

[Foundation](#) / [URL Loading System](#) / Analyzing HTTP traffic with Instruments

Article

Analyzing HTTP traffic with Instruments

Measure HTTP-based network performance and usage of your apps.

Overview

As a component of Instruments, a performance-analysis and testing tool for iOS, iPadOS, watchOS, tvOS, and macOS apps, the HTTP Traffic instrument intercepts, records, and visualizes targeted processes.

Important

This instrument records both encrypted and unencrypted HTTP traffic in the trace document and in system logs, in an unencrypted state. These trace documents and log files may contain sensitive information.

The instrument internally gathers detailed information—including header and body content, connection setup duration, and request and response times—from the URL Loading System. It then records this data and recombines it into precise intervals with additional metadata.

Related sessions from WWDC21

Session 10212: [Analyze HTTP traffic in Instruments](#)

Launch Instruments and start recording

From Xcode's Product menu, choose Profile. After Instruments launches, select the Network template, then click Choose.

Choose a profiling template for:  iPhone 12 mini (15.0) >  Dogs!

Standard Custom Recent

 Filter



Blank



Activity Monitor



Allocations



Animation Hitches



App Launch



Core Data



CPU Counters



CPU Profiler



File Activity



Game Performance



Leaks



Logging



Metal System Trace



Network



SceneKit



SwiftUI



System Trace



Time Profiler



Network

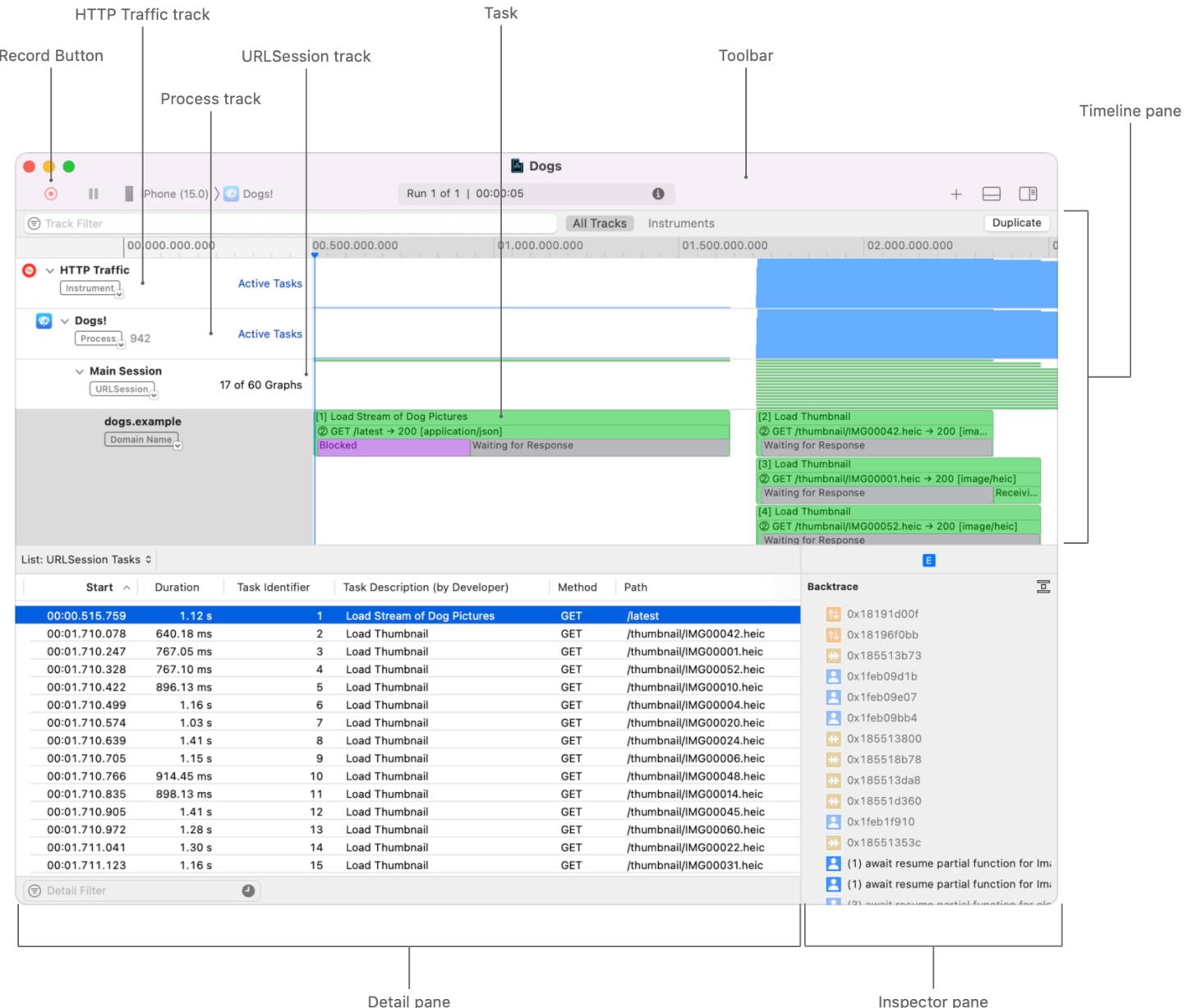
Analyze how your applications are making HTTP requests and are using TCP/IP and UDP/IP connections using the Connections instrument.

