

Real User Monitoring Lab

Throughout this lab, each section will be broken down into a series of steps. To navigate between sections, click each header to expand or collapse the sections.

Make sure you are logged into Datadog using the Datadog training account credentials provisioned for you. You can find that information by running `creds` in the lab terminal.

Review RUM Application Configuration

The Datadog RUM application has already been set up in your account. Take a look at how it's been set up:

1. Log in to Datadog using the trial credentials the lab created for you. You can run `creds` in the lab terminal whenever you need to retrieve your Datadog training account credentials.
2. Navigate to **UX Monitoring > RUM Applications**. You should see the Storedog application.
3. Click **Edit Application** on the Storedog application to check out the configurations.

Note: You may see **Setup Application**, depending on how long your lab has been running. If so, click that.

4. You will see options for integrating with different application types. You're integrating a web application by using JavaScript, so keep the default JS method selected.
5. Under **Instrument your application**, select **CDN Sync**. If you were using NPM to manage dependencies for a different application, you would integrate RUM by using the `@datadog/browser-rum` package.

You can see this already implemented in the Storedog application in the next step.

6. In the IDE, open `store-frontend/app/views/spree/layouts/spree_application.html.erb`.

This Ruby file is the main template for the application's HTML. By integrating the RUM script here, it will be executed on every application page.

You can see that the RUM script has already been added to this template, with many initialization arguments. (The code block starts with `window.DD_RUM &&`).

Notice the variable placeholders such as `<%= ENV['DD_CLIENT_TOKEN'] %>`. When this template is rendered, these will be replaced by the corresponding environment variables set in the `frontend` service's container. In this lab, the variables will be loaded from `/root/lab/.env` by Docker Compose.

You'll also notice that after initialization, the `startSessionReplayRecording()` method is called. This method will allow you to use **Session Replay** and play back the steps a user took during their session.

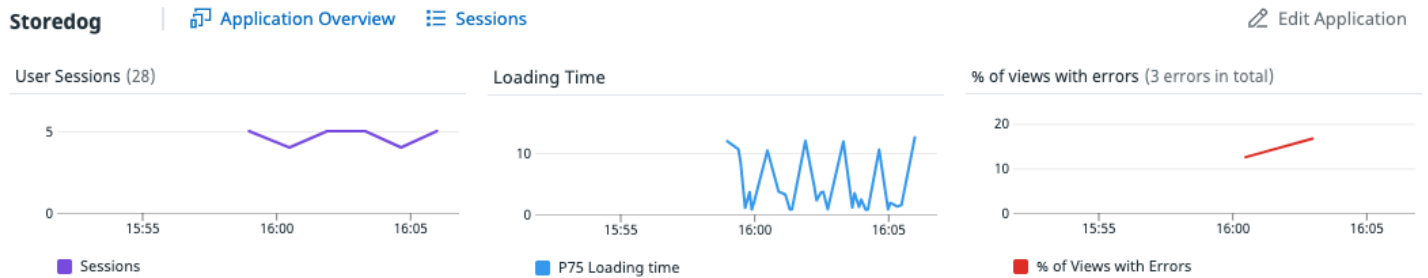
7. Click the **Storedog** tab to open the application. Explore Storedog by clicking around for a few minutes to generate some user session data. Be sure to add items to your cart and click **Checkout**.

Note: There is a Puppeteer service running to simulate user sessions as well. This way there will be plenty of events for you to view in Datadog.

8. Navigate back to the RUM Applications page in Datadog. After a minute or two, you'll see data for Storedog.

Now that the application is up and running with RUM, it is a good time to explore the different tools at your disposal for monitoring your application.

Explore RUM Application Overview



The **Application Overview** page shows you the high-level health of your application. With the information on this page, you can easily identify whether a new deployment has caused issues, generated a drop in usage, or caused an error spike in an application. Each section on the page contains a link to a built-in dashboard focused on one particular area. All metrics shown are alertable and customizable for your business needs.

1. Click on **Application Overview** for the Storedog app.
2. When the **RUM Applications** page opens, review the following bullet points to learn about the function of each section:

- At the top of the page, you can see drop-down lists for **Env:***, **Version:***, and other variables. These are known as template variables, and by adjusting their settings, you can limit the data displayed on the page to a specific environment, version, browser, view name, or other dimension available for filtering. For example, if you use the **Browser Name** drop-down menu to select a specific browser, you would filter the information on this page to show data reporting from that browser only.
- **Core Web Vitals P75** are a set of three metrics designed to monitor a site's user experience. These metrics focus on load performance, interactivity, and visual stability. Each metric comes with guidance on the range of values that translate to good user experience.

These graphs display values in the upper quartile of all values, labeled **P75**. "P75" refers to the values that are equal to or above the 75th percentile within the set. In other words, of all values, those in the graph represent the "top" 25%. But because Core Web Vitals measure time, larger values—in this case—are bad. Thus, these values in practice represent the worst performance values in the selected time period.

- **Page Views by Version** and **Total Errors by Version** give you a quick reference for application activity and errors, separated by application version. You may recall seeing the version as part of the configuration. The version number can be set through a variable when the application is built.
- Moving down the page, you'll see **User Analytics**. Here you'll see details on where your users are coming from, what browsers and devices are being used, and individual page activity. This information can help you to identify where to add CDN locations, to determine the top browsers and devices used to access your application, and to see how much time users spend on top pages.

