

# Advanced Real User Monitoring and Analytics

Hands-on Training 2020



**Matt Miller**

Sales Engineer



**Carl Morphet**

Manager, Business Insights



**Klaus Enzenhofer**

Product Manager



## Agenda

---

- Introductions & Prerequisites
- Setup Tenant & EC2 Instance
- Best Practices of Application Setup in Dynatrace
- Hands-on: Setup application
- Hands-on: Session/Action Properties
- Hands-on: Error Workflow
- Hands-on: User Metrics
- Hands-on: USQL

# Introductions

---

- Goals
  - Help you become more comfortable with advanced Dynatrace RUM capabilities
  - Drill into capabilities of data captured via Dynatrace RUM
  - Have fun
- What we will use
  - Dynatrace SaaS Tenant
  - AWS EC2 Instance
  - easyTravel Demo Application
  - Positive Attitude

## Prerequisites

---

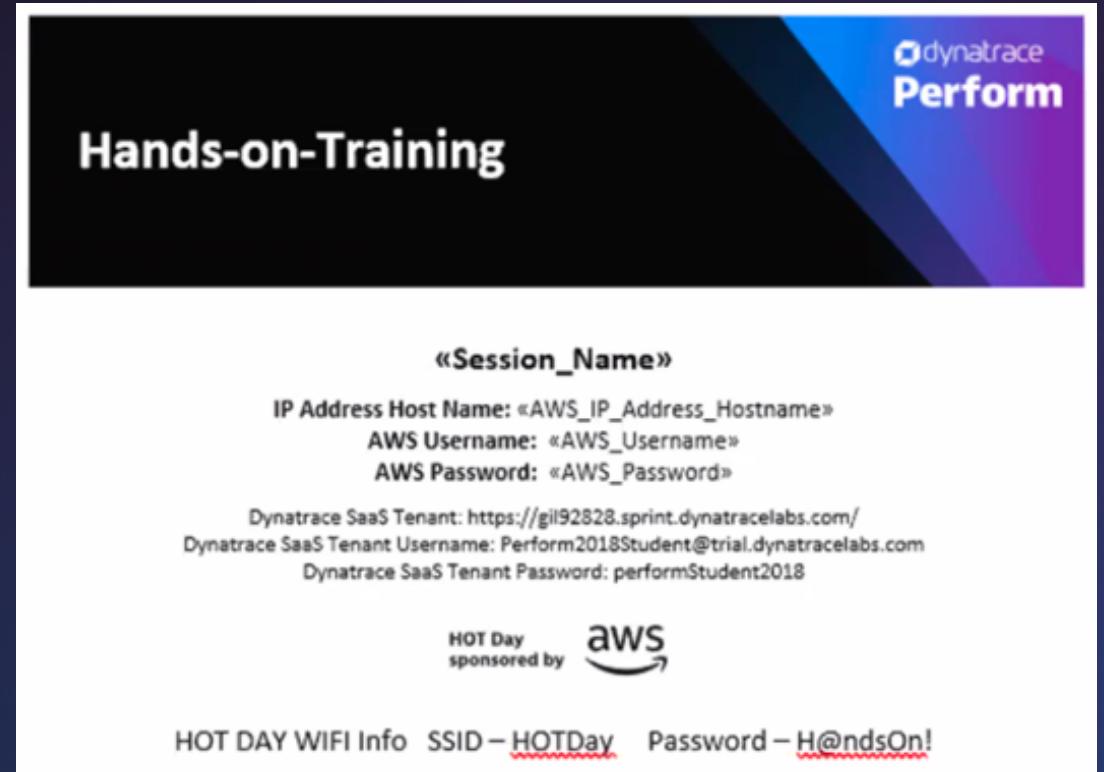
- Internet Access on laptop
- Dynatrace UI supported browser
- RDP client
- Basic Knowledge of Dynatrace Real-User Monitoring (RUM)
- Advanced knowledge of Easytravel application
- GitHub Repo: [\*\*https://github.com/Dynatrace/perform-2020-hotday\*\*](https://github.com/Dynatrace/perform-2020-hotday)

# Tenant & EC2 Instance Setup

## Tenant & EC2 Info

---

- Everyone should have received a card with:
  - Dynatrace Tenant URL and credentials
  - EC2 IP and credentials



## Lab Component Verification

---

- Open RDP client and connect to your Windows EC2 instance with provided IP
- Open EasyTravel Config UI
  - UEM -> Dynatrace Training
- Log in to Dynatrace SaaS tenant with provided URL and credentials
- Verify your EC2 host is connected and data is being captured under Applications -> My Web Application
- Disable deep instrumentation for Easytravel launcher process
  - com.dynatrace.easytravel.launcher.jar easyTravel

# Best Practices of Application Setup

# Default (Unconfigured) Session

 Desktop 1920 x 1200	 Basingstoke, United Kingdom Geolocation
 Windows 8 Operating system	 ENTANET ENTANET International ... Internet service provider
 Firefox 36 Browser	 78.32.12.0 IP address

Order	Type	User action	Conversion	Action duration	JavaScript errors	Apdex	Details
1	Load action	Loading of page /orange.jsf	-	364 ms	0	 Satisfied	
2	Load action	Loading of page /orange.jsf	-	355 ms	0	 Satisfied	
3	XHR action	http://et-test.europe-west2-c.c.smart-monitor-248307.internal:8079/orange.jsf	-	146 ms	0	 Satisfied	
4	User tag ev...	labuser	-	-	-	 Satisfied	
5	XHR action	http://et-test.europe-west2-c.c.smart-monitor-248307.internal:8079/orange.jsf	-	75 ms	0	 Satisfied	
6	XHR action	http://et-test.europe-west2-c.c.smart-monitor-248307.internal:8079/orange.jsf	-	102 ms	0	 Satisfied	
7	XHR action	http://et-test.europe-west2-c.c.smart-monitor-248307.internal:8079/orange.jsf	-	108 ms	0	 Satisfied	
8	XHR action	http://et-test.europe-west2-c.c.smart-monitor-248307.internal:8079/CalculateRecommendations	-	310 ms	0	 Satisfied	
9	Load action	Loading of page /orange-booking-review.jsf	-	239 ms	0	 Satisfied	
10	Load action	Loading of page /orange.jsf	-	697 ms	0	 Satisfied	
11	XHR action	http://et-test.europe-west2-c.c.smart-monitor-248307.internal:8079/orange.jsf	-	152 ms	0	 Satisfied	
12	XHR action	http://et-test.europe-west2-c.c.smart-monitor-248307.internal:8079/orange.jsf	-	133 ms	0	 Satisfied	
13	XHR action	http://et-test.europe-west2-c.c.smart-monitor-248307.internal:8079/orange.jsf	-	158 ms	0	 Satisfied	

# Configuration

## Session Properties ◀

Async Actions

User Action Naming

User Tag

Key User Actions

Conversion Goals

Availability Monitoring

## Session Properties

### Add property

Select the expression type that includes the metadata you want to use for this user tag rule. [More...](#)

Expression type

CSS selector

Data type

String, max 100 chars

Key

CSS selector

Specify a CSS selector. This mechanism will capture the first match's innerText/textContent value. To retrieve a specific attribute value of the element, append the '@' symbol, followed by the attribute name. i.e. '#someDomElement@someAttribute'

Display name

optional

Select at least one storage type

- Store as user action property  
 Store as session property

### CSS selector

Basket

button[name="payment:pay"]

Loyalty

statusYes

- Create a session property for basket value using a CSS selector
- Create a session property for loyalty using a JavaScript variable

### Tips

You probably already have tools, such as analytics, which are surfacing some of these values we can use

When using a CSS selector, make sure nothing in it is dynamic – ideally rely on IDs

### Regex

Pay\s\\$(.\*?)\.