Open an Existing File...

Cancel

Choose

The Network Connections and HTTP Traffic instruments appear in a new window, along with the timeline and detail panes. The following figure highlights the areas of interest.

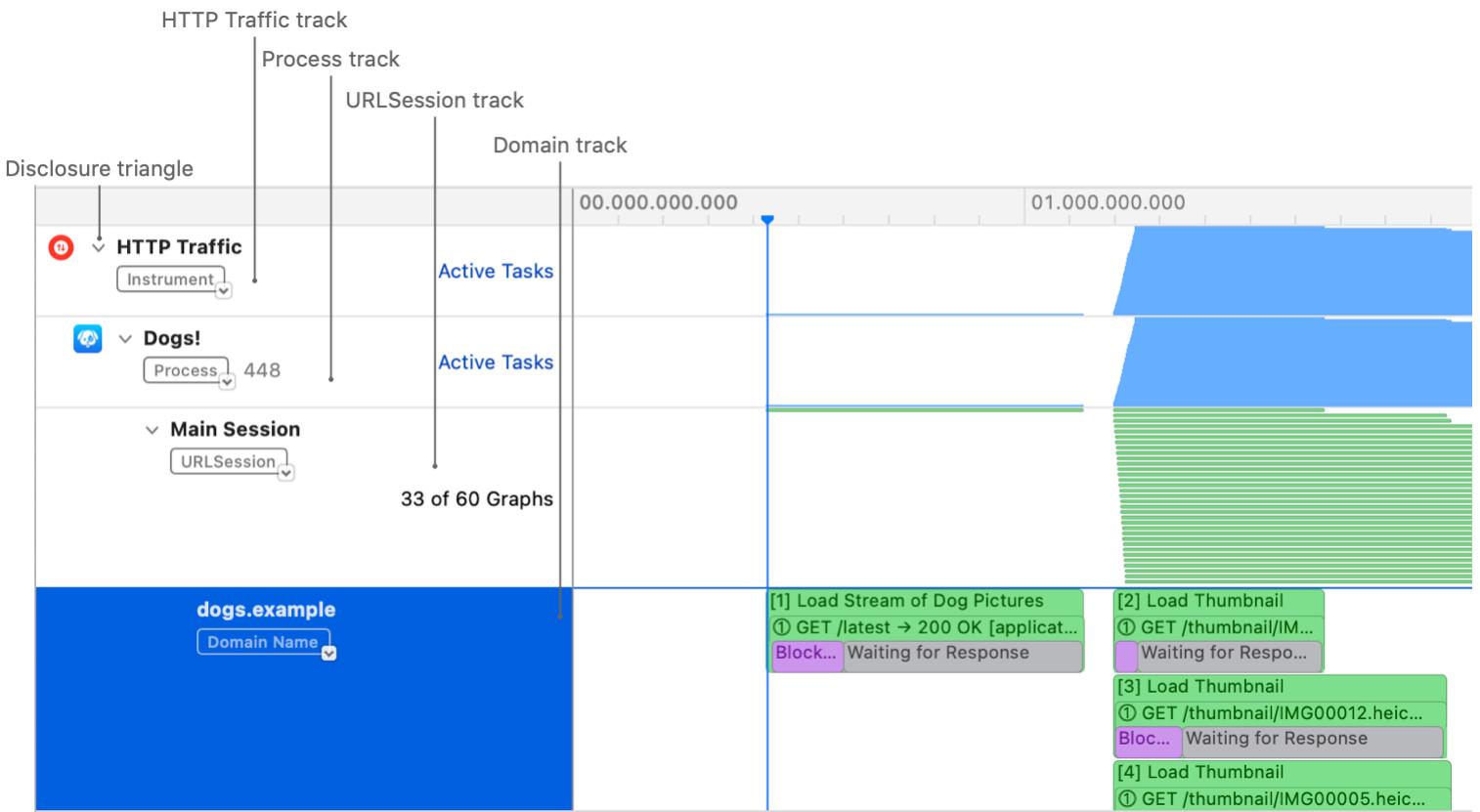


Click the Record button in the toolbar. A dialog appears that explains the risks associated with capturing HTTP traffic and storing it unencrypted. If you accept this risk, click Record Anyway. The app launches and the instrument starts monitoring.