At the top of the **User Analytics** section, you'll see links to view **Session Replays**, **Recent Sessions**, and the **User Analytics Dashboard**. Each section has a pre-built dashboard available.

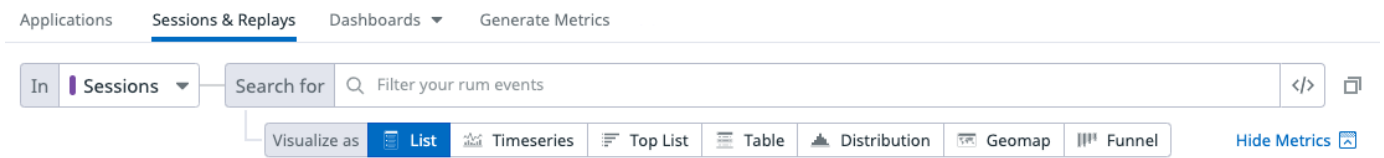
- The **Performance Monitoring** section gives information about how each page in the application is performing. Here you can sort the list by clicking on a heading to see the slowest pages based on their loading times or Core Web Vitals scores.
- The last section is for **Errors**, which displays **Total Errors** and the **Error Rate by Page**. In addition, the top of this section has links to view other pages about errors, such as the **Error Tracking** page and the **Errors Dashboard**.

The **Error Tracking** feature can show you JavaScript runtime errors, complete with stack traces and source code annotations. You must upload source maps to enable this feature, and you can learn more about this topic in the error tracking documentation.

Explore a User Session

One of the major benefits that Datadog RUM offers is the ability to dive into granular data.

1. Scroll to the top of the Storedog application's **Application Overview** page. (To return to this page, navigate to **UX Monitoring > RUM Applications**, and click **Application Overview** for the Storedog app.)
2. Click on the **Sessions & Replays** tab:

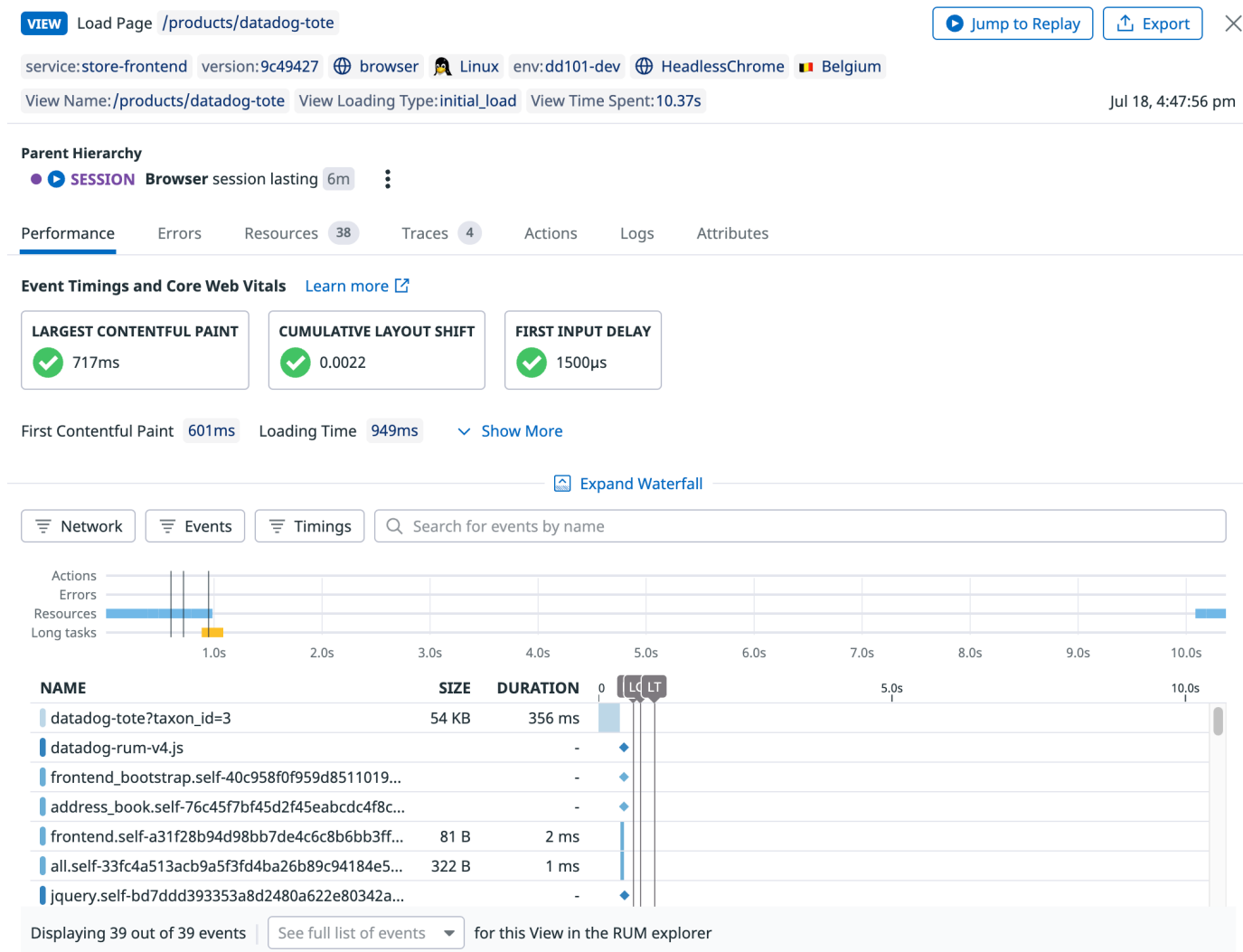


Use the **In** dropdown at the top left to review the event types within RUM data that you are able to search. The selection defaults to **Sessions**, which is a logical container for all RUM events collected during a user's session.