# Configuration

Session Properties

Async Actions

User Action Naming

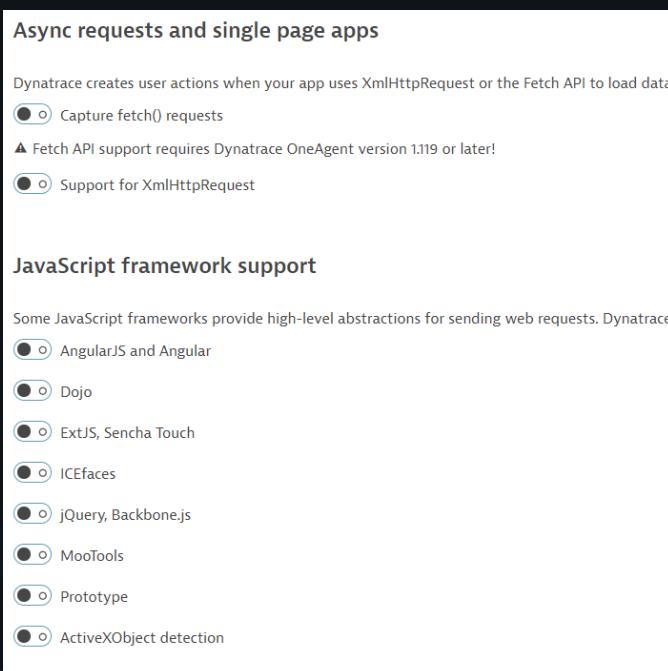
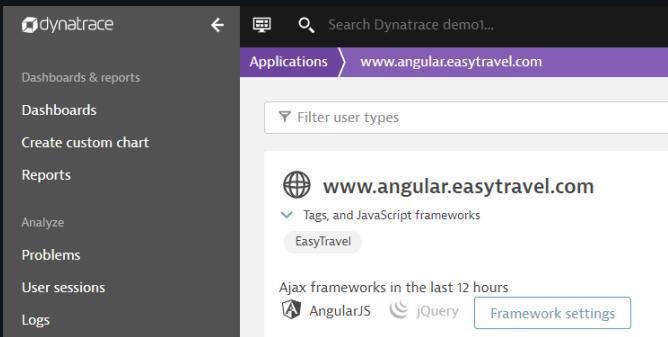
User Tag

Key User Actions

Conversion Goals

Availability Monitoring

## Async Actions



Async requests and single page apps

Dynatrace creates user actions when your app uses XMLHttpRequest or the Fetch API to load data

Capture fetch() requests

⚠ Fetch API support requires Dynatrace OneAgent version 1.119 or later!

Support for XMLHttpRequest

JavaScript framework support

Some JavaScript frameworks provide high-level abstractions for sending web requests. Dynatrace

AngularJS and Angular

Dojo

ExtJS, Sencha Touch

ICEfaces

jQuery, Backbone.js

MooTools

Prototype

ActiveXObject detection

- Find out which frameworks need to be enabled
- Enable the right frameworks in the settings

### Tips

The detected frameworks depends on traffic being put through

Tools like BuiltWith can give you an idea if you're not sure

# Configuration

Session Properties

Async Actions

User Action Naming ◀

User Tag

Key User Actions

Conversion Goals

Availability Monitoring

## User Action Naming

Add naming rule for load actions

Specify a naming pattern and conditions when this pattern should apply. You can use your defined placeholders for the naming pattern. [More...](#)

Naming pattern

{userInteraction} on {elementIdentifier} - {pageUrl}

Use [ ] to select placeholders.

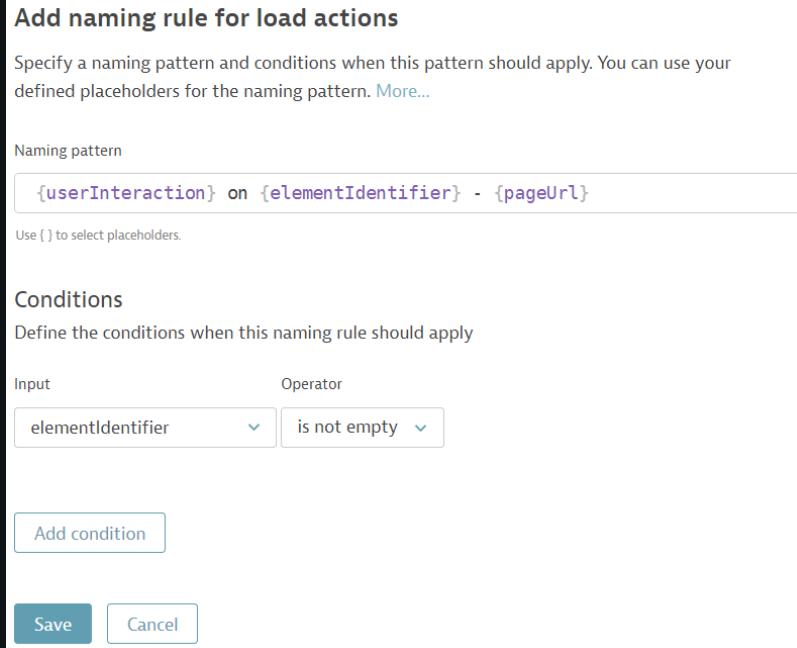
Conditions

Define the conditions when this naming rule should apply

Input	Operator
elementIdentifier	is not empty

Add condition

Save Cancel



- Create a naming rule for load actions
- Create a naming rule for XHR actions

### Tips

There are placeholders available such as “userInteraction” (click, keypress etc.) and “elementIdentifier” (name of the button pressed)

Use conditions to avoid a situation where an action name could be blank!

# Configuration

Session Properties

Async Actions

User Action Naming

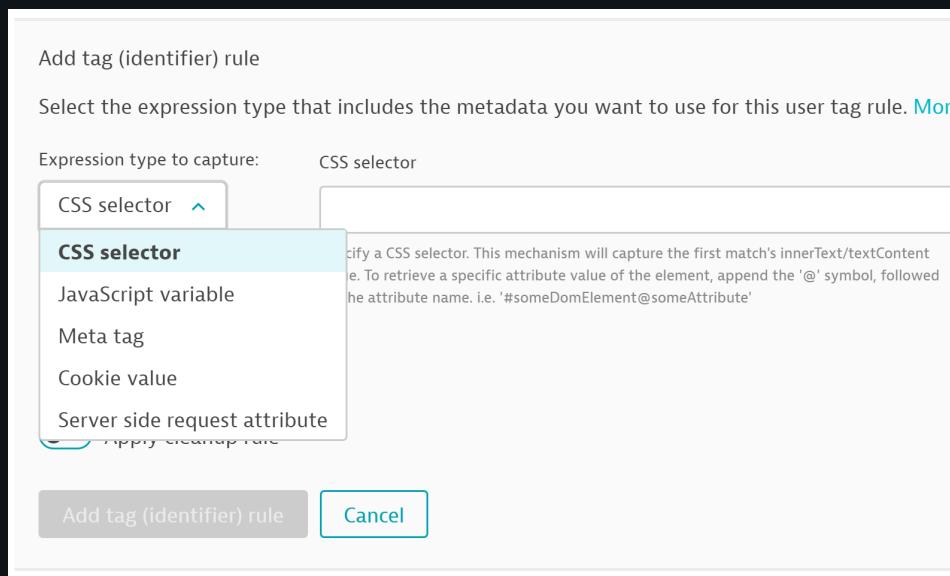
User Tag

Key User Actions

Conversion Goals

Availability Monitoring

## User Tagging



- Create a user tag based upon a CSS selector

### Tips

You can use Chrome developer tools as a quick way to get CSS selectors

Check with your security teams what sort of ID they are happy to capture, it doesn't have to be an email address it can be a session ID

CSS selector

.greeting

Regex

Hi, (.\*)+

# Configuration

Session Properties

Async Actions

User Action Naming

User Tag

Key User Actions

Conversion Goals

Availability Monitoring

## Key User Actions

The screenshot shows a Dynatrace dashboard for the application 'easytravel-angular-demo1'. It displays a journey named 'Loading of page /blog/'. A button labeled 'Mark as key user action' is visible. Below the journey, there are performance metrics: 10.5 s Visually complete, 1.1 /min User actions, 0.0 /min JavaScript errors, and a 0.39 (Unacceptable) Apdex rating. There are also resource counts: 38.5 Resources /action and 9.4 CDN and 3rd party resources.

