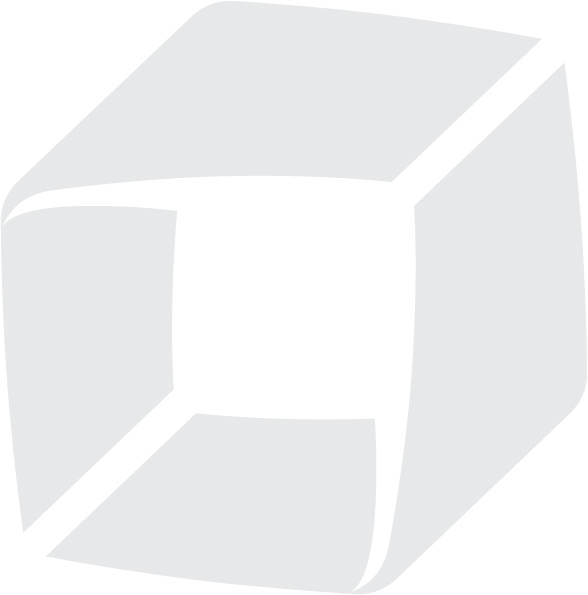
­

Dynatrace Intermediate Training Reference

Updated <Date>



# OVerview

This document details the dashboards and views that are covered in the Dynatrace Intermediate Training. The aim of this training is to enable users to view and interpret transaction level details to analyze root causes of outages and improve application performance.

Table of Contents

* Application Health – General Dashboard 3
  + Accessing the General Dashboard 3
  + PurePaths Dashlet4
    - PurePaths Dashlet Components5
    - PurePath Metrics Reference9
    - PurePath States Reference10
  + Web Requests Dashlet12
    - Web Request Metrics Reference12
  + Method Hotspots Dashlet14
  + Response Time Hotspots Dashlet15
    - Response Time Formulas16
  + Errors Dashlet16
  + Exceptions Dashlet16
* User Experience Management 17
  + Apdex17
  + User Experience Index17
  + User Actions18
  + User Action PurePaths19
  + Visits21
* Navigating the Data 22
  + Timeframes 22
  + Filters23
  + Drill Down 23

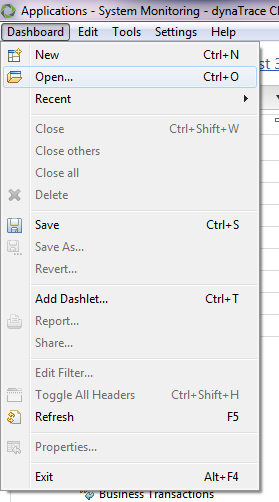
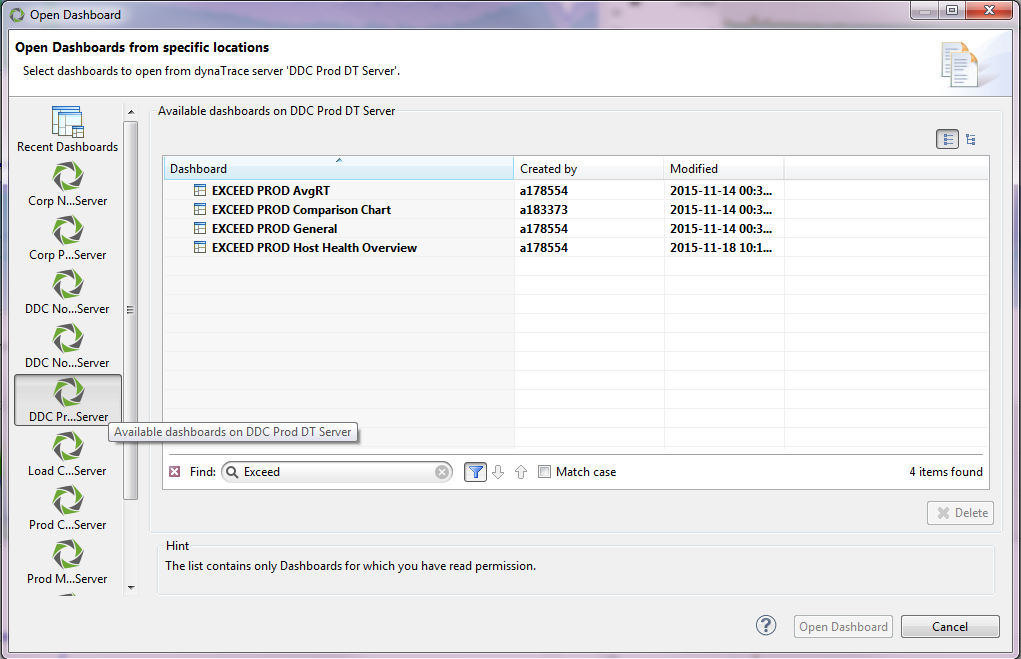
# Application Health – GeNERAL DAShboard

The General Dashboard provides deep level transactional data for an application. Many of the dashlets will be learned in future training, however, the Transaction Flow that is found here is a good transactional health overview.

Note: The General Dashboard contains very granular data. Due to this, opening shorter periods of time is highly recommended to reduce loading times and Dynatrace limits the number of transactions analyzed at one time.

Accessing the General Dashboard:

1. Click Dashboard menu in the Dynatrace Client and select open.
2. Choose the appropriate Dynatrace server for your application/environment on the left side of the pop up.
3. In the dashboards window, type the name of the application to quickly filter.
4. Locate and select the Application’s General dashboard and click open.

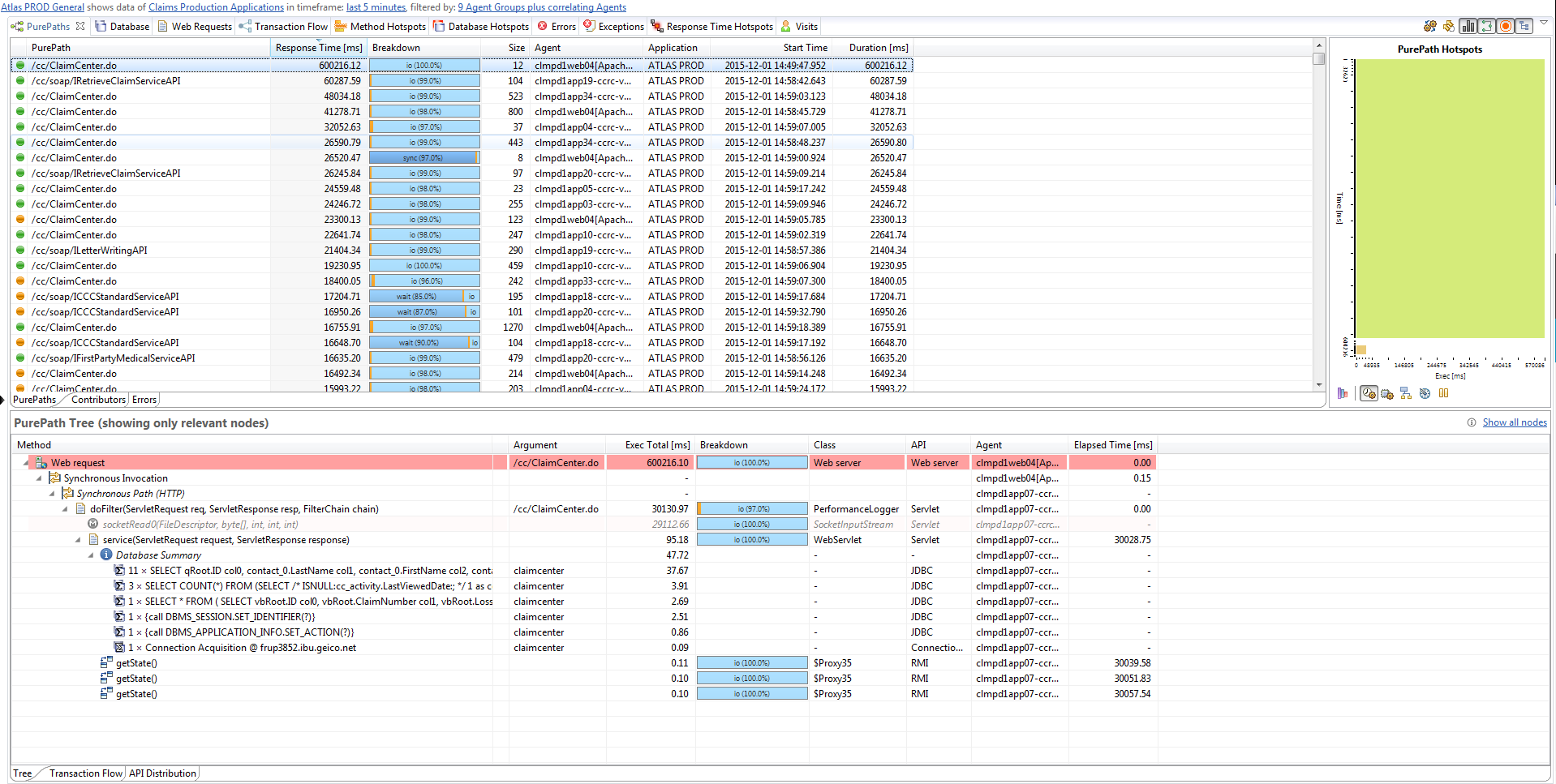
 

PurePaths Dashlet

PurePaths are the end to end transactions, from the user, through the datacenter, which enables Dynatrace to provide transactional and method level detail into application performance.