3. Under the **Session count** bar graph, you will see a list of user sessions. Click one of these user sessions to open up the side panel:

RELATIVE TIME	TYPE	SERVICES	EVENT
0	View Load	store-frontend	Load Page /
20.4 s	View Load	store-frontend	Load Page /t/bags
31.6 s	View Load	store-frontend	Load Page /products/datadog-tote
42.8 s	View Load	store-frontend	Load Page /cart
53.8 s	Custom	store-frontend	Custom user action checkout on page /cart
53.9 s	View Load	store-frontend	Load Page /checkout/registration
1 min 41 s	View Load	store-frontend	Load Page /checkout/address
3 min 39 s	View Load	store-frontend	Load Page /checkout/delivery

4. Under **Session**, click on a **View Load** event to open up another side panel:



By looking into this single view, you can see performance, errors, resources, user actions, and other information associated with this page load. This data can help debug and rectify issues for your users. If you share the URL from the browser on your view with an engineer, for example, that engineer will be brought to the same panel you are looking at. There will be no discrepancy of information.

- Take some time to explore this view. For example, on the **Performance** tab, hover over and click on different resources listed in the “waterfall” of the page load. Then you can click on each of the **Core Web Vitals** and select the option to “search views with poor performance.” Also be sure to click on the different tabs available to review the information presented in them.
- Close the **View** pane and the **Session** side panels. Back in the list of sessions, notice the column **ACTION COUNT**. Open a session with at least one action, and click on one of the **non-custom** user actions from the list, such as a **Click** action:

10 Views Loaded

3 User Actions

0 Errors

0 Frustration Signals

Session

Errors

Attributes

Q Search for events by name



13 events in this session including **10 views loaded**, **3 user actions**, and **0 errors**

RELATIVE TIME	TYPE	SERVICES	EVENT
0	View Load	store-frontend	Load Page /
17.1 s	View Load	store-frontend	Load Page /t/bags
27.1 s	View Load	store-frontend	Load Page /products/datadog-tote
37.5 s	View Load	store-frontend	Load Page /cart
47.3 s	Custom	store-frontend	Custom user action checkout on page /cart
47.3 s	View Load	store-frontend	Load Page /checkout/registration
1 min 34 s	View Load	store-frontend	Load Page /checkout/address
3 min 32 s	View Load	store-frontend	Load Page /checkout/delivery
3 min 48 s	View Load	store-frontend	Load Page /checkout/payment
3 min 51 s	Click	store-frontend	Click on Credit Card on page /checkout/payment
5 min 21 s	View Load	store-frontend	Load Page /checkout/confirm
5 min 37 s	View Load	store-frontend	Load Page /checkout/payment
5 min 40 s	Click	store-frontend	Click on Credit Card on page /checkout/payment

Note: Click actions are only displayed if they do not result in a new page load. Therefore even if there are many clicks displayed in an animated replay, only a couple are likely to show up in the actions list.

7. On the **Action** side panel that opens, you'll find a detailed view of how that action performed, any logs associated with it, what resources it used, and any errors that occurred because of the action.

Searching RUM Data

You can search and filter your RUM events by typing in the search bar and selecting attributes from the facets on the left side of the page. You can narrow down, broaden, and shift your focus on subsets of events.

1. Navigate back to the **Sessions & Replays** tab for the Storedog app in Datadog.
2. In the **Search facets** box in the left pane, type **session** to view attributes related to sessions:

Note: Be sure to select the **Error Count** for **session** not **view**! You can tell which event type an attribute or timing metric is related to by checking the start of the name. For example, `@session.error.count` is the error count related to *session data*, whereas `@view.error.count` is the error count related to *view data*.

Error Count

Initial Connection Start (@resource.connect.start)

Redirect Start (@resource.redirect.start)

Resource Duration (@resource.duration)

Secure Connection Start (@resource.ssl.start)

Time To First Byte Start (@resource.first_byte.start)

Long Task Duration (@long_task.duration)

Action Count (@session.action.count)