The screenshot shows a Dynatrace dashboard for the application 'www.angular.easytravel.com'. It displays a list of 'Key user actions' with the following data:

Type	User action	Domain	Total time consumption	Duration	Actions	JavaScript errors	Apdex
Link	/easytravel/rest/validate-creditcard	easytravel-angular-demo1.internal.dynatrace.com	~	Visually complete: 2.70 s	2.58 /min	0.18 /min	Fair
Link	Loading of page /easytravel/search	easytravel-angular-demo1.internal.dynatrace.com	~	Visually complete: 1.26 s	2.40 /min	0.00 /min	Good
Link	/easytravel/rest/journal	easytravel-angular-demo1.internal.dynatrace.com	~	Visually complete: 0.23 s	9.33 /min	0.00 /min	Excellent
Link	/easytravel/rest/login	easytravel-angular-demo1.internal.dynatrace.com	~	Visually complete: 0.19 s	3.11 /min	0.00 /min	Excellent
Link	Loading of page /easytravel/signup	easytravel-angular-demo1.internal.dynatrace.com	~	Visually complete: 0.52 s	0.13 /min	0.00 /min	Fair
Link	Finish	easytravel-angular-demo1.internal.dynatrace.com	~	Visually complete: -	-	-	-

- Mark some of the key actions in the journey as key user actions such as “search” and “login”

### Tips

These can be changed/added at anytime and won't interfere with what's being captured

# Configuration

---

Session Properties

Async Actions

User Action Naming

User Tag

Key User Actions

Conversion Goals ◀

Availability Monitoring

## Conversion Goals

---

Conversion goals

Conversion goals are a versatile way of measuring how well your application fulfills your company's business objectives. You can set up separate goals related to key pages (for example, a shopping cart "add item" page or a newsletter signup page), session length, or number of actions per session. You can then compare the conversion rates of various user actions against each of your conversion goals.

Name:

Type of goal:

Type URL or path

Case sensitive

- Create a conversion goal based on a user action/destination
- Create a conversion goal based on user behavior

### Tips

Check the RUM data to see if your “goal” is a button click or a page load

Conversions don't have to be a “purchase,” it can be the completion of any workflow

# Configuration

---

Session Properties

Async Actions

User Action Naming

User Tag

Key User Actions

Conversion Goals

Availability Monitoring ◀

## Availability Monitoring ADD SCREENSHOTS

---

**Conversion goals**

Conversion goals are a versatile way of measuring how well your application fulfills your company's business objectives. You can set up separate goals related to key pages (for example, a shopping cart "add item" page or a newsletter signup page), session length, or number of actions per session. You can then compare the conversion rates of various user actions against each of your conversion goals.

Name:

Type of goal:

Type URL or path

Case sensitive

- Create a conversion goal based on a user action/destination
- Create a conversion goal based on user behavior

### Tips

Check the RUM data to see if your “goal” is a button click or a page load

Conversions don't have to be a “purchase,” it can be the completion of any workflow

# Fully Configured Session

basket: 1,898.0      loyalty: bronze

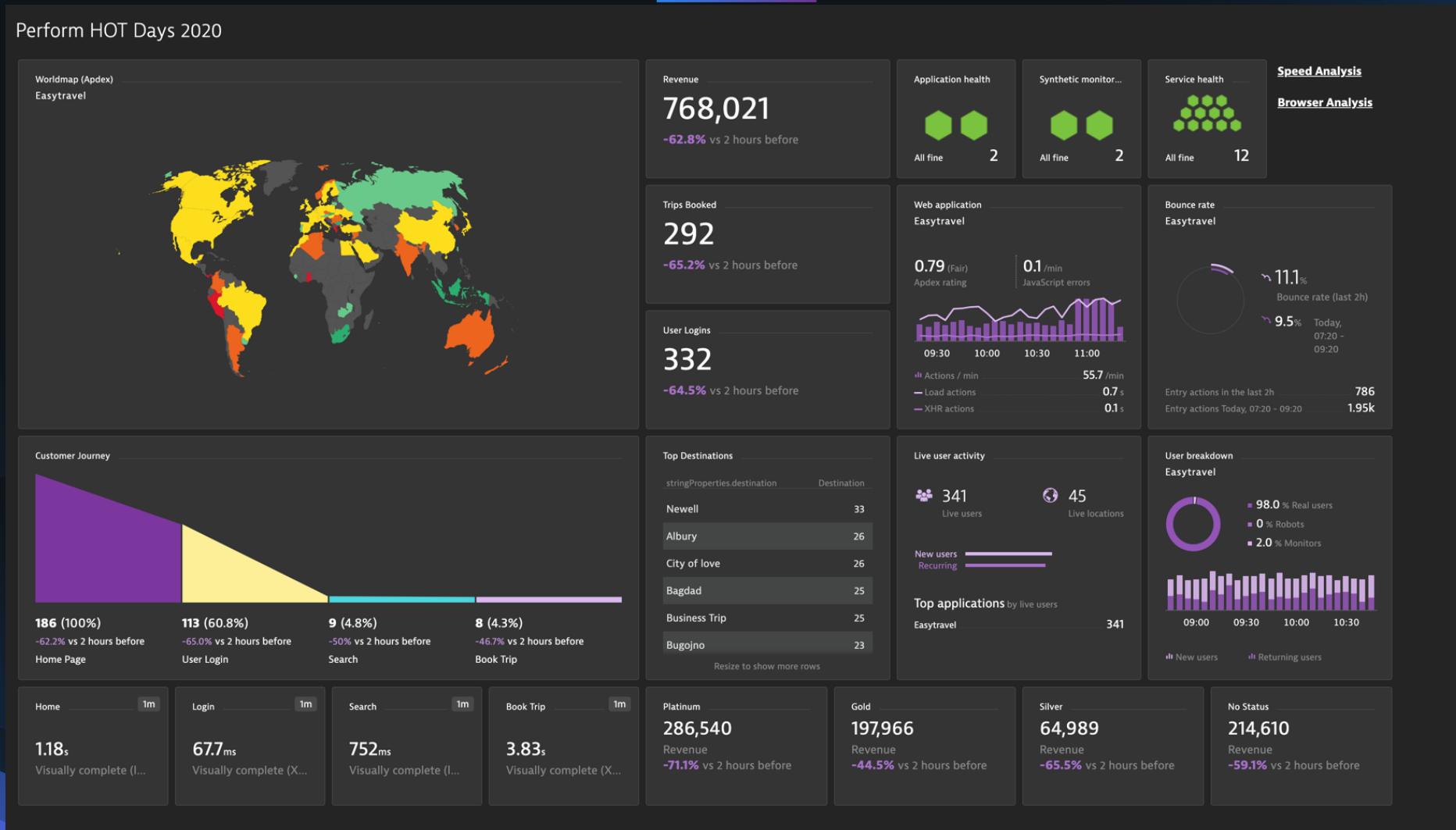
 Desktop 2048 x 1152	 Maidenhead, United Kingdom Geolocation	 maria User tag
 Windows 10 Operating system	 BT-UK-AS BTnet UK Regional net... Internet service provider	 Satisfied User Experience
 Chrome 77 Browser	 86.188.144.0 IP address	

Order	Type	User action	Conversion	Action duration	JavaScript errors	Apxd	Details
1	Load action	Loading of page /		4.89 s	0	 Tolerat...	
2	Load action	loading of page /		3.59 s	0	 Satisfied	
3	XHR action	click on "search" - /calculaterecommendations		172 ms	0	 Satisfied	
4	Load action	click on "book now" - http://34.89.12.243:8080/orange-booking-review.jsf?journeyid=8267		2.2 s	0	 Satisfied	
5	XHR action	click on "login" - /orange-booking-review.jsf		82 ms	0	 Satisfied	
6	User tag ev...	maria		-	-	 Satisfied	
7	Load action	click on "next" - http://34.89.12.243:8080/orange-booking-payment.jsf?journeyid=8267		2.38 s	0	 Satisfied	
8	XHR action	click on "iceform:j_idt118" - /orange-booking-payment.jsf		461 ms	0	 Satisfied	
9	Load action	click on "next" - http://34.89.12.243:8080/orange-booking-finish.jsf?journeyid=8267		2.99 s	0	 Satisfied	
10	Load action	click on "finish" - http://34.89.12.243:8080/orange-booking-finish.jsf?success=1&journeyid=8267		2.52 s	0	 Satisfied	

## BEFORE Homescreen

---

# AFTER Home Screen



# Hands-on: Application Setup

## Application Setup

---

- Frameworks and asynchronous actions
- User action naming
- User tag
- Key user actions
- Conversion goals
- Availability monitoring
- (Session/Action properties)

## Framework Detection and Asynchronous Actions

The screenshot shows the Dynatrace application monitoring interface for the 'Easytravel' application. At the top, there's a navigation bar with icons for Overview, Application, Infrastructure, and Data. Below the navigation, the application name 'Easytravel' is displayed with a globe icon. A dropdown menu for 'Tags, and JavaScript frameworks' is open, showing 'AngularJS', 'Icefaces', 'jQuery', and 'Prototype'. There's also a button to '+ Add tag'. A blue rectangular box highlights the 'Framework settings' button. Below this, a section titled 'Ajax frameworks in the last 12 hours' lists the same four frameworks. Underneath, a section titled 'Traffic for this application is detected by the following rules:' lists two domain rules. At the bottom, a large button labeled 'Performance analysis' is visible.