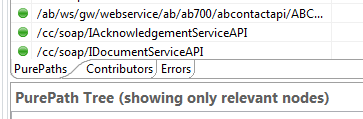
The **PurePaths** Dashlet is the central representation of all captured PurePaths. You can use this dashlet to identify performance-critical methods and analyze their position within the PurePath, to recognize the cause of a performance problem. Because all special diagnosis dashlets can be traced back to the PurePathsdashlet, it is the basis of performance analysis.

The **PurePaths** Dashlet can be opened from the General Dashboard by clicking the **PurePaths** Tab shown below.

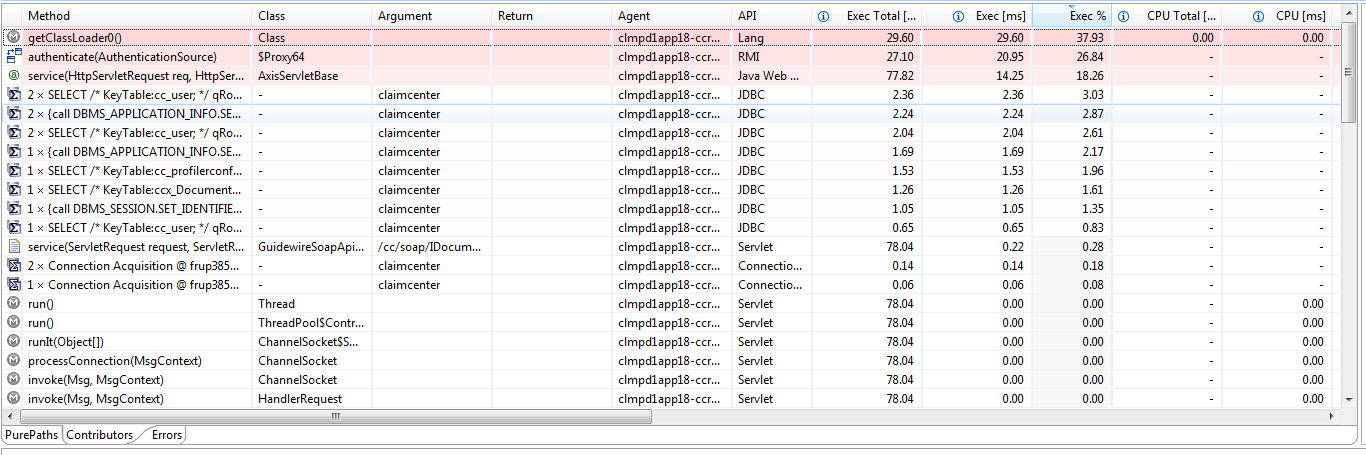


PurePaths Dashlet Components

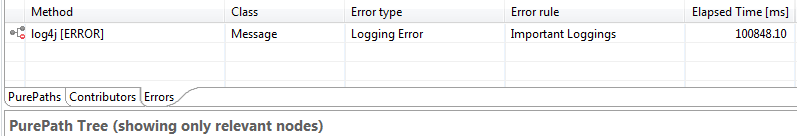
The **PurePaths Tab** is the basis for all components in this dashlet and displays all of the recorded PurePaths in the selected time frame.



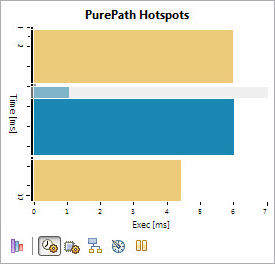
The **Contributors Tab** lists all instrumented methods of the selected PurePath. Depending on the sort column, the most runtime-critical methods are highlighted by background coloring.



The **Errors Tab** shows the errors that were detected on the selected PurePath.



The **PurePath Hot Spots** pane gives a visual overview of the currently selected PurePath. It displays duration metrics for the natural tree order of the PurePath contributors. Because there usually are more contributors than space available to render them, minor contributors are aggregated. Click a hot spot to select the corresponding calls in the **Tree** tab, and vice versa.



The vertical axis defines the execution time. The horizontal axis is the metric selected in the toolbar below the chart:

https://community.dynatrace.com/community/download/attachments/182356666/exec.gif?version=1&modificationDate=1245230348213&api=v2 **Execution**

https://community.dynatrace.com/community/download/attachments/182356666/exec_cpu.gif?version=1&modificationDate=1245230348213&api=v2 **CPU Execution**

https://community.dynatrace.com/community/download/attachments/182356666/sync.gif?version=1&modificationDate=1245230348197&api=v2 **Synchronization**

https://community.dynatrace.com/community/download/attachments/182356666/clock.gif?version=1&modificationDate=1245230348197&api=v2 **Wait**

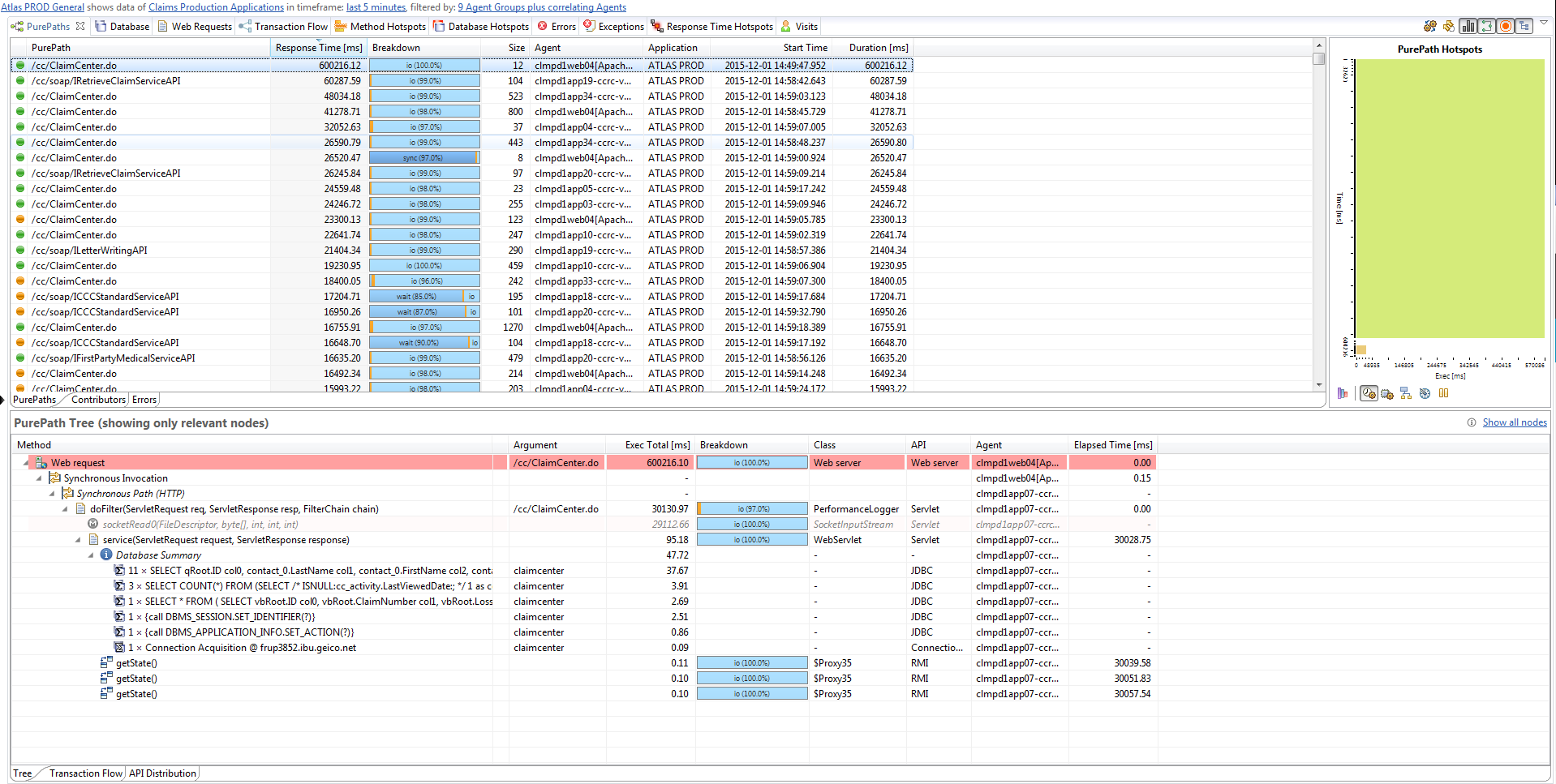
* **Suspension**

The bar color is defined by the most significant API of each method call; a gray bar indicates a set of aggregated APIs.