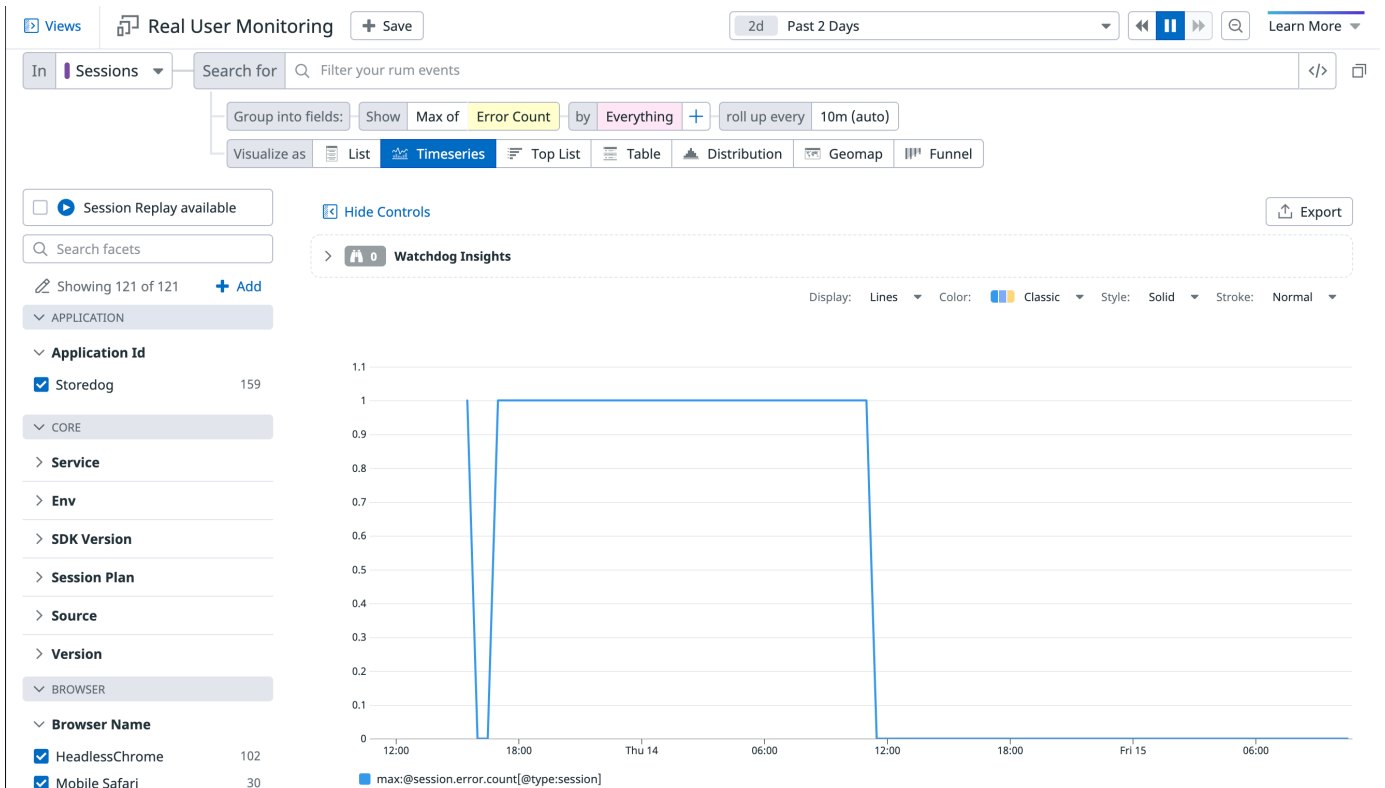
Crash Count (@session.crash.count)

Error Count (@session.error.count)

Error Count (@session.error.count)

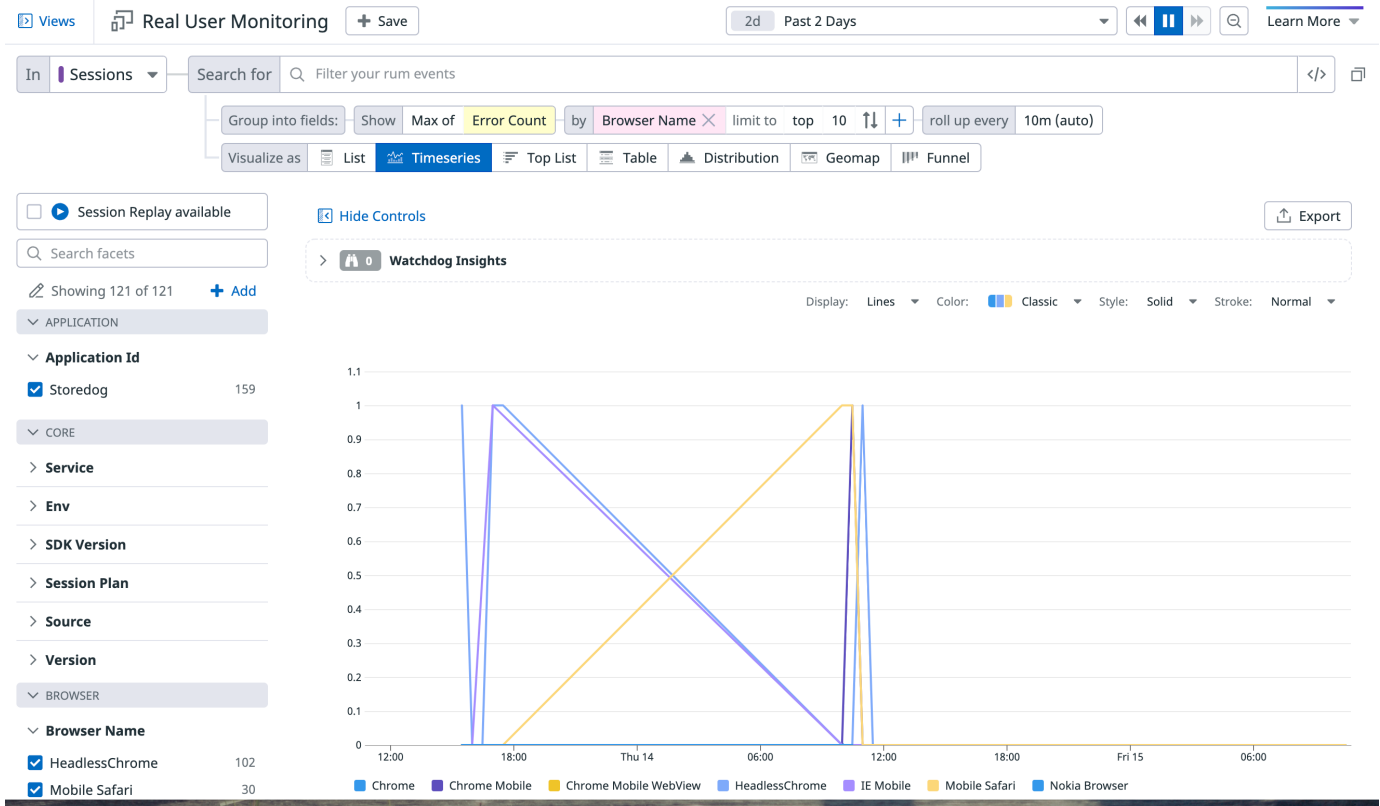
Number of errors recorded for the session.

