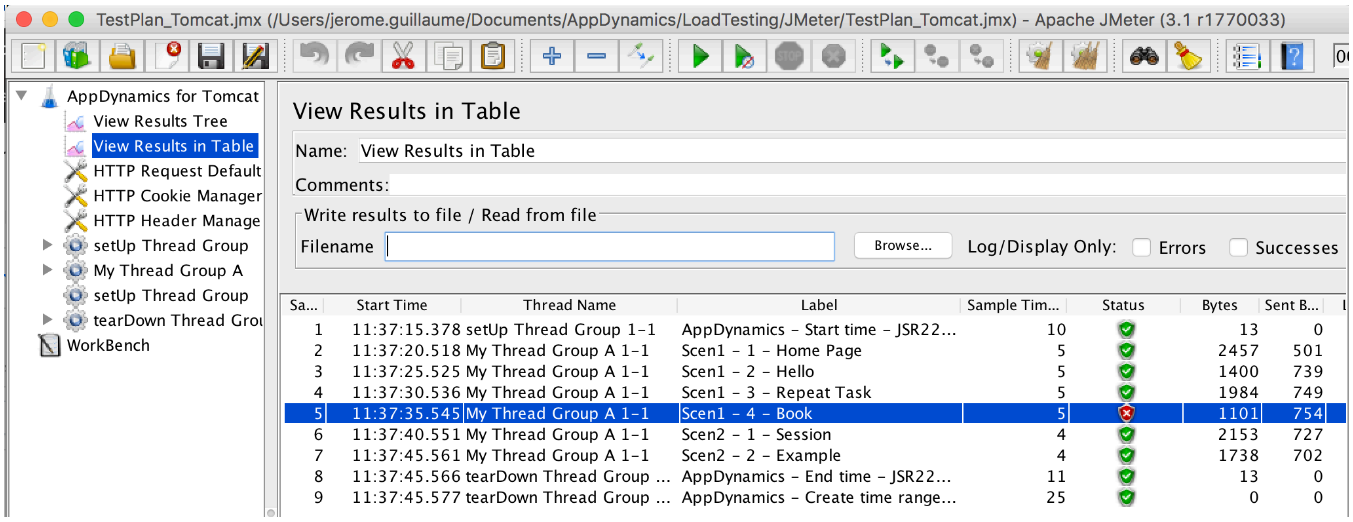
- [✓] Run Thread Groups consecutively (i.e. run groups one at a time)
- [✓] Run tearDown Thread Groups after shutdown of main threads
- [ ] Functional Test Mode (i.e. save Response Data and Sampler Data)

How to configure AppDynamics (with JMeter)

Make a browser refresh of AppDynamics Controller page to see the new Custom time range

3) Retrieve the JMeter Thread name within the AppDynamics snapshot

Here, the thread name `My Thread Group A 1-1` has failed on request `Scen1 – 4 – Book` : AppDynamics (with the corresponding snapshot) explains why there is a failure.



TestPlan_Tomcat.jmx (/Users/jerome.guillaume/Documents/AppDynamics/LoadTesting/JMeter/TestPlan_Tomcat.jmx) - Apache JMeter (3.1 r1770033)

- AppDynamics for Tomcat
  - View Results Tree
  - View Results in Table
  - HTTP Request Default
  - HTTP Cookie Manager
  - HTTP Header Manage
  - setUp Thread Group
  - My Thread Group A
  - setUp Thread Group
  - tearDown Thread Grou
- WorkBench

View Results in Table

Name: View Results in Table

Comments:

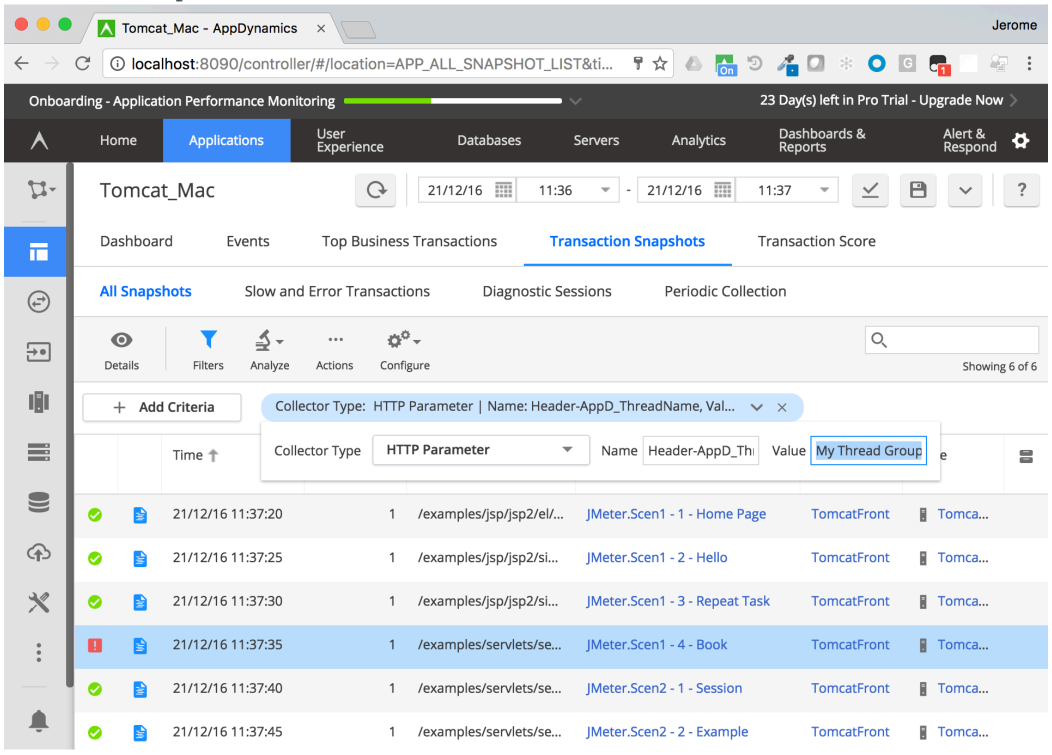
Write results to file / Read from file

Filename | Browse... | Log/Display Only: Errors Successes

Sa...	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent B...
1	11:37:15.378	setUp Thread Group 1-1	AppDynamics – Start time – JSR22...	10	✓	13	0
2	11:37:20.518	My Thread Group A 1-1	Scen1 – 1 – Home Page	5	✓	2457	501
3	11:37:25.525	My Thread Group A 1-1	Scen1 – 2 – Hello	5	✓	1400	739
4	11:37:30.536	My Thread Group A 1-1	Scen1 – 3 – Repeat Task	5	✓	1984	749
5	11:37:35.545	My Thread Group A 1-1	Scen1 – 4 – Book	5	✗	1101	754
6	11:37:40.551	My Thread Group A 1-1	Scen2 – 1 – Session	4	✓	2153	727
7	11:37:45.561	My Thread Group A 1-1	Scen2 – 2 – Example	4	✓	1738	702
8	11:37:45.566	tearDown Thread Group ...	AppDynamics – End time – JSR22...	11	✓	13	0
9	11:37:45.577	tearDown Thread Group ...	AppDynamics – Create time range...	25	✓	0	0

To retrieve all AppDynamics snapshots in regards of a JMeter Tread name:

- Open Transaction Snapshots page
- Set a criteria with Collector Type=`HTTP Parameter`, Name=`Header-AppD_ThreadName` and Value= `My Thread Group A 1-1`



Tomcat_Mac - AppDynamics — Jerome

localhost:8090/controller/#/location=APP_ALL_SNAPSHOT_LIST&ti...

Onboarding - Application Performance Monitoring — 23 Day(s) left in Pro Trial - Upgrade Now

Home | Applications | User Experience | Databases | Servers | Analytics | Dashboards & Reports | Alert & Respond

Tomcat_Mac — 21/12/16 11:36 - 21/12/16 11:37

Dashboard | Events | Top Business Transactions | Transaction Snapshots | Transaction Score

All Snapshots | Slow and Error Transactions | Diagnostic Sessions | Periodic Collection