The **PurePath Hotspots** pane can also display the https://community.dynatrace.com/community/download/attachments/182356666/hierarchy_api.gif?version=1&modificationDate=1245234216547&api=v2 **API Call Hierarchy**. This option renders the hierarchy of the PurePath contributors, colored by the individual call's API, to give further insight into the PurePath's structure.

The pane's context menu gives access to the Details summary. This simplified version of the regular contributor details displays call statistics only. When you select a hotspot for multiple contributors, aggregated values are shown.

The **Tree** tab displays the contributors to the currently selected PurePath arranged as a tree that maps the calling hierarchy of the contributing methods.

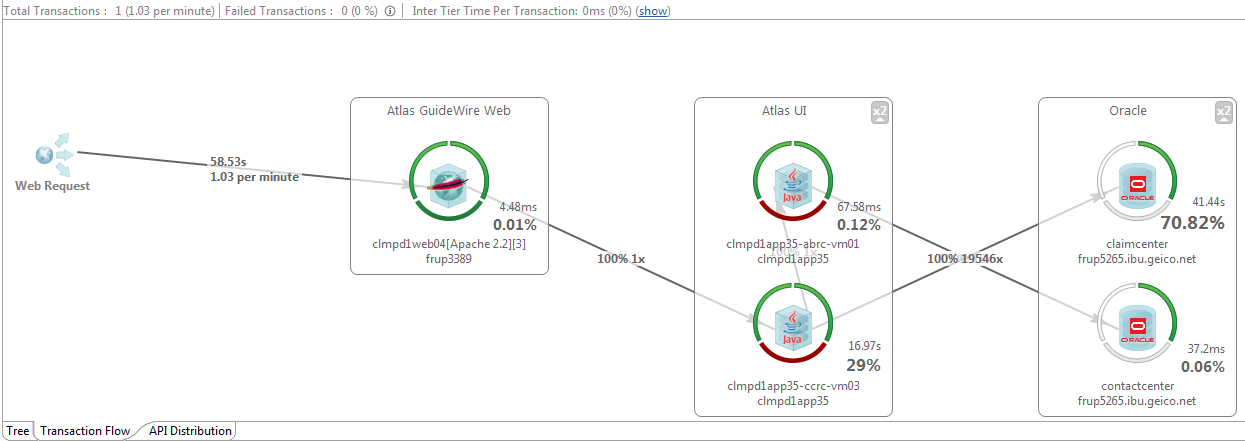


The whole tree or parts of it can be expanded or collapsed to focus on the important methods calls and to hide parts of the PurePath not relevant for the current performance diagnosis task.

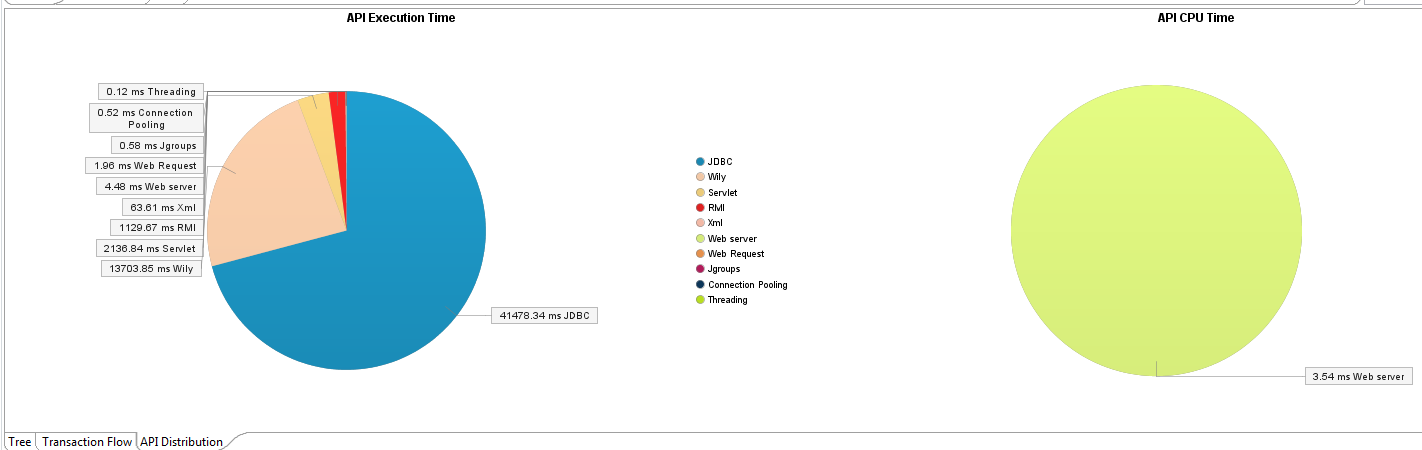
The call hierarchy of the selected PurePath can span multiple threads or processes, depending on the applied sensors. To indicate thread or process changes within a PurePath, the following special nodes are inserted within the tree view:

* Synchronous Path – A node that groups the nodes of the child path that is executed synchronously on another Agent/thread. The parent node of this Synchronous Path node blocks until the child path has finished its execution.
* Asynchronous Path – A node that groups the nodes of the child path that is executed asynchronously on another Agent/thread.
* Asynchronous Invocation – A grouping node of one or more Asynchronous Paths as a single call (e.g. sending one JMS message) that can cause several threads/Agents to start processing (e.g. several subscribers on a JMS topic). A Synchronous Invocation grouping node is not shown for a Synchronous Path, because all synchronous invocations lead to exactly one Synchronous Path.

The **Transaction Flow** tab displays a diagram of the distribution of selected PurePaths over components or tiers of the system monitored by distinct Dynatrace Agent instances. The Transaction Flow components involved in the execution of a selected PurePath are highlighted. PurePaths that span multiple components are indicated by dashed lines between the involved components. If several tiers are involved in a PurePath and if they communicate with one another directly, the communication channels are visualized by lines between communicating tiers.



The **API Distribution** tab provides an aggregated view of the distribution of execution time, CPU utilization, and synchronization over the APIs used by the selected PurePath(s).



PurePaths Metrics Reference

The **PurePaths** table displays the following performance metrics:

* **Error State** − The first, unlabeled column displays icons that indicate the state of the PurePath. See PurePath States for more information.
* **PurePath** − A string representing the PurePath identifier.
* **Response Time [ms]** − The total execution time for the PurePath in milliseconds, excluding asynchronous paths that may take longer independently.
* **Exec Sum [ms]** − The overall time spent executing any captured method within the PurePath.
* **Breakdown** − The breakdown in percentage of CPU, Sync, Wait, and I/O times.   
  A high I/O time doesn't necessarily mean that the application spends its time in I/O operations, because the I/O time also contains kernel time.
* **Size** − The number of method calls within a PurePath. Only instrumented methods are included.
* **Agent** − The name of the Agent from which this PurePath starts.
* **Application** − The application to which the PurePath belongs. This is the first matched application, as defined in the System Profile.
* **Start Time** − The time when this PurePath started.
* **Transactions** − The names of the Business Transactions applied to the PurePath.
* **Duration [ms]** − The time from the first method until the last method completed, including asynchronous method calls that may finish after the first method completed.
* **CPU Sum [ms] −** The overall required CPU time for executing this path. This value does not have to be equivalent to the total CPU time of the first method of this path if more than one thread is involved in the PurePath.
* **Sync Sum [ms]** − The overall required sync time (e.g. wait for monitors) for executing this path. This value is equivalent to total sync time of the first method of this path if more than one thread is involved in the PurePath. Only instrumented methods for which synchronization time recording is enabled are measured.
* **Wait Sum [ms]** − The overall required wait time (e.g. waiting for other threads or I/O) for executing this path. This value is equivalent to the total wait time of the first method of this path if more than one thread is involved in the PurePath. Only instrumented methods for which synchronization time recording is enabled are measured.
* **Suspension Sum [ms]** − The time the path was in suspension (typically garbage collector runs).
* **Agent ID** − The numerical ID of the Agent instance from which this PurePath starts.
* **Tag ID** − The continuous numerical ID of PurePaths recorded by the given Agent.
* **Thread Name** − The name of the root path's thread.
* **End Time** − The time with the PurePath ended.
* **Web Request Response Time [ms]** − The processing time required to complete a web request, including (if applicable) Servlet 3 asynchronous processing time.
* **Client IP** − The client IP- address for web requests.