6. Next to **Show**, click **Avg** of and change this value to **Max** to see a graph of the count of max errors for the sessions:



7. You can also group the data by attributes for the session.

Next to **by**, click **Everything** and select **Browser Name (@browser.name)** from the dropdown. The results will now be grouped by browser, allowing you to see if a specific browser has more errors than others:

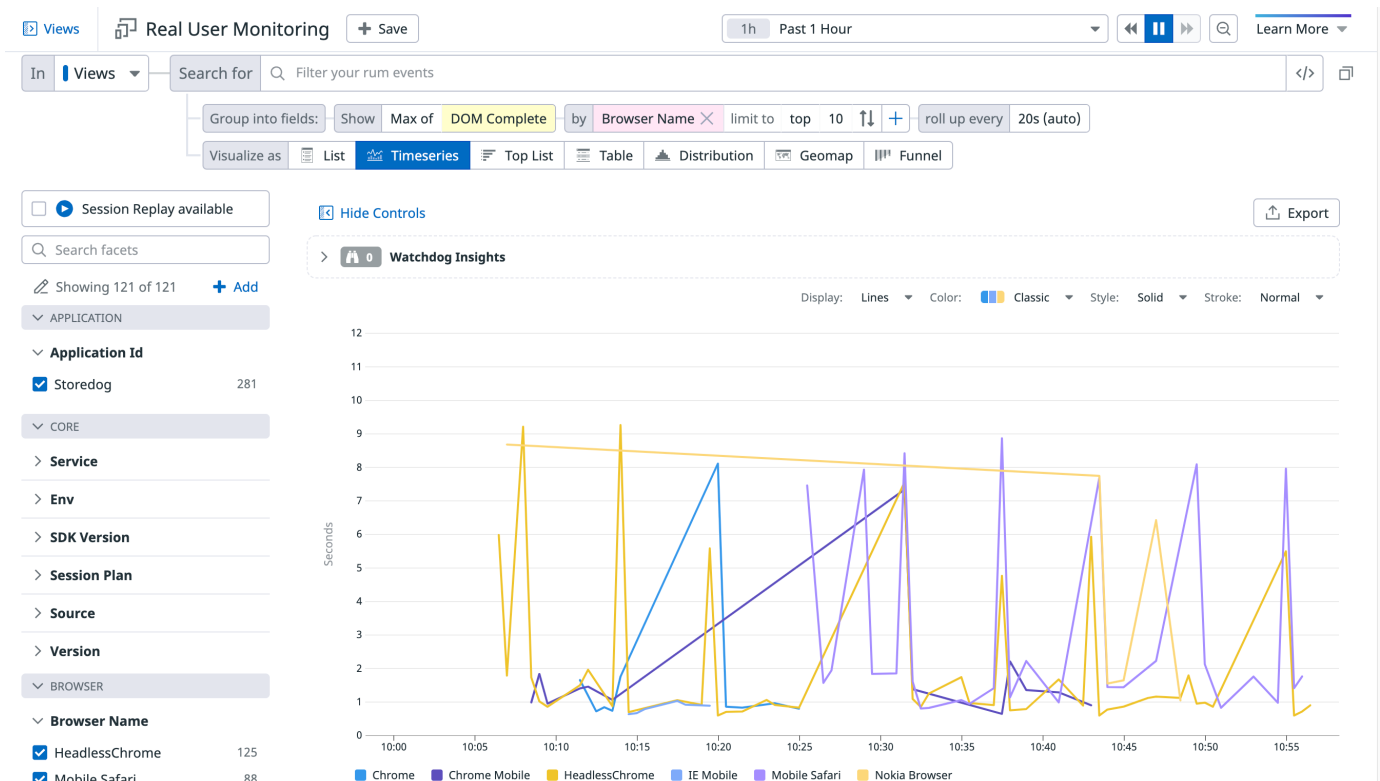


8. You can also graph timing metrics, such as DOM Complete, which refers to the time taken to parse and load an entire document along with its sub-resources.

Click **Error Count** and select **DOM Complete (@view.dom_complete)** from the dropdown.