Easytravel

Tags, and JavaScript frameworks

+ Add tag

Ajax frameworks in the last 12 hours

AngularJS Icefaces jQuery Prototype

Framework settings

Traffic for this application is detected by the following rules:

- domain contains 100.20.68.100
- domain contains 10.0.0.16

Performance analysis

# User Action Naming

User action naming rules

1 / 250 user defined naming rules

Split user actions by domain [More...](#)

Case insensitive naming [More...](#)

Naming rules for load actions Naming rules for XHR actions

First matching rule is applied. Use the priority column to manage rule matching behaviour.

[Add naming rule](#)

Template	Move up/down	Delete	Edit
{userInteraction} "{elementIdentifier}" {pageUrlPath}	▲ ▼	✖	✎
{xhrUrl}	▲ ▼	✖	✎

# User Tag

## User tag

Set up rules to automatically tag the users of this application with metadata pulled from cookies, JavaScript variables, or other elements to gain deeper insight into the individual journeys of each of your customers. Tags for individual users – which may be comprised of user names, user IDs, or other metadata captured via a browser or CSS selector – can dramatically enrich the user experience insights you gain from [User session analysis](#).

Alternatively, you can use the JavaScript API call `dtrum.identifyUser()` to tag each user session with user-specific metadata such as a username or JavaScript variable. For details, see [JavaScript API](#).

**Note:** When multiple tags match a single user, only the first tag is used. Subsequent tags are ignored. Sort your tags to increase the order.

**Adaptive traffic management**  might be applied to protect your network bandwidth. As a result, not all server-side request attributes will be captured during periods of peak load. Any user tags that you define based on missing server-side request attributes also won't be captured.

[Add tag \(identifier\) rule](#)

Expression type name	Expression type	Move up/down	Delete	Edit
.greeting	CSS selector	 	 	
#loginForm\:j_idt39	CSS selector	 	 	

# Key User Actions

Key user actions

To define additional user actions as key user actions, select an action from the Top 100 user actions list below. [More...](#)

Show active key user actions only

▼ Start typing to filter user action and domain names...

Type	User action	Total time consumption	Duration
⌚	loading of page /easytravel/search	<div style="width: 80%;"></div>	Visually complete: 0.66 s
⌚	loading of page /easytravel/home	<div style="width: 60%;"></div>	Visually complete: 1.14 s
🖱️	click "search" /easytravel/search	<div style="width: 10%;"></div>	Visually complete: 0.03 s
🖱️	/easytravel/rest/login	<div style="width: 0%;"></div>	Visually complete: -

## Conversion Goals

---

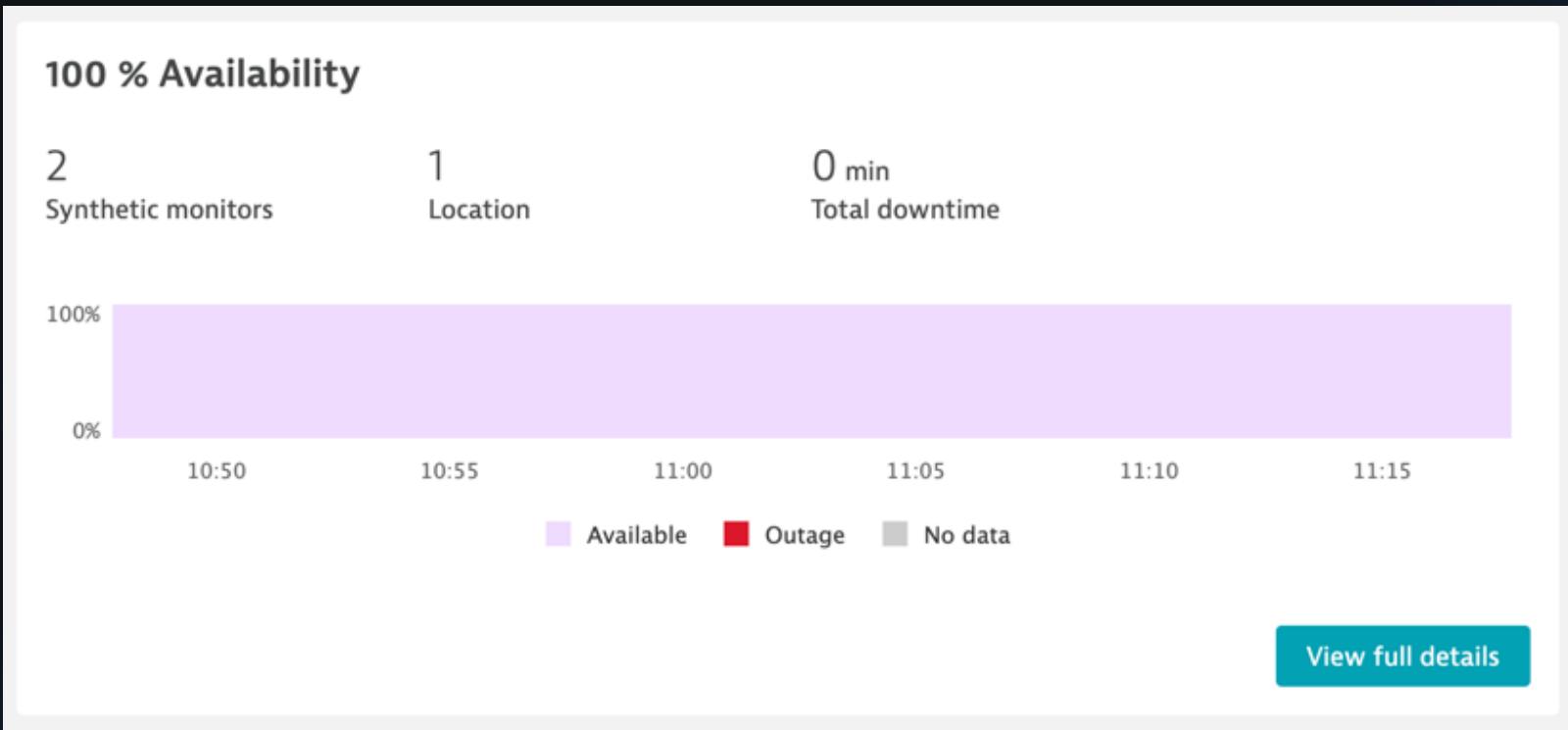
### Conversion goals

Conversion goals are a versatile way of measuring how well your application fulfills your business needs. You can define goals based on session length, or number of actions per session. You can then compare the conversion rates across different user segments.

[Add goal](#)

**Book** : Action name contains **book**

## Availability Monitoring



# Hands-on: Session/Action Properties

# Session Properties

## Session and user action properties

Session properties can be used to enrich monitored user sessions and actions with additional details that facilitate fine-grain user-session and application-performance analysis.

You might create key/value pairs for

- purchases above a certain amount (for example, `ordersum = 100`),
- datacenter location (for example, `datacenter = south1`), or for
- a segment of users receiving a new promotion (for example, `abtesting = c`)

Once defined, such properties can be used for filtering your monitoring results across many Dynatrace analysis views. Such defined session properties are also included with user sessions upon export. The number of properties is limited to 20 properties per session. If an application spans multiple applications, the first 20 properties are captured.

[Adaptive traffic management](#) might be applied to protect your network bandwidth. As a result, not all server-side request attributes will be captured during periods of peak load. Any properties that you define based on missing server-side request attributes also won't be captured.