View the track hierarchy

The HTTP Traffic track is the topmost track. It shows all HTTP traffic for the current recording session. You can also view a break down of the captured traffic across different dimensions using a hierarchy of the HTTP Traffic instrument subtracks. To reveal the subtracks, click the disclosure triangle next to HTTP Traffic.

The first hierarchy level organizes the traffic by process, revealing one track per captured process. Each process contains one or more URLSession tracks, and each session contains a set of one or more domain tracks.



The HTTP Traffic track and the process tracks show an aggregated view that displays how many tasks or transactions are active at any point. Session tracks display individual intervals for each task and transaction.

A session track corresponds to a [URLSession](#) that you create in code. The session name comes from the [sessionDescription](#) property. Naming your session allows you to reference it when viewing HTTP traffic for your app. Here's an example showing how to name a session:

```
let session = URLSession(configuration:.default)
session.sessionDescription = "Main Session"
```

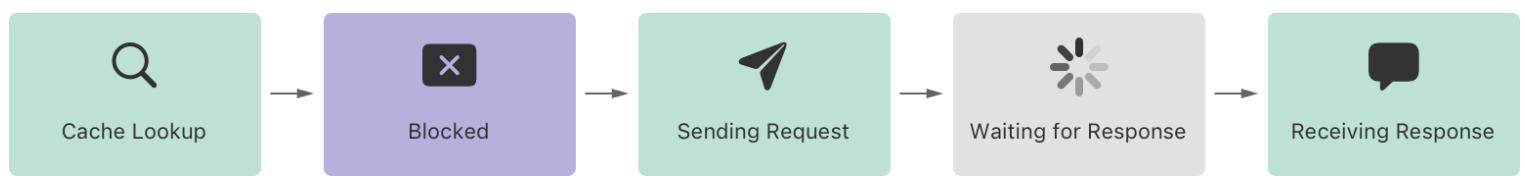
View tasks and transactions

There are times when completing a [URLSession](#) task requires several rounds of requests and responses. Each pair of requests and responses is a *transaction*.

Domain tracks provide an increased level of detail. In addition to information about the task, they also display the underlying transactions and transaction states. Here's the dogs.example domain track showing a task with two transactions.