Details | Filters | Analyze | Actions | Configure — Showing 6 of 6

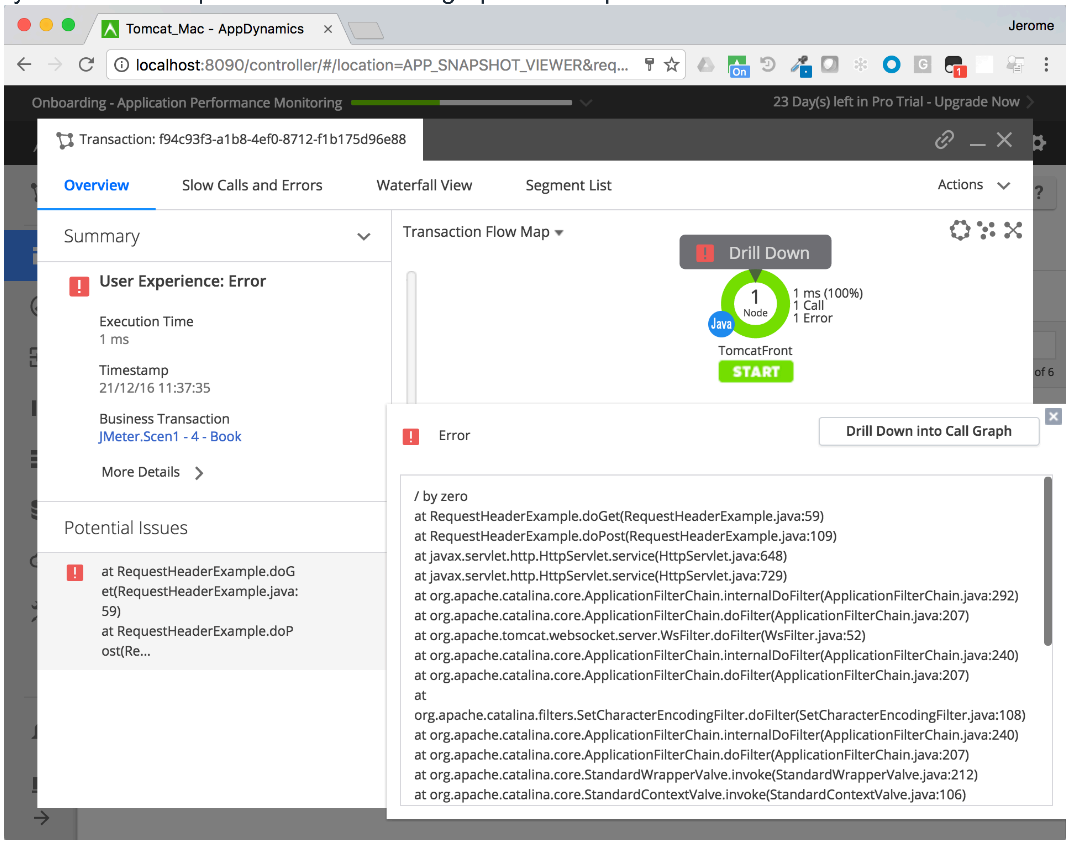
+ Add Criteria — Collector Type: HTTP Parameter | Name: Header-AppD_ThreadName, Val...

Collector Type: HTTP Parameter — Name: Header-AppD_Thi — Value: My Thread Group

		Time ↑					
✓	📄	21/12/16 11:37:20	1	/examples/jsp/jsp2/el/...	JMeter.Scen1 - 1 - Home Page	TomcatFront	Tomca...
✓	📄	21/12/16 11:37:25	1	/examples/jsp/jsp2/si...	JMeter.Scen1 - 2 - Hello	TomcatFront	Tomca...
✓	📄	21/12/16 11:37:30	1	/examples/jsp/jsp2/si...	JMeter.Scen1 - 3 - Repeat Task	TomcatFront	Tomca...
!	📄	21/12/16 11:37:35	1	/examples/servlets/se...	JMeter.Scen1 - 4 - Book	TomcatFront	Tomca...
✓	📄	21/12/16 11:37:40	1	/examples/servlets/se...	JMeter.Scen2 - 1 - Session	TomcatFront	Tomca...
✓	📄	21/12/16 11:37:45	1	/examples/servlets/se...	JMeter.Scen2 - 2 - Example	TomcatFront	Tomca...

- Open the snapshot of Business Transaction `JMeter.Scen1 – 4 – Book` which has the **Error** status ❗

- Analyze the flow map and access to call-graph or exception detail



Transaction: f94c93f3-a1b8-4ef0-8712-f1b175d96e88

User Experience: Error

Execution Time
1 ms

Timestamp
21/12/16 11:37:35

Business Transaction
JMeter.Scen1 - 4 - Book

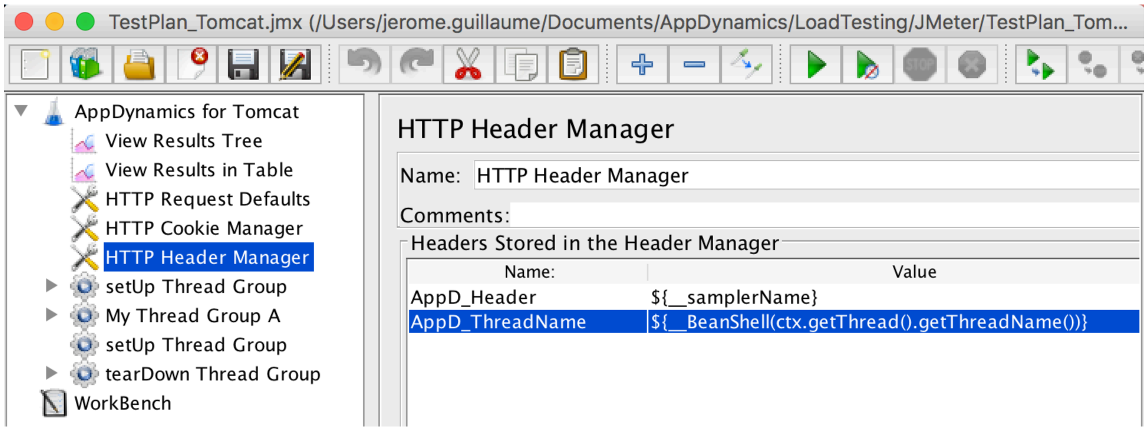
at RequestHeaderExample.doGet(RequestHeaderExample.java:59)
at RequestHeaderExample.doPost(Re...