[Show less...](#)

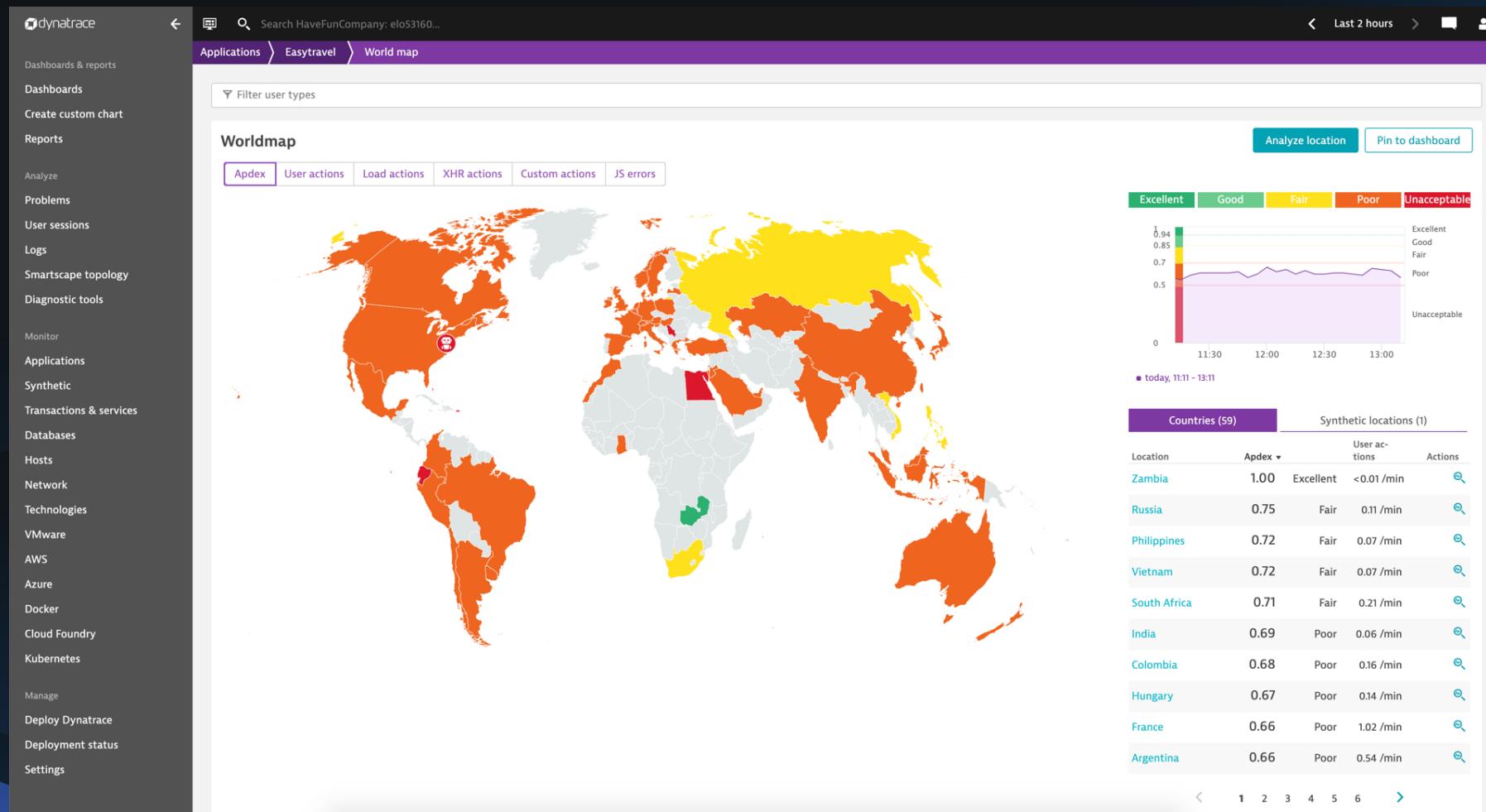
[Add property](#)

▼ Filter for properties...

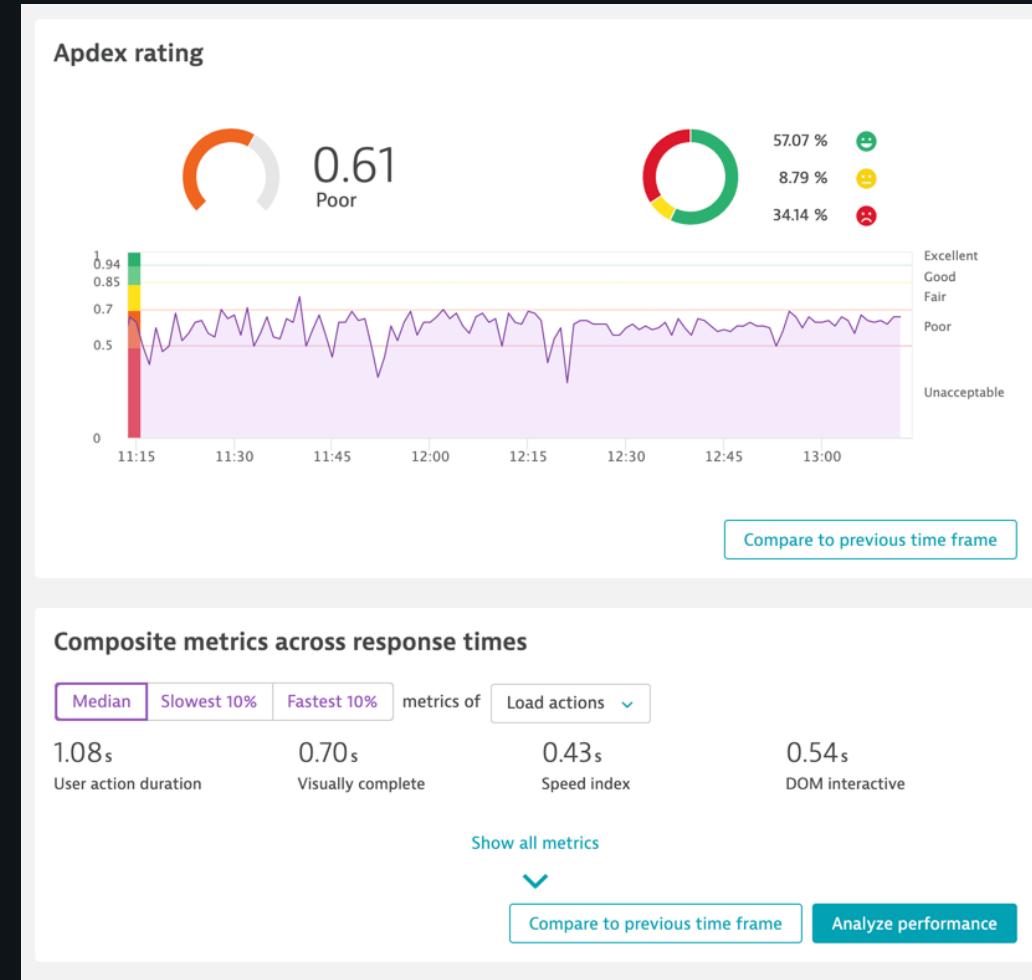
Key ▲	Data type ▲	Expression type ▲	Capture expression ▲	Property ▲	Delete	Details
destination	String	CSS selector	body > app-root > app-book-journey > app-main-layout > div > div > main > div.steps > div.step.step2.active > h2	Session		
revenue	Double	Server side request attribute	Revenue	Session		
status	String	Server side request attribute	Loyalty	Session		