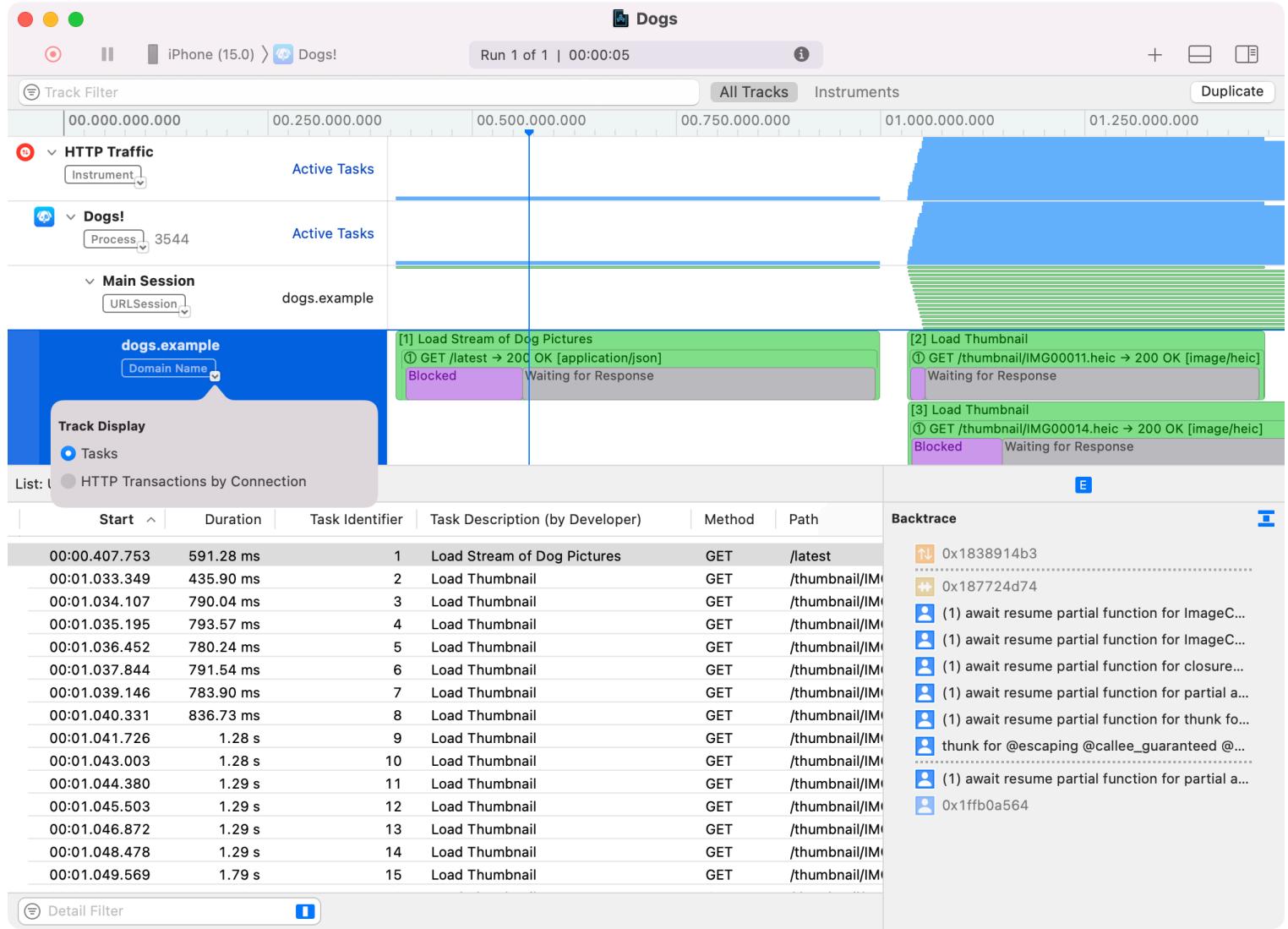


A transaction has five states that occur in the following order: cache lookup, blocked, sending request, waiting for response, and receiving response. The following illustration shows these states.