Note: Hovering over any column header in Dynatrace will bring up the description of that column. By default, not all metrics are included in the table. To select which columns to display, right-click a column head and select or clear the column checkboxes as needed.

PurePath States Reference

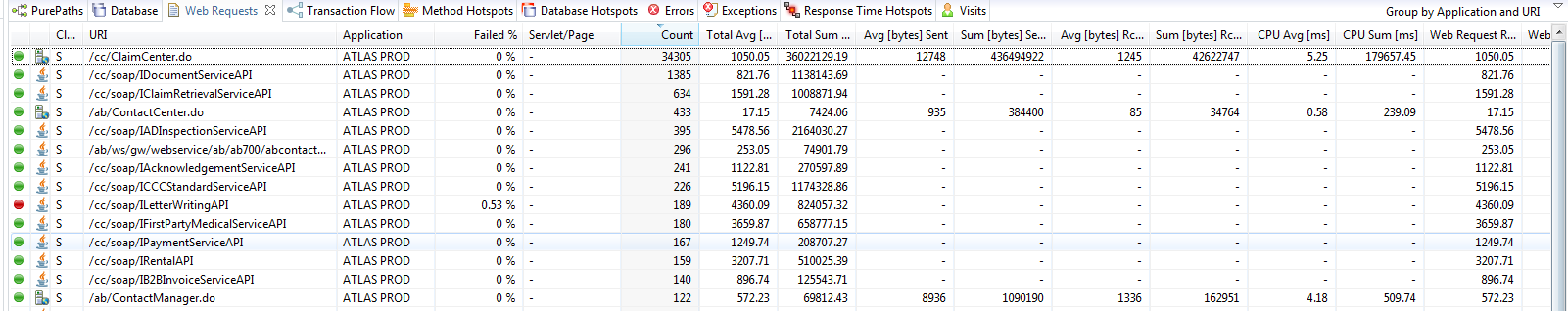
The icons in the first column of the **PurePaths** table indicate the state of the PurePath:

* https://community.dynatrace.com/community/download/attachments/196642776/green%20bullet.png?version=1&modificationDate=1385086603190&api=v2 **Complete** – The PurePath is completed. All expected events for this PurePath were received, and it is no longer waiting for new events.
* https://community.dynatrace.com/community/download/attachments/196642776/asynchronous_status_example.png?version=1&modificationDate=1369104379587&api=v2 **Asynchronous Complete** – A PurePath on which asynchronous Servlet processing took place is completed.
* https://community.dynatrace.com/community/download/attachments/196642776/worddaved35bc2970466ece398fbb5b7590d3b2.png?version=1&modificationDate=1225784556860&api=v2 **Processing** – The PurePath is currently being processed. Additional events describing this PurePath are expected. Data for PurePaths in this state is not used for other analysis dashlets than the PurePath dashlet.
* https://community.dynatrace.com/community/download/attachments/196642776/Processing%20-%20Timed%20Out.png?version=1&modificationDate=1385089650670&api=v2 **Processing** – The PurePath timed out while being processed.
* https://community.dynatrace.com/community/download/attachments/196642776/worddavd899c78fd43bd8a9f30c117c6fa05cdd.png?version=1&modificationDate=1225784556920&api=v2 **Waiting** – A completed PurePath received all expected events but is still waiting for possible events created by asynchronously executed parts of the PurePath. (See Note 1, below.) If any Agent crashes, this state might remain until the Agent is removed from the Dynatrace Server by a timeout (default 30 seconds), because this Agent might have been called by existing PurePaths.
* https://community.dynatrace.com/community/download/attachments/196642776/worddav773c52ffedd7da5e2962d5281fe43ca0.png?version=1&modificationDate=1225784556953&api=v2 **Timeout** – A PurePath did not receive all expected events within the timeout period. One of the following applies:
  + At least one method within the PurePath has not returned, and for the configured timeout and no new data has arrived for the PurePath.
  + The PurePath is a long-running transaction that has not been finished within the Maximum PurePath Age timeout.
* https://community.dynatrace.com/community/download/attachments/196642776/Path%20Timed%20Out.png?version=1&modificationDate=1385087936377&api=v2 **Path Timed Out** – The PurePath did not finish in time.
* https://community.dynatrace.com/community/download/attachments/196642776/purepath_complete_corrupt.gif?version=1&modificationDate=1288270269420&api=v2 **Corrupted** – A problem occurred while creating the PurePath. PurePaths marked as corrupted are not used for other analysis views than the PurePath view. Corrupted PurePaths are not analyzed at all: they do not provide any measurements or incidents. A PurePath can be corrupted when:
  + At least one event carrying data for this PurePath has been skipped. Check the PurePath details for the number of skipped events. Events are skipped when they are not read fast enough from the instrumented application where they were created. This can be caused by:
  + Slow network between the instrumented Application and the Dynatrace Collector
  + Slow network between the Dynatrace Collector and the Dynatrace Server
  + Too much instrumentation on one or more instrumented applications
  + The instrumented application was shut down while the PurePath was in the **Processing** state.
* https://community.dynatrace.com/community/download/attachments/196642776/worddav2daf97e3246a2b45db438e96041520b4.png?version=1&modificationDate=1225784557297&api=v2 **Incomplete** – The PurePath was stored in a stored session while the PurePath was in the Processing or Waiting state. It cannot be guaranteed that the PurePath is completed. Incomplete PurePaths can only occur in stored sessions.
* https://community.dynatrace.com/community/download/attachments/196642776/orange%20bullet.png?version=1&modificationDate=1385087223407&api=v2  **Error** – The PurePath contains an error.
* https://community.dynatrace.com/community/download/attachments/196642776/gray%20bullet.png?version=1&modificationDate=1385085866343&api=v2 **Path Corrupted** – It is not possible to determine PurePath success because the PurePath is corrupted. Either an involved Agent may have been shut down before the PurePath completed, or a node may contain corrupted information.
* https://community.dynatrace.com/community/download/attachments/196642776/red%20bullet.png?version=2&modificationDate=1385085857660&api=v2 **Transaction Failed** – The transaction failed.

Web Requests Dashlet

The **Web Requests** dashlet displays collected information about web requests for diagnosing and tuning software components that handle web requests like servlets and JSPs.

The **Web Requests** dashlet can be opened from the General Dashboard by clicking the **Web Requests** Tab shown below.



Web Request Metrics Reference

These columns are available in the dashlet. Not all columns are displayed by default. To select which columns to display, right-click a column head and select or clear the columns as needed.

* **Error State** **–** The first, unlabeled column displays icons that indicate the overall error state of the aggregation. For example, a green icon indicates that no errors occurred in the aggregated line
* **Platform -** The second, unlabeled column displays icons that identify the software platform that executed the web request: Java or .NET.
* **Client or Server -** Indicates whether the client side (**C**), server side (**S**) or both sides (**C/S**) of the web request call were detected.
* **URI -** The part of the URL that identifies the web request.
* **Application -** The application associated with the web request.
* **Query -** The query part that is contained in the web request URL.
* **Tier -** The tier (Agent Group) that processed the web request.
* **Web Application -** For servlet requests, the name of the web application as described in the web.xml descriptor. For ASP.NET pages, the name of the virtual application.
* **Servlet/Page -** For servlet requests, the name of the servlet as specified in the web.xml descriptor. For ASP.NET pages, the name of the requested page.
* **Failed% -** The percentage of failed web requests.
* **Count –** The number of times the web requests was called.
* **Total Avg [ms]** **–** The average time, in milliseconds, spend on the web request.
* **Total Sum [ms] –** The accumulated time, in milliseconds, spent on the web request.
* **Avg [bytes] Sent -** The average number of bytes sent as a response from the web request.
* **Sum [bytes] Sent -** The accumulated number of bytes sent as a response from all web requests.
* **Avg [bytes] Rcvd -** The average number of bytes received as a request from the web request.
* **Sum [bytes]Rcvd -** The accumulated number of bytes received as a request from all web requests.
* **Sum [bytes]Rcvd -** The accumulated number of bytes received as a request from all web requests.
* **CPU Avg [ms] -** The average CPU time, in milliseconds, used for all invocations of the web request.
* **CPU** **Sum [ms] -** The accumulated CPU time, in milliseconds, used for all invocations of the web request.
* **Total Min [ms] -** The minimum time, in milliseconds, spent in a call to this web request.
* **Total Max [ms] -** The maximum time, in milliseconds, spent in a call to this web request.
* **Exec Avg [ms] -** The average time, in milliseconds, spent to call the web request, excluding time spent in nested calls to other methods.
* **Exec Min [ms] -** The minimum time, in milliseconds, spent in a call to this web request, excluding time spent in nested calls to other methods.
* **Exec Max [ms] -** The maximum time, in milliseconds, spent in a call to this web request, excluding time spent in nested calls to other methods.
* **Exec Sum [ms]** ***-*** The accumulated time, in milliseconds, spent for all previous calls of this web request, excluding time spent in nested calls to other methods.
* **Min [bytes] Sent -** The minimum number of bytes sent as a response from the web request.
* **Max [bytes] Sent -** The maximum number of bytes sent as a response from the web request.
* **Min [bytes] Rcvd -** The minimum number of bytes received as a request to the web request.
* **Max [bytes] Rcvd** **-** The maximum number of bytes received as a request to the web request.
* **CPU Min [ms] -** The minimum CPU time, in milliseconds, used for an invocation of this web request.
* **CPU Max [ms]** **-** The maximum CPU time, in milliseconds, used for an invocation of this web request.
* **Web Request Response Time Avg [ms]** **-** The average time, in milliseconds, spent on this web request, including (if applicable) Servlet 3 asynchronous processing time.
* **Web Request Response Time Sum [ms] -** The accumulated time, in milliseconds, spent on this web request, including (if applicable) Servlet 3 asynchronous processing time.
* **Web Request Response Time Min [ms]** **-** The minimum time, in milliseconds, spent in a call to this web request, including (if applicable) Servlet 3 asynchronous processing time.
* **Web Request Response Time Max [ms]** **-** The maximum time, in milliseconds, spent in a call to this web request, including (if applicable) Servlet 3 asynchronous processing time.