# Hands-on: Error Workflow

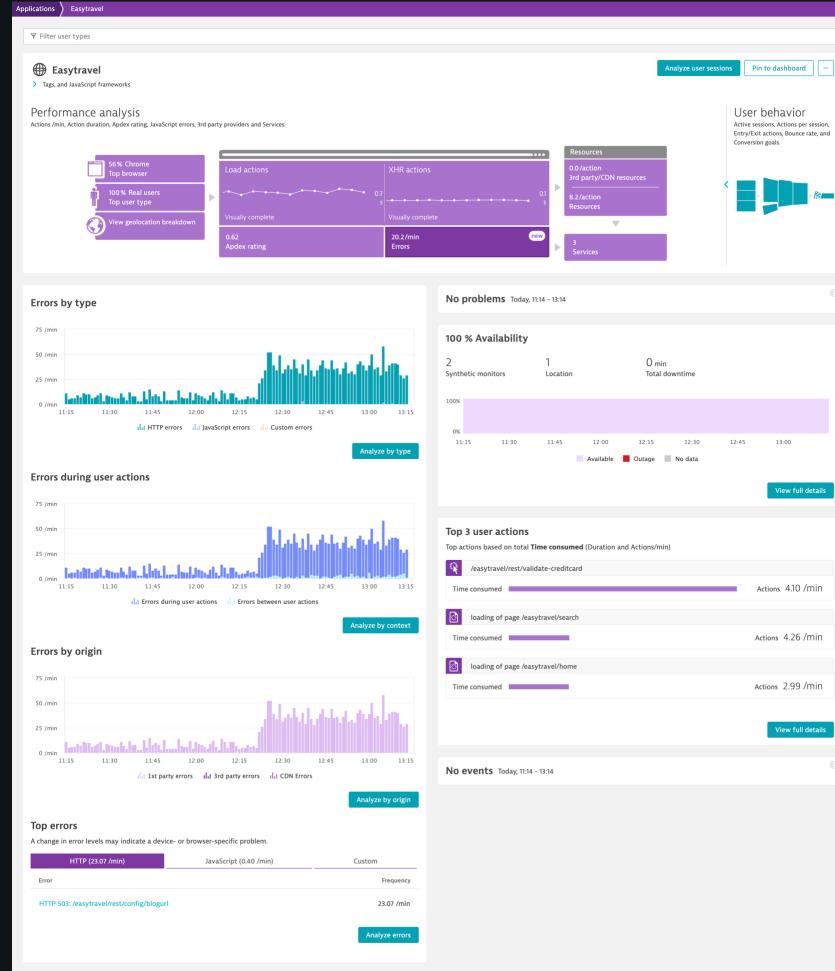
# Application Performance



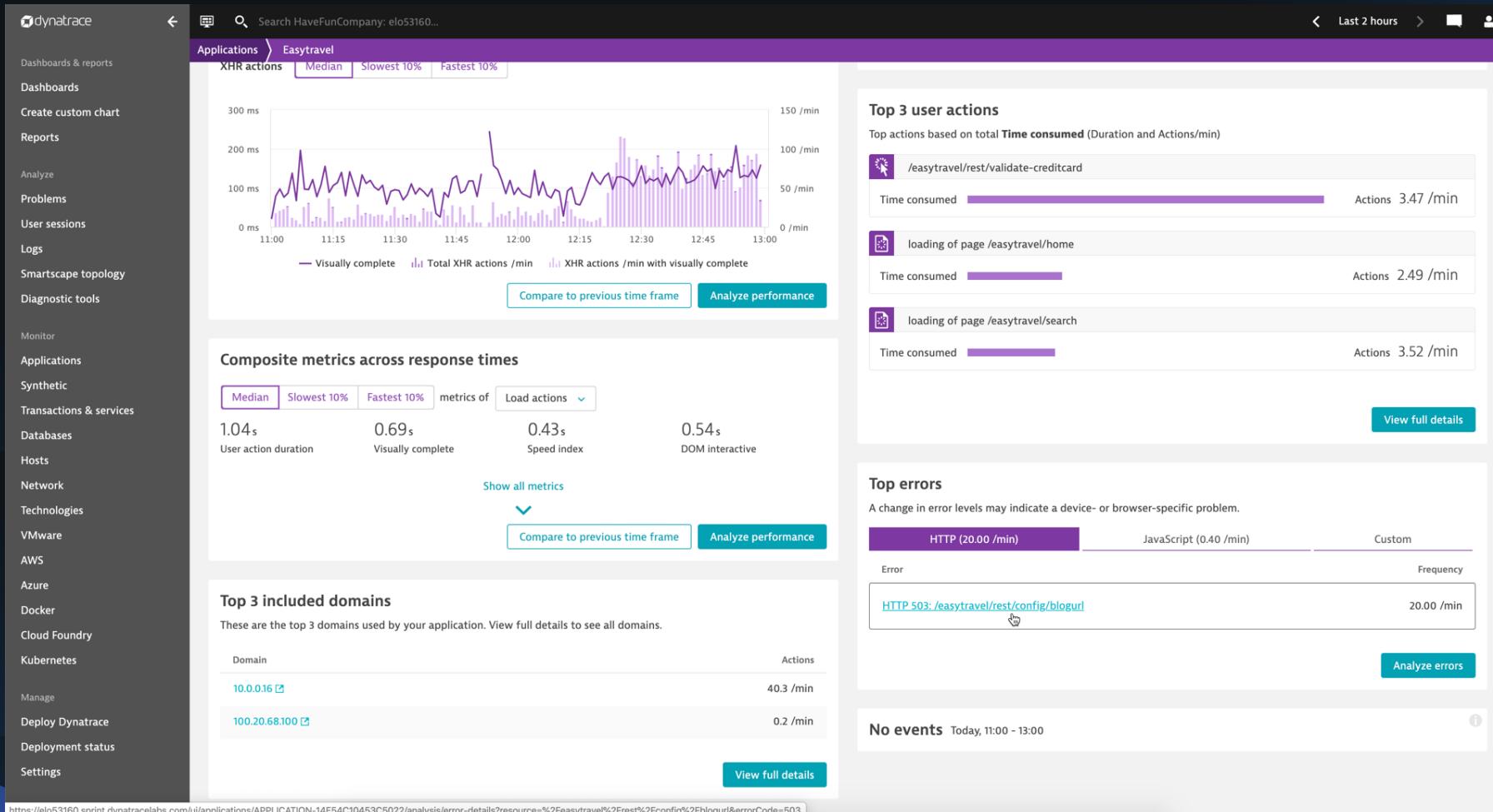
# Application Performance



# Error Workflow



# Individual Error Analysis



# Individual Error Analysis

The screenshot shows the Dynatrace Individual Error Analysis interface for an 'HTTP 503: /easytravel/rest/config/blogurl' error. The left sidebar contains navigation links for Dashboards & reports, Dashboards, Create custom chart, Reports, Analyze, Problems, User sessions, Logs, Smartscape topology, Diagnostic tools, Monitor, Applications, Synthetic, Transactions & services, Databases, Hosts, Network, Technologies, VMware, AWS, Azure, Docker, Cloud Foundry, Kubernetes, Manage, Deploy Dynatrace, Deployment status, and Settings.

The main content area displays the error details:

- HTTP 503: /easytravel/rest/config/blogurl**
- Create detection rule based on error pattern** (button highlighted with a blue box)
- When this error occurs...**
  - 95.8 % occurred during user action
  - 100 % 1st party
  - 52 % Windows 10
  - Multiple browsers
  - Multiple locations
- Error details**

HTTP Error Code	503
URL	10.0.0.16:8079/easytravel/rest/config/blogurl http://10.0.0.16:8079/easytravel/rest/config/blogurl 100.20.68.100:8079/easytravel/rest/config/blogurl
Error count	1896
Impacted users	343
Errors detected via	RUM JavaScript tag, OneAgent
- Error-detection rule HTTP 500-599 \***
  - Capture error Yes
  - Include in Apdex calculations Yes
  - Include in Davis AI causation analysis Yes
- Edit detection rules for this error pattern**
- Create detection rule based on this error pattern**
- User sessions affected by this error**

Type	User session id	Error on user action	Error origin	Browser	Operating system	Location
UBXWLR	Yes	1st party	IE 11	Windows 8.1	Germany	
DELXQC	Yes	1st party	Firefox 52	Windows 10	United States	
JMSEYZ	Yes	1st party	Chrome 55	Windows 10	United States	
LASBGF	Yes	1st party	Chrome 55	Windows 10	Seattle	

# Individual Error Analysis

The screenshot shows the Dynatrace web interface with the following navigation path: Applications > Easytravel > Settings > Errors > Add HTTP error rule.