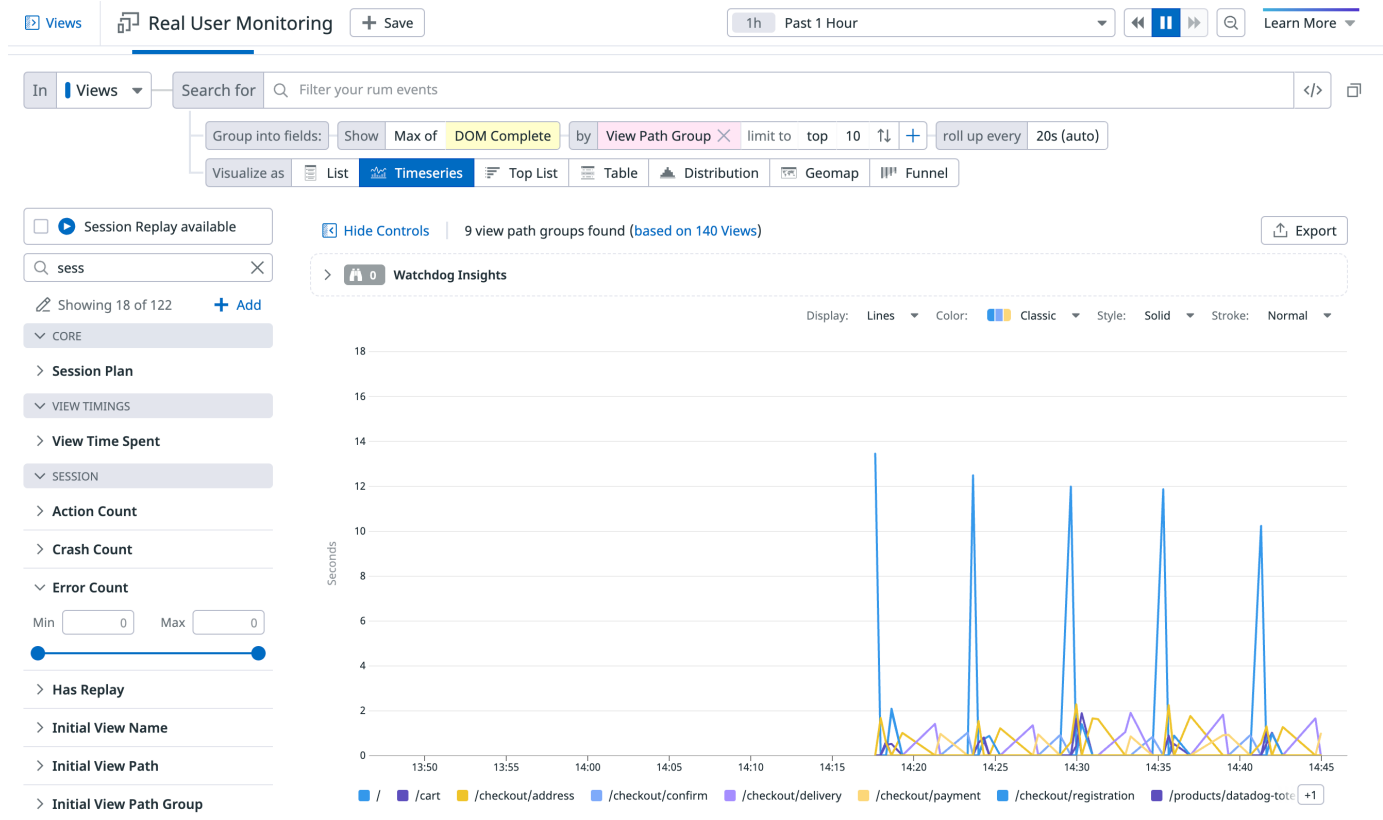
9. The graph will be empty at first because you're still looking at session data; the DOM Complete metric appears only within Views.

Next to **In**, click **Sessions** and select **Views** from the dropdown. Now the graph will show the average time a page and all of its sub-resources took to parse and load:



10. You can choose different attributes by which to group the results. For example, try grouping by **View Path Group** (@view.url_path_group) instead of browser name.

Next to **by**, scroll down and select **View Path Group** in the dropdown:



11. There are times when a timeseries graph may not be the best way to visualize your data.

Next to **Visualize as**, select **Top List** to see the same data in a top 10 list. This type of visualization makes it easier to see which path group has the slowest time to DOM Complete:



12. You can filter the results by choosing an option in the facet bar on the left.

For example, selecting specific values from the **VIEW URL** group or **VIEW** group will change how the results are displayed:

VIEW URL

> View Path

> View Path Group

> View Hash

> View Host

> View Scheme

> View URL

VIEW

> View Id

View Name

<input checked="" type="checkbox"/>	/	69
<input checked="" type="checkbox"/>	/checkout/payment	66
<input checked="" type="checkbox"/>	/t/bags	59
<input checked="" type="checkbox"/>	/products/datadog-t...	56
<input checked="" type="checkbox"/>	/cart	51
<input checked="" type="checkbox"/>	/checkout/registration	46
<input checked="" type="checkbox"/>	/checkout/address	44
<input checked="" type="checkbox"/>	/checkout/delivery	38
<input checked="" type="checkbox"/>	/checkout/confirm	33

> Error Count

When you change merely how you group or visualize the data, the search values will not change.

Create Funnels

Funnel analysis enables you to use your RUM sessions to visualize aggregated traffic across each step in key user journeys. Once you select the sequence of views and actions you want to analyze, the resulting funnel graph shows what percentage of users move from each step to the next one. By observing how many users make the jump between steps—and how many don't—you can monitor the rate of success for the workflow and see where there might be sources of friction that are causing users to churn away.

To help shed more light on your application's health, you can also add funnels to your dashboards alongside your other key performance indicators.