Method Hotspots Dashlet

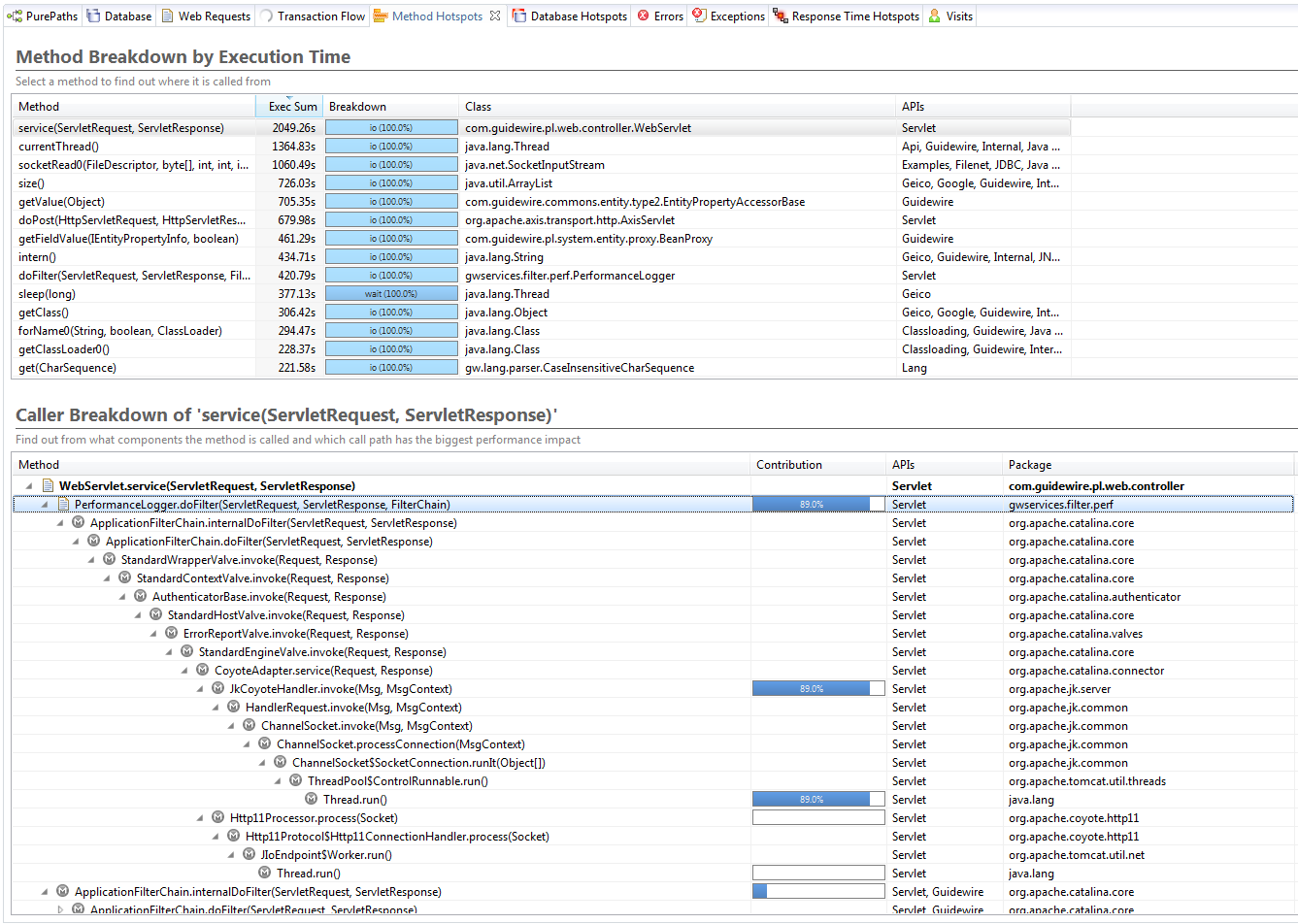
The **Method Hotspots** dashlet shows methods executed during the specified timeframe. The

methods are aggregated according to the breakdown time selected in the dashlet toolbar:

* Execution Time
* CPU Time
* Synchronization Time
* Wait Time

The **Method Breakdown** table in the top part the dashlet lists a maximum of 100 methods, beginning with the slowest method. Selecting a method displays the hierarchy of the method in the **Caller Breakdown** in the lower section of the dashlet. In the **Caller Breakdown**, or *Contribution Tree*, the called method is displayed at the top of the tree, and the callers are shown below the method.

The **Method Hotspots** dashlet can be opened from the General Dashboard by clicking the **Method Hotspots** Tab shown below.