The main content area is titled "Application settings" for the "Easytravel" application. It displays various configuration sections:

- General**: See your application setup.
- Data privacy**: Adjust data-capture & storage settings.
- User actions**: Generate consistent user action names.
- Conversion goals**: Measure success against business goals.
- User tag**: Add user tags to identify your session.
- Session and user action properties**: Use metadata to analyze user sessions.
- Metrics**: Manage and create metrics.
- Async requests and single page apps**: Select your frameworks.
- Content capture**: Define resource monitoring.
- Session Replay**: Configure session replay settings.
- Usability analytics**: Configure usability analytics settings.
- Anomaly detection**: Fine-tune the sensitivity and thresholds.
- Errors**: Configure how errors are captured.
- Advanced setup**: (partially visible)

The "Errors" section is currently active, showing the "Add HTTP error rule" configuration screen. The configuration includes:

- Add HTTP error rule**: Enable at least one matching option.
  - Match by error code
  - Match by errors that have unknown status codes
- Error code**: 503 (Use distinct error codes or specify ranges)
- Filter by URL**:
  - equals /easytravel/rest/config/blogurl
- Capture settings**:
  - Capture this error type
  - Include error type in Apdex calculations
  - Include error type in Davis AI problem-detection and analysis

At the bottom right of the configuration screen, there are buttons: "You have unsaved changes.", "Discard changes", and "Save changes".

Be sure to use “contains” for URL filter

# Hands-on/Demo: User Metrics

## User Metrics

---

**Metrics**

Type to filter by metric name

Metric name ▾

- Visually by Geolocation
- HTTP Errors
- Home TTFB
- Search TTFB
- VC - Browser
- DomInteractive - Browser
- BrowserTTFB

# Hands-on: User Session Query Language

# Usersession

 **User sessions query**  
Create advanced queries on completed user sessions using user sessions query language. [View Documentation](#)

```
SELECT * FROM usersession
```

**User sessions query results** 

Information (1) 

The number of results was limited to 50. To view more results, click **Load more**.

Visualization: [Bar chart](#) [Line chart](#) [Pie chart](#) [Single value](#) **Table only**

Session details	appVersion	applicationType	bounce	browserFamily	browserMajorVersion	browserMonitorId	browserMonitorName
 749		CUSTOM_APPLICATION	false	null	null	null	null
 null		WEB_APPLICATION	false	Synthetic monitor	Synthetic monitor	SYNTHETIC_TEST-0000000...	easyTravel demo booking
 749		CUSTOM_APPLICATION	false	null	null	null	null
 740		CUSTOM_APPLICATION	false	null	null	null	null
 2.38		CUSTOM_APPLICATION	false	null	null	null	null
 2.36		CUSTOM_APPLICATION	false	null	null	null	null
 2.37		CUSTOM_APPLICATION	false	null	null	null	null

**SELECT \* FROM usersession**

# Useraction

 **User sessions query**  
Create advanced queries on completed user sessions using user sessions query language. [View Documentation](#)

```
SELECT * FROM useraction|
```

**Run query**

**User sessions query results** 

 [Information \(1\)](#)

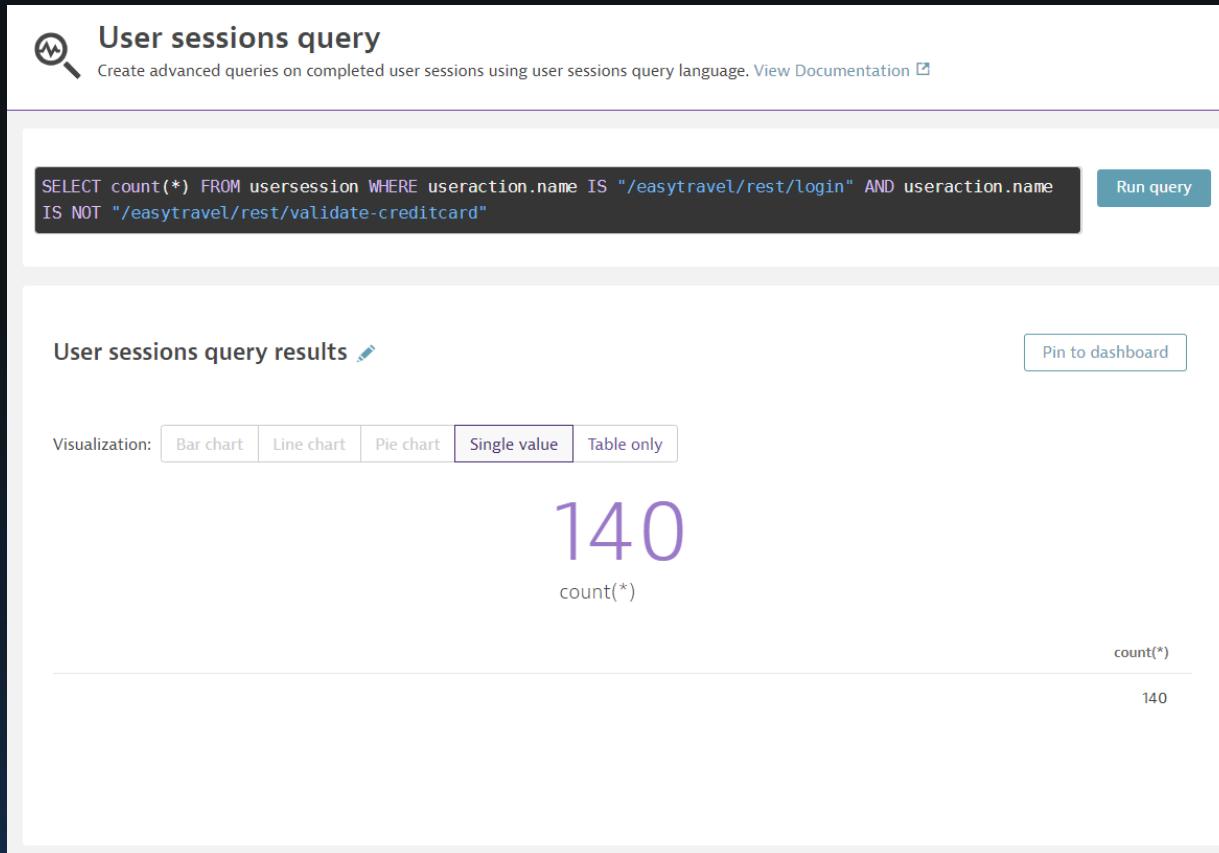
The number of results was limited to 50. To view more results, click [Load more](#).

Visualization: [Bar chart](#) [Line chart](#) [Pie chart](#) [Single value](#) **Table only**

apdexCategory	application	cdnBusyTime	cdnResources	documentInteractiveTime	domCompleteTime	domContentLoadedTime	domain	duration
SATISFIED	Citrix_RUM	null	null	null	null	null	null	0 ms
SATISFIED	Citrix_RUM	null	null	null	null	null	null	0 ms
SATISFIED	SAP_RUM	null	null	null	null	null	null	397 ms
SATISFIED	SAP_RUM	null	null	null	null	null	null	132 ms
SATISFIED	SAP_RUM	null	null	null	null	null	null	20 ms
SATISFIED	SAP_RUM	null	null	null	null	null	null	2,905 ms
SATISFIED	SAP_RUM	null	null	null	null	null	null	247 ms

**SELECT \* FROM useraction**

## Users who did X and NOT Y



### Use cases:

- eCommerce users that added to basket but did not convert
- Internal users that started a process but did not complete
- Customers that log into their account but take no further action

```
SELECT count(*) FROM usersession WHERE useraction.name IS "click on ""sign in"" - /easytravel/rest/login" AND useraction.matchingConversionGoals IS NOT "Booking confirmation"
```

## Users inside/outside SLA

User sessions query

Create advanced queries on completed user sessions using user sessions query language. [View Documentation](#)

```
SELECT count(usersession.userId) FROM useraction WHERE name IS "/easytravel/rest/validate-creditcard"  
AND visuallyCompleteTime > 5000
```

Run query

User sessions query results

Pin to dashboard

Visualization: Bar chart | Line chart | Pie chart | **Single value** | Table only

101

count(usersession.userId)

count(usersession.userId)
101

### Use cases:

- Quantify user impact to justify optimizations
- Provide a (business) customer facing SLA report

```
SELECT count(*) FROM useraction WHERE name IS "loading of page /easytravel/home" AND  
visuallyCompleteTime > 2000
```