1. Navigate back to the **Sessions & Replays** tab for the Storedog app in Datadog.
2. Next to **Visualize as**, select **Funnel**.
3. In the central area of the window, beneath Watchdog Insights, a new dropdown menu appears with **View** selected. Beneath this new menu, a **View or Action Name** field appears, alongside an add function (+) button, and a **Reset** button.

 **Hide Controls**

>  0 **Watchdog Insights**

View ▼

View or Action Name



Reset

4. Click on the **View or Action Name** field to reveal the dropdown of query options available:

View ▾

View or Action Name

VIEWS
/checkout/payment
/t/bags
/products/datadog-tote
/
/cart
/checkout/registration
/checkout/address
/checkout/delivery
/checkout/confirm

+

Reset

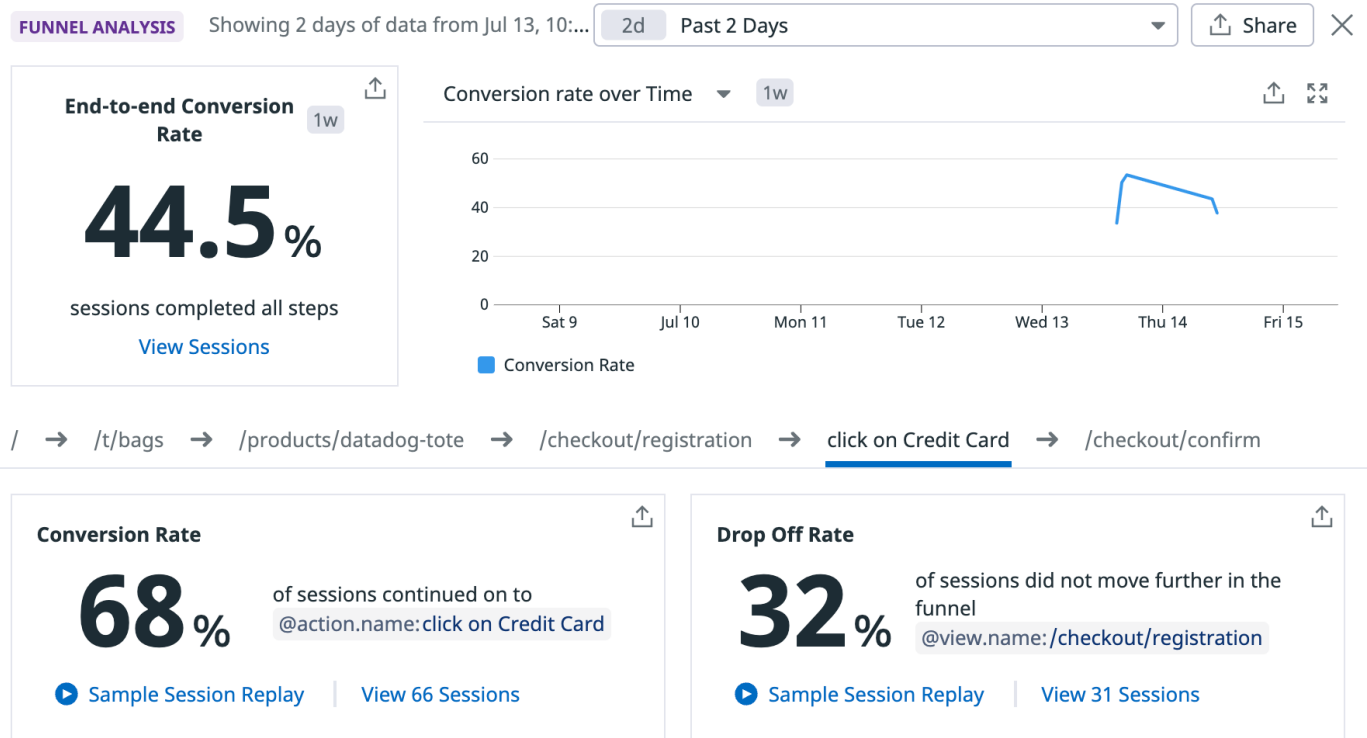
5. To begin constructing a funnel, choose a query from the dropdown menu. For example, you might want to choose **Views** → **/**.
6. Click **+** and select another query from the dropdown menu to visualize the funnel. For example, you might choose **Views** → **/t/bags**. A conversion rate between the two steps and associated bar graph visualization is displayed below the funnel definition. Go ahead and build a funnel with the following query options in this order:
 - View **/**
 - View **/t/bags**
 - View **/products/datadog-tote**
 - View **/cart**
 - Action **checkout**
 - View **/checkout/confirm**

That will result in a bar graph similar to the following image:



Note: This is illustrated best if you have more than 40 sessions recorded, so if you aren't seeing a lot of data, wait a few minutes and try again.

- Click on a bar on the bar graph. A funnel analysis appears in a side panel, displaying the step's **Conversion Rate** and **Drop Off Rate**:



Note: Scroll down to see information about the step's performance, along with information about user behavior and any associated outstanding issues. Under **User Behavior**, the **Conversion rate by Country** is a dropdown that provides more options by which to group conversion rate.

- In the **Conversion Rate** and **Drop Off Rate** boxes, click on the buttons that say **Sample Replay Session**. For each, you'll be shown a video of a user's session beginning with this step that ends in a conversion and a drop-off, respectively.

The `startSessionReplayRecording()` function we saw at the beginning of this lab is what allows you to play back a user session. This Session Replay feature shows you how a user has interacted with your site and helps you determine what might have led to an error, performance issue, or drop off.