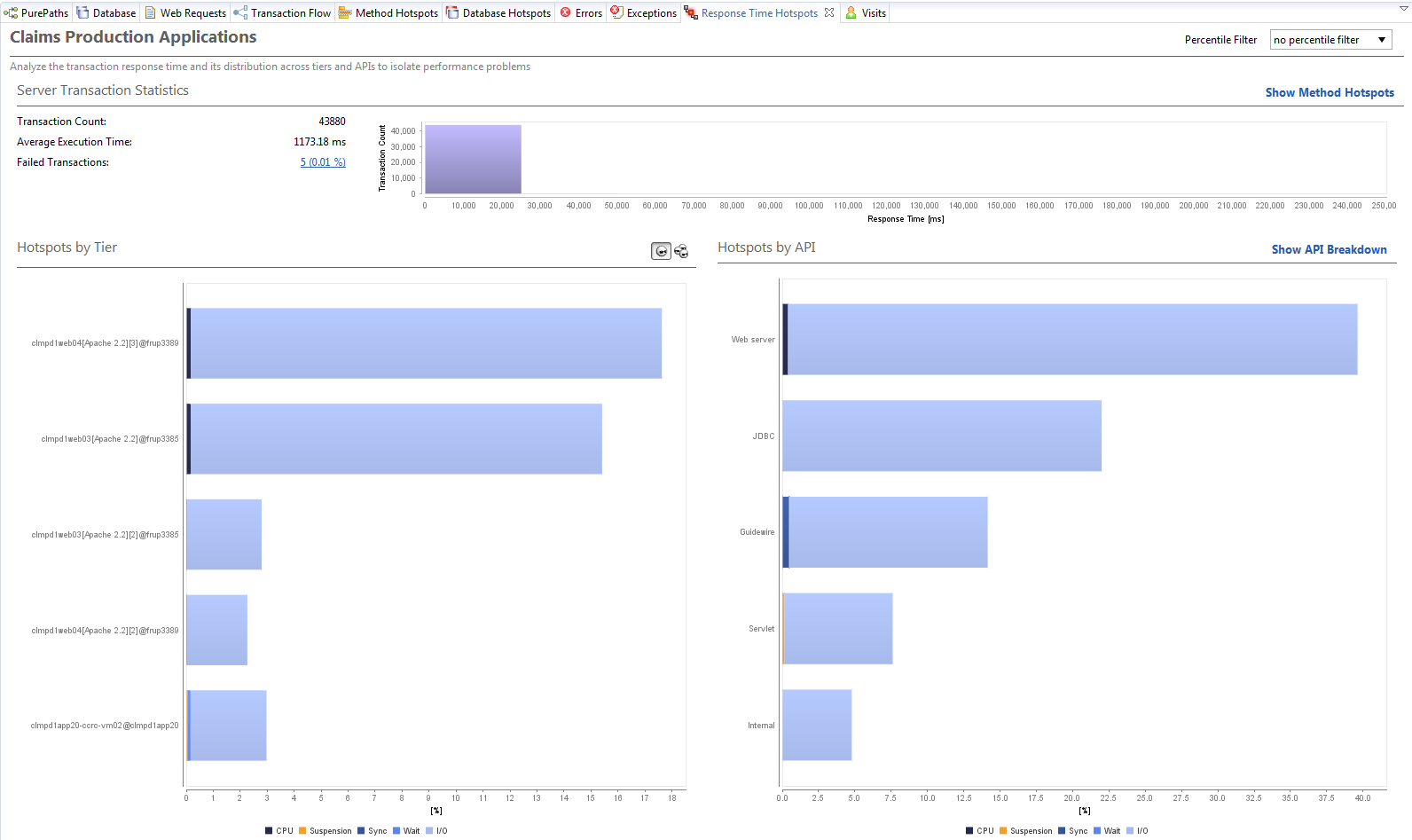


The **Caller Breakdown**, or Contribution Tree, displays the detailed call aggregation of the selected method. The called method is displayed at the top of the tree, and the callers are shown below the method. Because the Contribution Tree is an aggregation of many paths, it is likely that the called method at the top is reached via many different call paths in all involved PurePaths. The Contribution Tree therefore consists of branches that are weighted according to the breakdown time contribution of each tree branch. Branches with high percentage values are typical of greater interest, because these are the caller branches that may be the likeliest sources of performance problems.

Response Time Hotspots Dashlet

The **Response Time Hotspots** dashlet provides a high-level view of a session to give you a quick impression of where the time is spent. A typical use case is to filter for Business Transactions to see how similar transactions are distributed regarding response time and where most of the time is contributed, by tier and by API.

The **Response Time Hotspots** dashlet can be opened from the General Dashboard by clicking the **Response Time Hotspots** Tab shown below.



You can use the filter list in the top right corner of the dashlet to filter the content to the slowest 1%, slowest 5%, slowest 10% or fastest 90% of transactions in the session.

|  |  |
| --- | --- |
| [https://community.dynatrace.com/community/download/thumbnails/182356794/ResponseTimeHotspotsPercentileFilter.png?version=1&modificationDate=1403198501967&api=v2](https://community.dynatrace.com/community/download/attachments/182356794/ResponseTimeHotspotsPercentileFilter.png?api=v2) |  |

Response Time Formulas

When a PurePath is executed, Dynatrace captures CPU time used by the threads that are executing this transaction. It also captures the information on how much time was spent in sync (sync blocks), wait (waiting on an object) and runtime suspension (when garbage collection pauses the thread). The rest of the time is contributed by I/O.

The formula for response time therefore is:

Response Time = CPU + Sync + Wait + Runtime Suspension + I/O

I/O is calculated and everything else is measured:

I/O = Response Time - CPU - Sync - Wait - Runtime Suspension

Errors Dashlet

The **Errors** dashlet shows all the detected errors sorted by error categories. By default, errors can be caused by HTTP response codes, exceptions, logging messages or browser errors.

The **Errors** dashlet can be opened from the General Dashboard by clicking the **Errors** Tab.

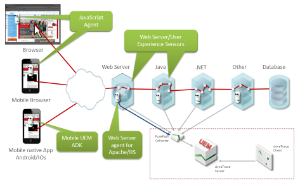
Exception Dashlet

The **Exception** dashlet displays all of the exceptions in the stored session, or in the PurePaths of your live session. You can see the exception class, message, which class threw the exceptions, and how often it occurs.

The **Exception** dashlet can be opened from the General Dashboard by clicking the **Exception** Tab.

# User Experience Management

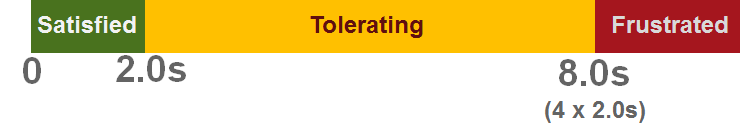
Dynatrace User Experience Management (UEM) captures transactions triggered within browsers (desktop or mobile) or mobile apps from end to end: browser or app, over web server and application server, into the database. This is accomplished through JavaScript injection into web apps or through the Mobile ADK for mobile apps.



Apdex

Apdex is an open, standard way to depict the performance of User Actions. Apdex is a ratio of slow and fast requests. It is application independent and represents performance as a single value between 0 and 1. Everything below .5 is unacceptable everything above .93 is excellent. Use this value to compare the performance of different applications and to understand long-term performance trends.

User actions are mapped to the following zones based on their response time:



User Experience Index

We use the term User Experience rather than Apdex. User experience analyzes how the performance is perceived by the application's users, and categorizes visits into satisfying, tolerating and frustrating, depending on the performance of user actions and errors.

Users have a frustrating experience if:

* Their last action failed ("The Web site does not work - I'm leaving!")
* Their last action was frustrated ("The Web site is too slow - I'm leaving!")
* More than 50% of all actions were frustrating

Users have a satisfying experience if:

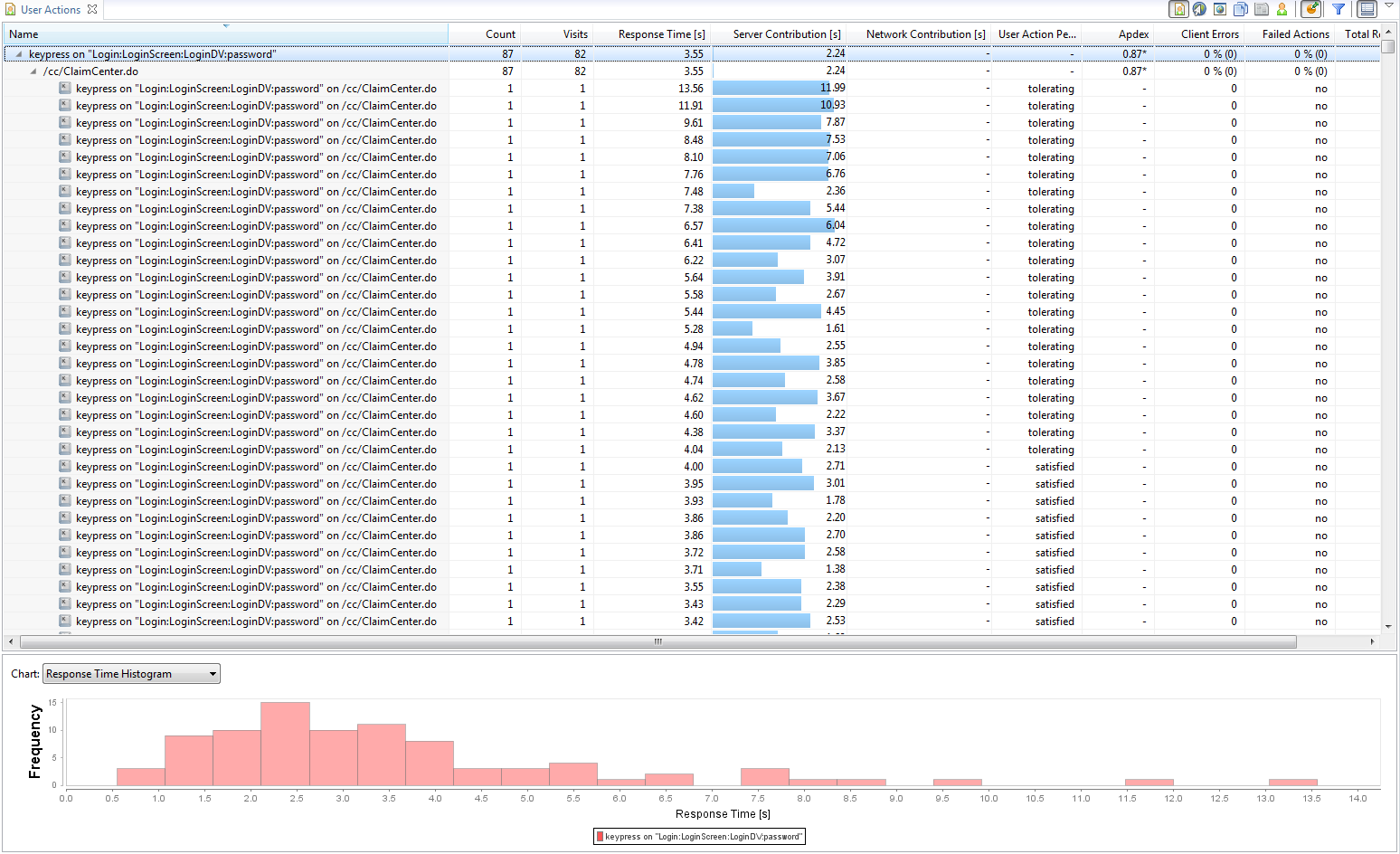
* No action failed
* More than 50% of all actions were satisfying