## Users that exit on a certain page

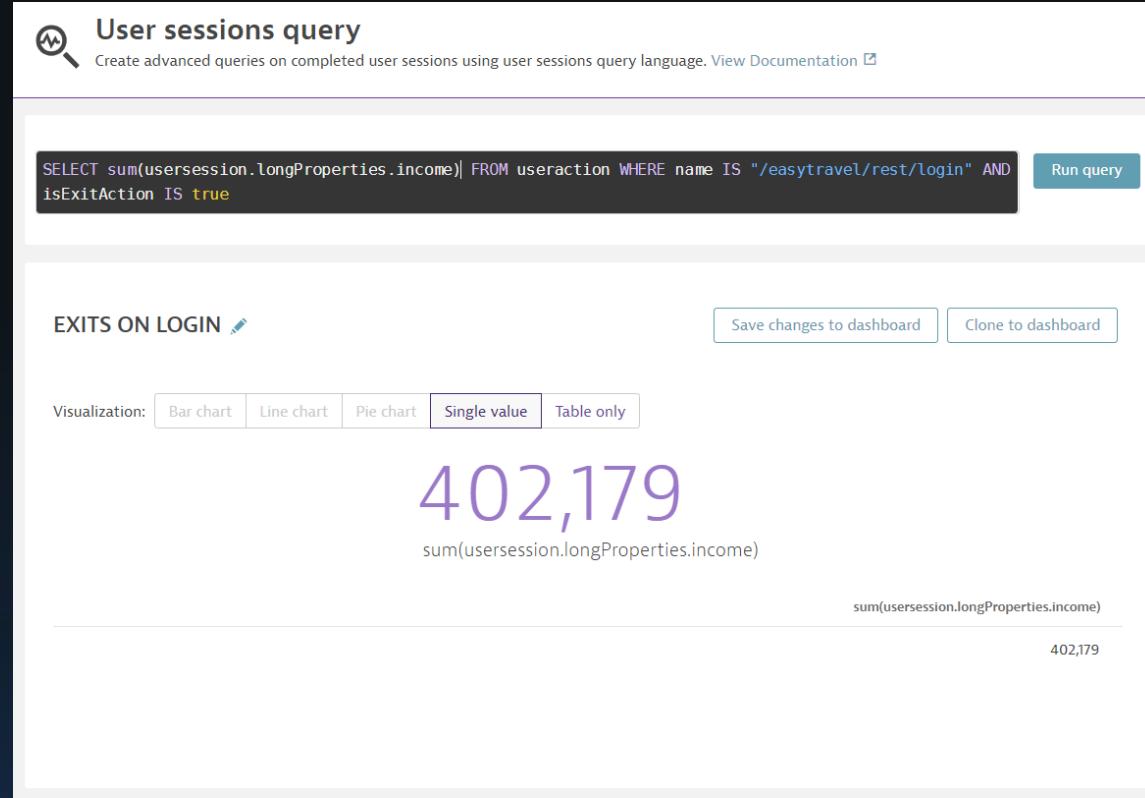
The screenshot shows the 'User sessions query' interface. At the top, there's a search icon and the title 'User sessions query'. Below it, a sub-instruction says 'Create advanced queries on completed user sessions using user sessions query language.' with a 'View Documentation' link. A code editor contains the query: 'SELECT count(\*) FROM useraction WHERE name IS "/easytravel/rest/login" AND isExitAction IS true'. To the right of the editor is a 'Run query' button. Below the editor, the text 'EXITS ON LOGIN' is displayed next to a pencil icon. On the right, there are two buttons: 'Save changes to dashboard' and 'Clone to dashboard'. Underneath, a 'Visualization' section offers five options: Bar chart, Line chart, Pie chart, Single value (which is selected and highlighted in blue), and Table only. The main result area shows a large number '536' in purple, labeled 'count(\*)'. Below it is a smaller table with one row, also labeled 'count(\*)' and containing the value '536'.

### Use cases:

- Identify where users are dropping out from a process
- One step further, look at how many are leaving because of bad performance/errors

```
SELECT count(*) FROM useraction WHERE name  
STARTSWITH "click on ""sign" AND isExitAction IS true
```

## Business value of users that exit



### Use cases:

- Estimate the business cost of poor performance
- Which “type” (loyalty status, product etc.) of user is likely to leave

```
SELECT sum(usersession.longProperties.basket) FROM useraction WHERE name  
STARTSWITH "click on ""sign" AND isExitAction IS true
```

## List of “high priority” users to investigate

Extract key information to provide  
a quick overview of that user.  
Why not add in basket value?

Choose your poor performance  
indicators, maybe they are SLAs?  
Maybe add in Rage Clicks?

```
SELECT userId AS "USERNAME", city, totalErrorCount FROM usersession WHERE useraction.errorCount > 5 OR useraction.duration > 10000 AND  
userId IS NOT NULL AND city IS NOT NULL ORDER BY totalErrorCount DESC
```

Make the list more actionable, filter out  
users with blank information, maybe  
only choose users with session replay  
data? Maybe pick users with over a  
certain amount in their basket?

# List of “high priority” users to investigate

 **User sessions query**  
Create advanced queries on completed user sessions using user sessions query language. [View Documentation](#)

```
SELECT userId AS "USERNAME", city, totalErrorCount FROM usersession WHERE useraction.errorCount > 5 OR useraction.duration > 10000 AND userId IS NOT NULL AND city IS NOT NULL ORDER BY totalErrorCount DESC
```

**Run query**

**Users with many errors or slow pages** 

 [Information \(1\)](#)

The number of results was limited to 50. To view more results, click [Load more](#).

Visualization: [Bar chart](#) [Line chart](#) [Pie chart](#) [Single value](#) [Table only](#)

Session details	USERNAME	city	totalErrorCount
	loren	Kizugawa	7
	null	Gdańsk	7
	Zara Hutahaean	Mersin	6
	delphinelan	Fes	5
	kusuma	Detroit	5
	qiang	Detroit	5
	sybylla	Siegen	5
	herold	Shenzhen	5
	bayu	Geelong	5

# **Lorem ipsum**

---

**Menu item 1** ◀

**Menu item 2**

**Menu item 3**

**Menu item 4**

**Menu item 5**

**Menu item 6**

# **Lorem ipsum dolor sit amet**

---

- Lorem ipsum dolor sit amet consectetur
- Adipiscing elit
  - Aenean mollis nunc
  - Etiam egestas orci
    - Integer non neque id metus ornare fringilla
- Proin egestas vitae est

# **Lorem ipsum**

---

Menu item 1

Menu item 2 ◀

Menu item 3

Menu item 4

Menu item 5

Menu item 6

## **Lorem ipsum dolor sit amet**

---

- Lorem ipsum dolor sit amet consectetur
- Adipiscing elit
  - Aenean mollis nunc
  - Etiam egestas orci
    - Integer non neque id metus ornare fringilla
- Proin egestas vitae est

# **Lorem ipsum**

---

**Menu item 1**

**Menu item 2**

**Menu item 3** ◀

**Menu item 4**

**Menu item 5**

**Menu item 6**

# **Lorem ipsum dolor sit amet**

---

- Lorem ipsum dolor sit amet consectetur
- Adipiscing elit
  - Aenean mollis nunc
  - Etiam egestas orci
    - Integer non neque id metus ornare fringilla
- Proin egestas vitae est

# **Lorem ipsum**

---

**Menu item 1**

**Menu item 2**

**Menu item 3**

**Menu item 4** ◀

**Menu item 5**

**Menu item 6**

## **Lorem ipsum dolor sit amet**

---

- Lorem ipsum dolor sit amet consectetur
- Adipiscing elit
  - Aenean mollis nunc
  - Etiam egestas orci
    - Integer non neque id metus ornare fringilla
- Proin egestas vitae est

# **Lorem ipsum**

---

**Menu item 1**

**Menu item 2**

**Menu item 3**

**Menu item 4**

**Menu item 5** ◀

**Menu item 6**

# **Lorem ipsum dolor sit amet**

---

- Lorem ipsum dolor sit amet consectetur
- Adipiscing elit
  - Aenean mollis nunc
  - Etiam egestas orci
    - Integer non neque id metus ornare fringilla
- Proin egestas vitae est

# **Lorem ipsum**

---

**Menu item 1**

**Menu item 2**

**Menu item 3**

**Menu item 4**

**Menu item 5**

**Menu item 6** ◀

# **Lorem ipsum dolor sit amet**

---

- Lorem ipsum dolor sit amet consectetur
- Adipiscing elit
  - Aenean mollis nunc
  - Etiam egestas orci
    - Integer non neque id metus ornare fringilla
- Proin egestas vitae est