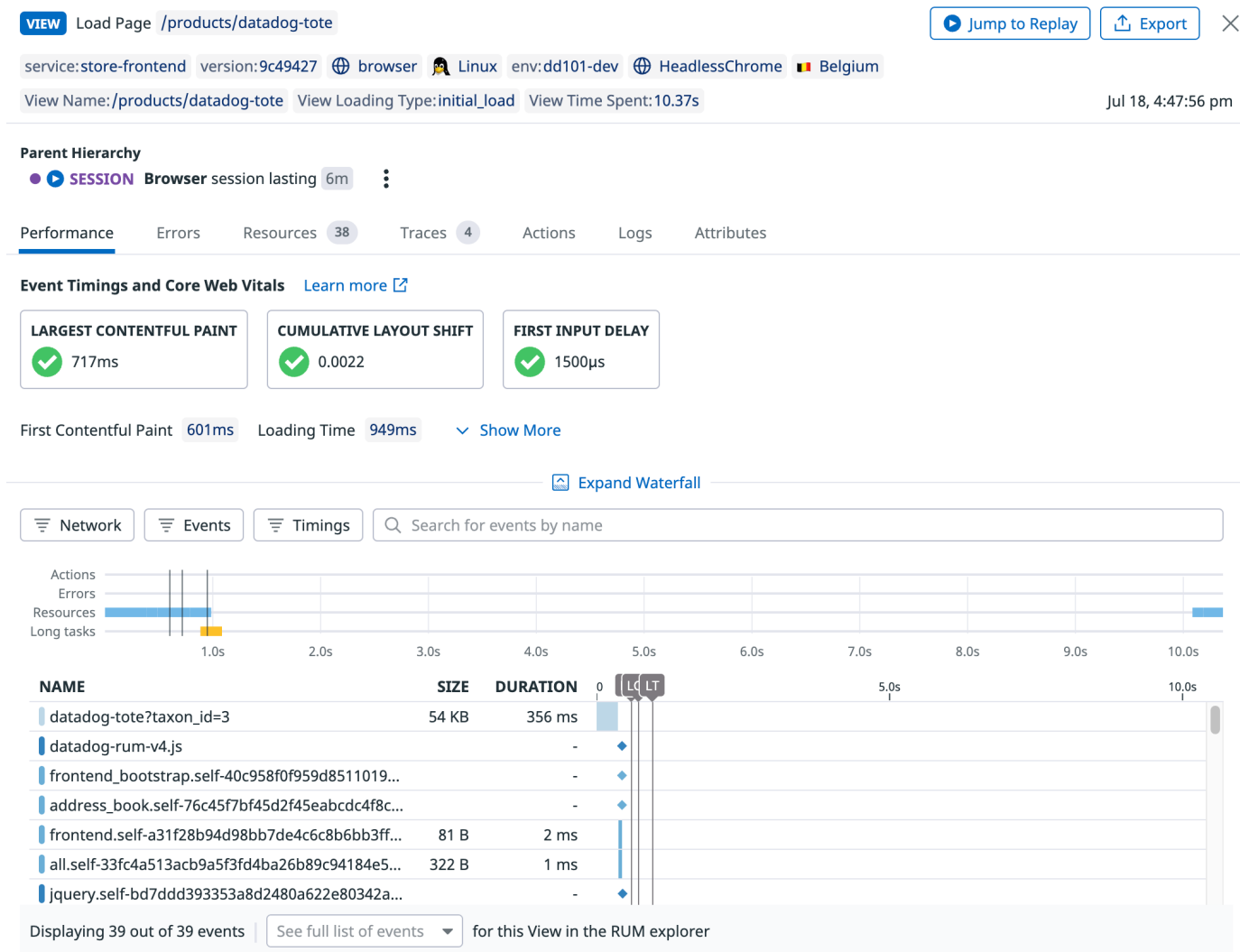
Explore Session Replay

Session Replay allows you to see a pixel-by-pixel recreation of your users' journeys. To explore this feature, you can use the **User Journey** event timeline to navigate directly to a captured user activity.

1. Navigate back to the **Sessions & Replays** tab for the Storedog app in Datadog.
2. In the **In** dropdown, set the event type to **Sessions**, if this option is not already selected.
3. In the **Search for** field, clear all of your search terms.
4. Next to **Visualize as**, select **List**. A list of sessions appears.
5. Locate the session in the list that has the highest value in the **Time Spent** column.

Click on the play arrow icon to the left of this session. The **Browser Session Replay** window appears, with a **USER JOURNEY** timeline on the right.

6. Hover over one of the items in the timeline and click **Details**. The side panel that opens is the same that you looked at earlier in this lab.



- Close the side panel.
- In the **Browser Session Replay** window, click the **Dev Tools** button at the top of the page. You'll now see additional resources for analyzing and troubleshooting the currently selected user journey.
- Click the play arrow at the bottom left of the player to restart playback. The **Dev Tools** section will update and follow along with the playback. This additional information can help identify performance issues that a user experienced during the session.

Lab Conclusion

This lab gave you a quick introduction to what you can do with Real User Monitoring. As with all the other topics in this introductory course, there is so much more you can do with RUM. Be sure to read the RUM Documentation to learn more.

When you're done, enter the following command in the terminal:

```
finish
```

Click the **Check** button in the lower right corner of the lab and wait for the lab to close down before moving on to the next lesson.