Users have a tolerating experience if their experience is neither frustrating nor satisfying, which means that:

* Their last action was not frustrating or failed
* Less than 50% of all actions were satisfying
* More than 50% of all actions were at least tolerating

User Actions

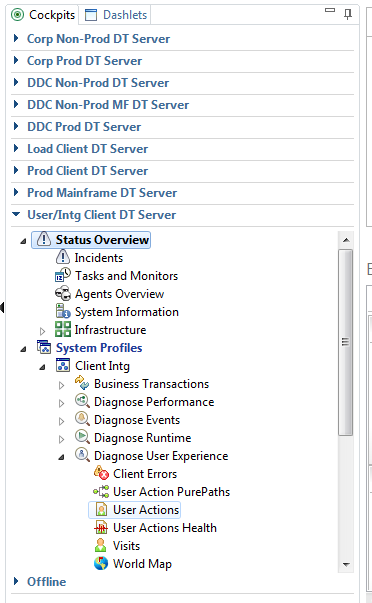
A **user action** contains all client activity during the transition from one state to another. Every user action contains at least one web request. For example, when a user navigates to a page, this causes the browser to download many resources and call the JavaScript onload handler, which in turn might perform XHR requests, etc. The duration of a user action is called response time . It represents the time that the user waits until it is possible to continue with the use case. This means a low response time is better than a high one.



The User Action dashlet has been added to the General Dashboards for the applications that have UEM enabled. In this case, it will be another tab as all of the other dashlets in the General Dashboard.

If the User Action dashlet has not yet been added, it can be opened as follows:

* In the Cockpit, select the **System Profile** for the appropriate application.
* Expand the **Diagnose User Experience** node.
* Double-click **User Actions** to open the dashlet*.*



User Action PurePaths

The **User Action PurePaths** dashlet represents all captured PurePaths, which begins at the start of an action like loading a page in the browser. Use this to analyze user actions like page loading or button clicks. It contains all web requests to instrumented system components (web server, application server). You can analyze all actions that begin in the browser to track functional problems and user performance issues.

The User Action dashlet has been added to the General Dashboards for the applications that have UEM enabled. In this case, it will be another tab as all of the other dashlets in the General Dashboard.

If the User Action PurePaths dashlet has not yet been added, it can be opened as follows:

* In the Cockpit, select the **System Profile** for the appropriate application.
* Expand the **Diagnose User Experience** node.
* Double-click **User Action PurePaths** to open the dashlet*.*

The dashlet is divided into upper and lower areas. The upper part consists of the following tabbed views:



PurePaths: A list of PurePaths, each with corresponding dashboard or dashlet filter (timeframe, application, etc.), and basic information (name, response time, start time) for each PurePath.



Contributors: A list of all contributing actions and methods of the selected PurePath. Colors represent contributors with the highest execution times. The selected contributor is represented in the tree in the lower part of the dashlet.



Errors: All errors in the selected PurePath. Errors are defined via several rules.

The lower part consists of the following tabbed views:



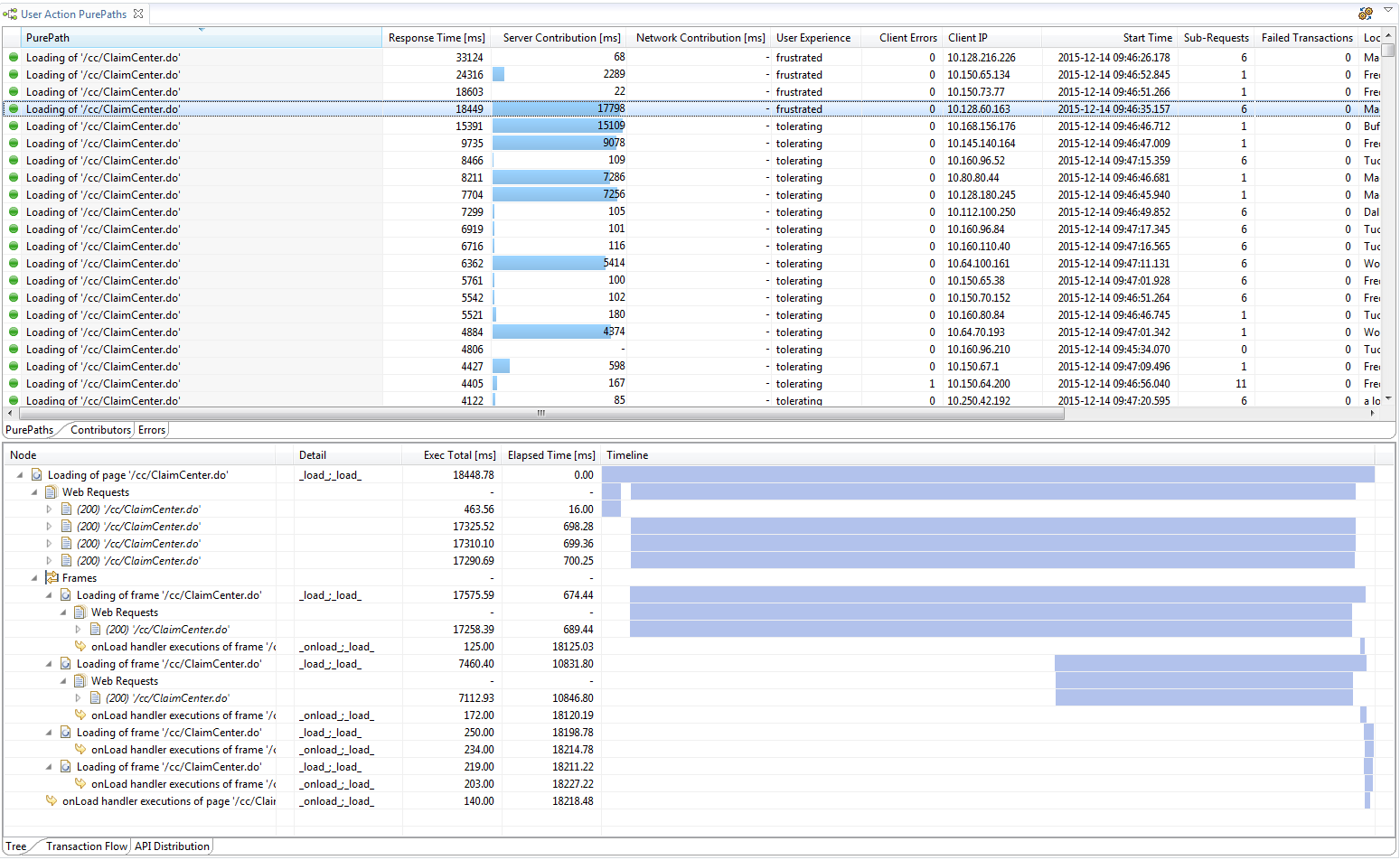
Tree: The sequential representation of all contributors in a tree.



Transaction Flow: The PurePath contributing system components visualized in a transaction flow.