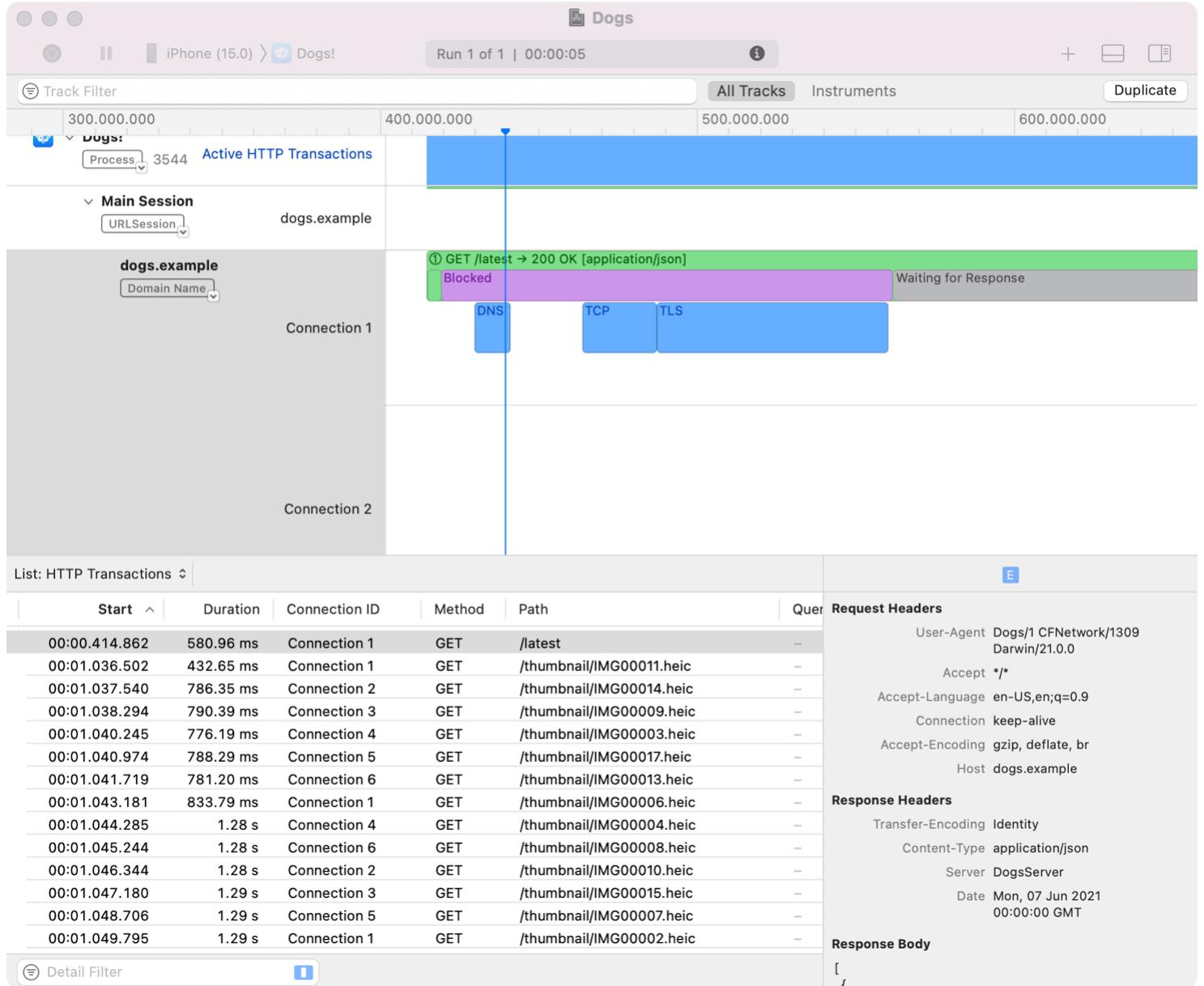


Some states have a very short duration compared to others. For example, cache lookup may take a few milliseconds, whereas blocking occurs for a second or longer. This leads to the cache lookup state not being visible at the default zoom level. If you'd like to view a specific state, use your mouse or trackpad to zoom in to the corresponding area of the transaction to reveal more information.

Instruments provides two display styles in the track area. The default display style shows a list of NSURLSession tasks. Here are the tasks for the dogs.example session.



You can also display transactions grouped by connection. To switch to this style, click the arrow next to the domain name. This helps you understand how your transactions execute and the influence of the underlying connections. Here are the transactions grouped by connection.



The detail pane and inspector update when you change the track display style. When selecting the Tasks display, the detail pane shows information about the tasks. The inspector shows the backtrace of when the selected task resumed.

When selecting the HTTP Transactions by Connection display, the detail pane shows information about the transactions. The inspector shows the request headers, response headers, and the request and response body of the selected transaction.

In addition to the two lists in the detail pane, there's a summary view. From the pop-up menu, select "Summary: Transaction Durations." The summary view groups connections hierarchically by IP address, connection, and path.

Summary: Transaction Durations ▾

Server IP / Connection ID / Path / Transaction ID	Count	Avg Duration	Max Durati...	Total Durati...	A
✓ * All *	61	2.92 s	5.32 s	2.97 min	
✓ 192.168.0.239	61	2.92 s	5.32 s	2.97 min	
✓ Connection 1	12	2.64 s	5.31 s	31.64 s	
✓ /thumbnail/IMG00039.heic	1	5.31 s	5.31 s	5.31 s	
F8F34CEB-79D2-41A0-AB68-C7FB0BD302FE	1	5.31 s	5.31 s	5.31 s	
> /thumbnail/IMG00017.heic	1	4.80 s	4.80 s	4.80 s	
> /thumbnail/IMG00024.heic	1	4.30 s	4.30 s	4.30 s	
> /thumbnail/IMG00006.heic	1	3.80 s	3.80 s	3.80 s	
> /thumbnail/IMG00051.heic	1	3.29 s	3.29 s	3.29 s	
> /thumbnail/IMG00009.heic	1	2.78 s	2.78 s	2.78 s	
> /thumbnail/IMG00042.heic	1	2.28 s	2.28 s	2.28 s	
> /thumbnail/IMG00035.heic	1	1.77 s	1.77 s	1.77 s	
> /thumbnail/IMG00036.heic	1	1.28 s	1.28 s	1.28 s	
> /thumbnail/IMG00043.heic	1	870.49 ms	870.49 ms	870.49 ms	

 Detail Filter

See Also

Essentials

 Fetching website data into memory

Receive data directly into memory by creating a data task from a URL session.

`class URLSession`

An object that coordinates a group of related, network data transfer tasks.

`class URLSessionTask`

A task, like downloading a specific resource, performed in a URL session.