Error

```
/ by zero
at RequestHeaderExample.doGet(RequestHeaderExample.java:59)
at RequestHeaderExample.doPost(RequestHeaderExample.java:109)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:648)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:729)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:292)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:207)
at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:240)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:207)
at
org.apache.catalina.filters.SetCharacterEncodingFilter.doFilter(SetCharacterEncodingFilter.java:108)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:240)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:207)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:212)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:106)
```

How to configure JMeter (with AppDynamics)

- Start JMeter and open the Test Plan
- Open the `HTTP Header Manager`
- Add a header
  - Set name with `AppD_ThreadName` and value with `${__BeanShell(ctx.getThread().getThreadName())}` and click on Save
- See below the expected result



HTTP Header Manager

Name: HTTP Header Manager

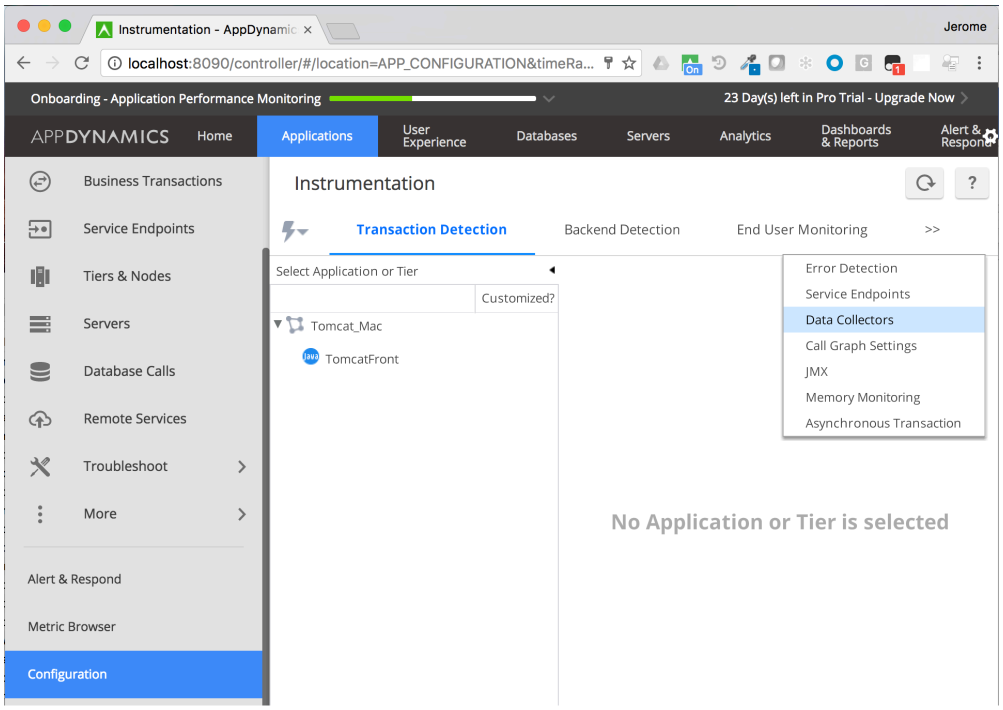
Comments:

Headers Stored in the Header Manager

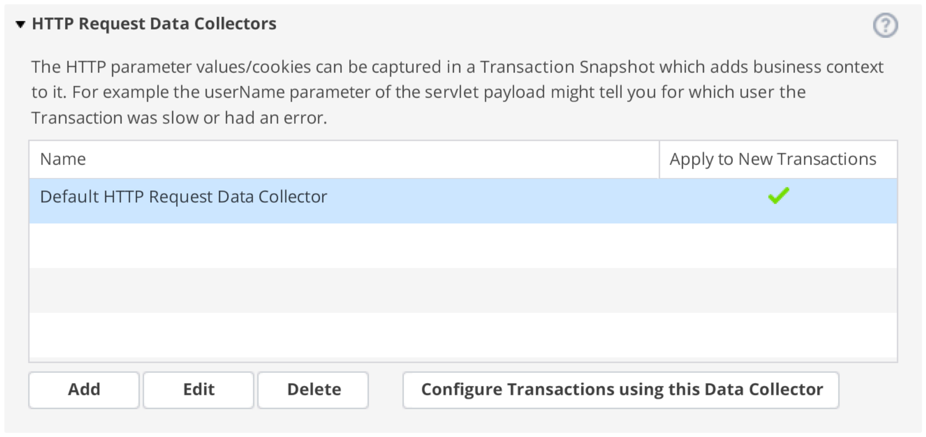
Name:	Value
AppD_Header	${__samplerName}
AppD_ThreadName	${__BeanShell(ctx.getThread().getThreadName())}

How to configure AppDynamics (with JMeter)

- Connect to AppDynamics Controller
- Select the Application
- Click on Configuration
- Click on Instrumentation
- Click on >> and Data Collectors



- Select Default HTTP Request Data Collectors and click on Edit

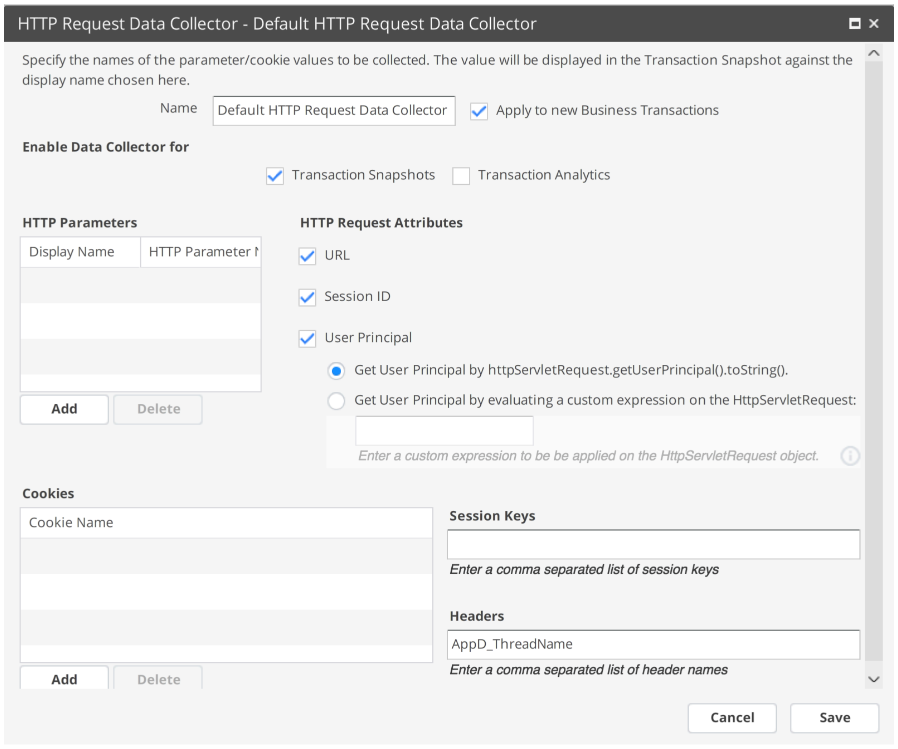


HTTP Request Data Collectors

The HTTP parameter values/cookies can be captured in a Transaction Snapshot which adds business context to it. For example the userName parameter of the servlet payload might tell you for which user the Transaction was slow or had an error.

Name	Apply to New Transactions
Default HTTP Request Data Collector	✓

- Set a Header with the value `AppD_ThreadName`
- Click on Save



HTTP Request Data Collector - Default HTTP Request Data Collector

Specify the names of the parameter/cookie values to be collected. The value will be displayed in the Transaction Snapshot against the display name chosen here.

Name: Default HTTP Request Data Collector [✓] Apply to new Business Transactions

Enable Data Collector for

[✓] Transaction Snapshots [ ] Transaction Analytics

HTTP Parameters

Display Name	HTTP Parameter N

Add Delete

HTTP Request Attributes

[✓] URL

[✓] Session ID

[✓] User Principal

● Get User Principal by httpServletRequest.getUserPrincipal().toString().

○ Get User Principal by evaluating a custom expression on the HttpServletRequest:

Enter a custom expression to be be applied on the HttpServletRequest object.

Cookies

Cookie Name

Add Delete

Session Keys

Enter a comma separated list of session keys

Headers

AppD_ThreadName

Enter a comma separated list of header names

Cancel Save