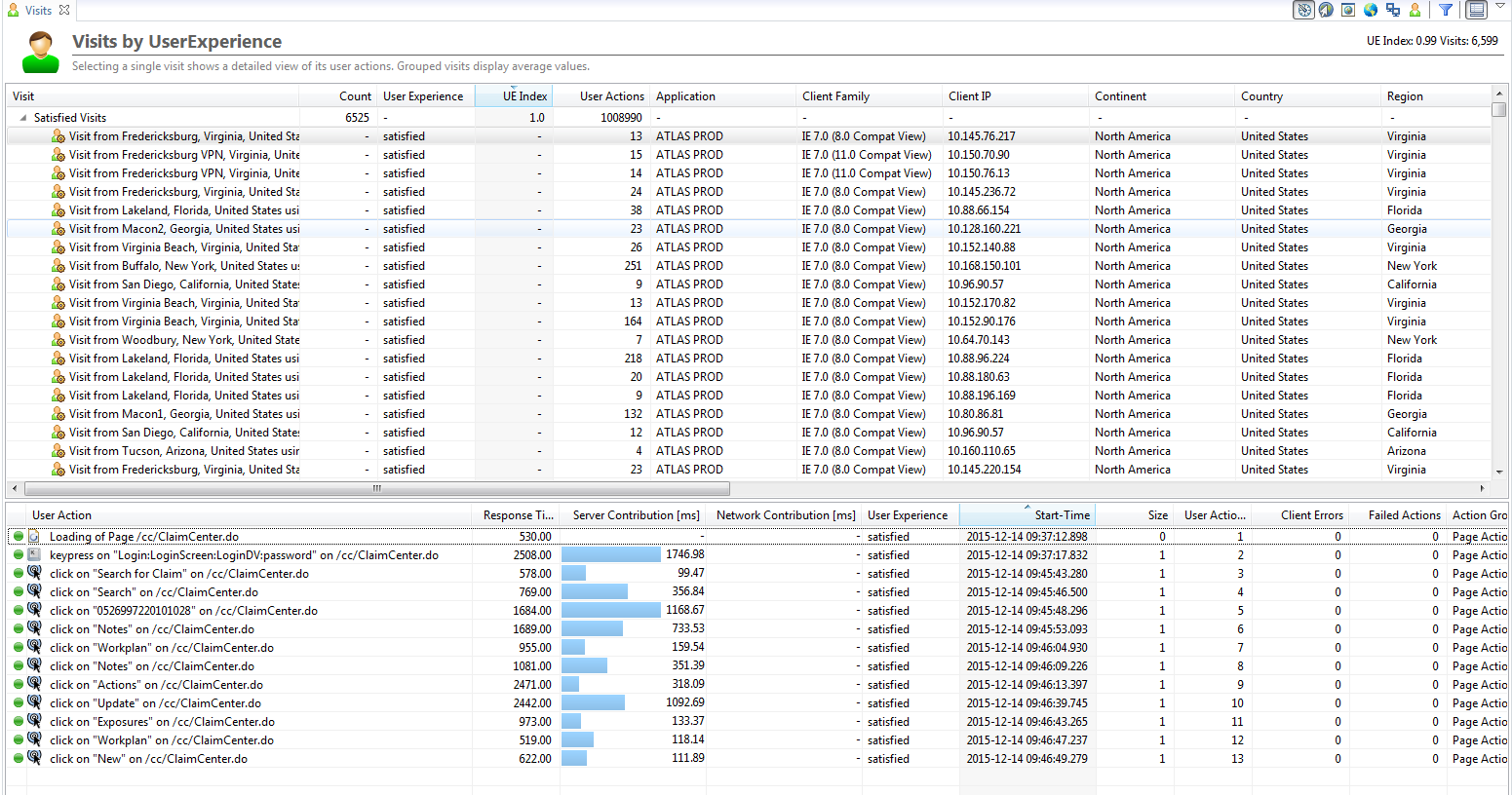
API Distribution: Execution and CPU time distribution of the PurePath's contributing APIs.



Visits

A **visit** is a group of user actions performed by the same user within a certain time period. It is created with the first user action and ends after the user is inactive for a specified period of time. This user inactivity time setting defaults to 30 minutes and you can define it on an application-level. A visit represents the user's click-path and helps to analyze problems. Visit duration is calculated from the start of the first user action to the end of the last user action.

The Visits dashlet has been added to the General Dashboards for the applications that have UEM enabled. In this case, it will be another tab as all of the other dashlets in the General Dashboard.



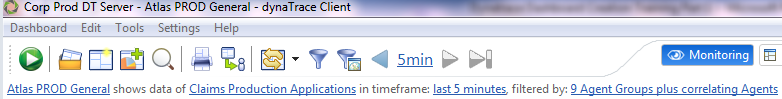
If the Visits dashlet has not yet been added, it can be opened as follows:

* In the Cockpit, select the **System Profile** for the appropriate application.
* Expand the **Diagnose User Experience** node.
* Double-click **Visits** to open the dashlet*.*

# Navigating the Data

With all of the data that comes into Dynatrace, it is essential to be able to navigate through this data to dial into the appropriate timeframe, application, user actions, and transactions. The dashboards created for each Geico application have been filtered to only show the current data for that application, however, different times and filtering may be needed.

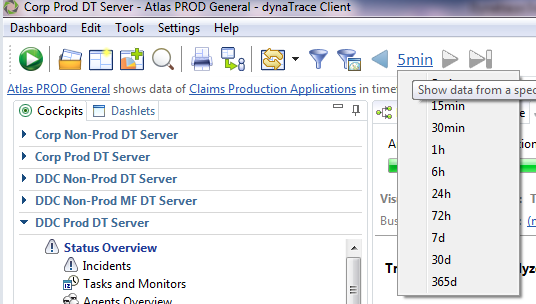
Below is an example filter that will be explained in the following sections



Timeframes

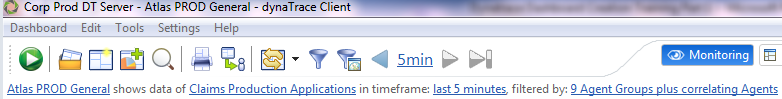
At the top of every dashboard is the large time selector and arrows.

* Clicking the blue time will allows selection of different ranges of times
* Clicking the left and right arrows allows shifting this range of times
* The far right arrow brings the timeframe back to current time



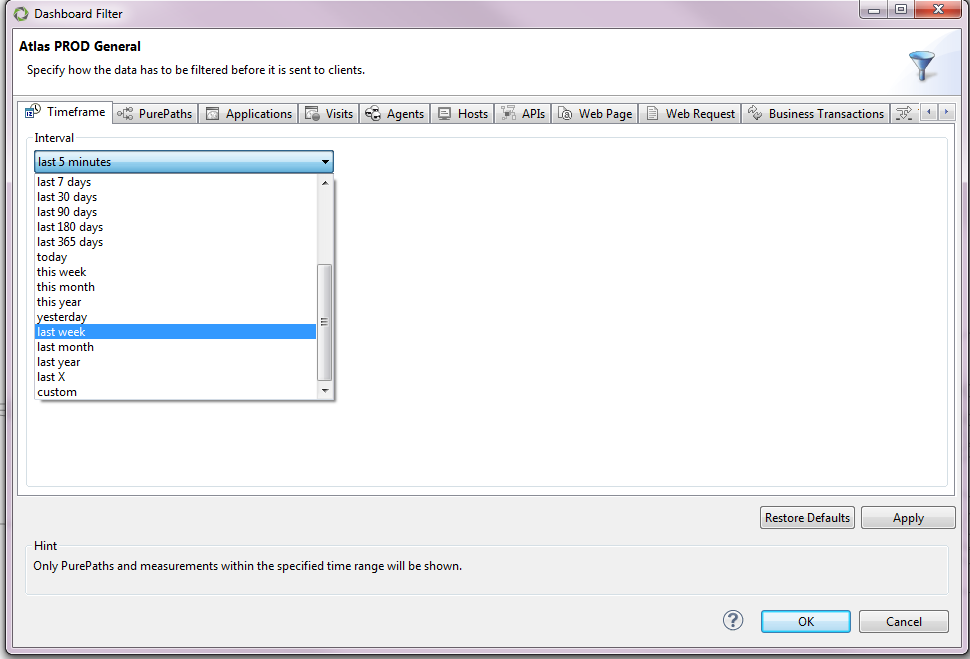
While the large time selector is easy to use for recent timeframes, going further back in data or looking at a custom time is necessary.

To enter custom times: click the blue “in timeframe: last X minutes”



In the Dashboard Filter Pop Up:

* Select Timeframe
* Choose your custom timing
* Hit apply and OK



Note: Dynatrace’s transactional data is only saved for up to 2 weeks. Metrics/Chart Data is held indefinitely and aggregated over time. If select transactional data is needed for longer, it can be saved separately via session store.

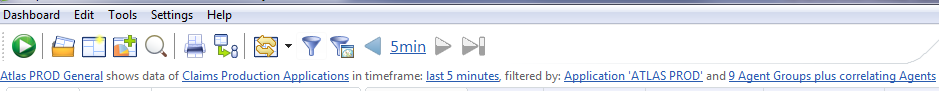
Filters

Beyond the Timeframes, there are also many filters applied to each dashboard. The most commonly utilized filters are in the below example.



* Atlas PROD General – Dashboard Name
* Claims Productions Application – System Profile of the Dynatrace Server
  + Refer to Wiki for correct System Profile and Dynatrace Server for your application
  + http://gnie.geico.net/sites/ApplIntfDiv/Monitor\_Event\_Mgmt/Dynatrace%20APM%20Overview/Home.aspx
* Last 5 Minutes – Timeframe
* Application ‘ATLAS PROD’ – Application Defined by URI patterns
* 9 Agent Groups plus correlating Agents – The agents on the JVMs/.NET Process/ISS Server that we are monitoring and any agents these transactions follow to.

There are many more filters that can be added to a dashboard which can be added through the blue funnel filter button.



Drill Down

The drill down feature is a powerful tool of Dynatrace used to isolate and view data in different ways.

This can be accomplished via right clicking on transactions, agent groups, web requests, visits etc. and selecting drill down, and selecting the new view you would like to see that data in.

In the following example, by drilling down from a Visit into the PurePaths, method level visibility is gained into the performance of the transactions.

