

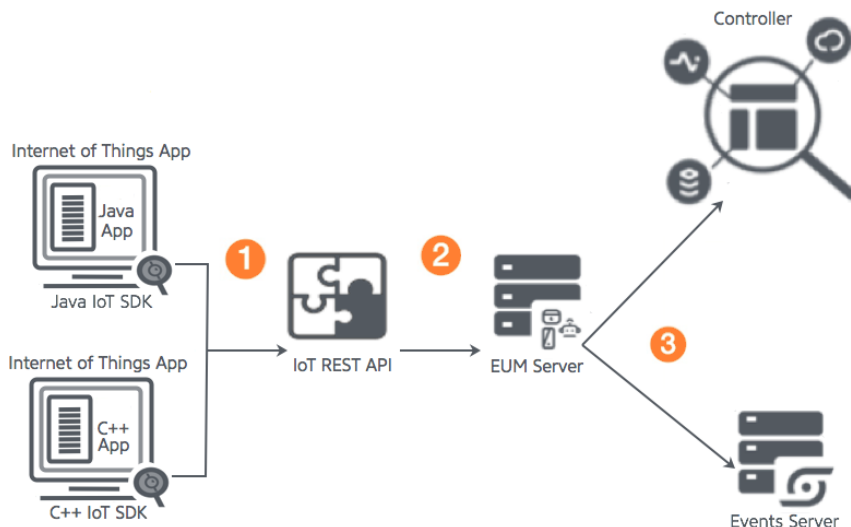
## IoT Monitoring

**On this page:**

- [Benefits of IoT Monitoring](#)
- [IoT Monitoring Use Cases](#)
- [IoT SDKs](#)
- [IoT Monitoring UI](#)
- [IoT Monitoring Workflow](#)
- [IoT Analytics](#)
- [IoT Licenses and Limits](#)

AppDynamics IoT Monitoring enables you to track and understand the transactions of your IoT applications. Because IoT devices are diverse, both in terms of the platforms they use and their business function, AppDynamics has developed a REST API in addition to language SDKs to provide the maximum flexibility for reporting IoT data. This API can be used from any device that supports HTTPS and is connected to the Internet.

IoT Monitoring requires application developers to instrument their code. To make this process easier, AppDynamics has developed C/C++ and Java SDKs, so that developers using the platforms supporting these languages can leverage the features of the SDK instead of using the REST API. The IoT SDKs **1** use the REST APIs to report IoT Data to the EUM Server **2**, where the data is aggregated and made available to the AppDynamics Controller and the Events Service **3** as shown in the diagram below.



## Benefits of IoT Monitoring

IoT Monitoring enables you to capture and diagnose network requests, errors, and custom application and domain-specific events for applications running on devices that impact business objectives.

With IoT Monitoring, you can monitor the following for devices:

- **Availability**
- **Network Performance:**
  - slow interaction
  - latency between the device and backend services
- **Usage:** User and system behavior and patterns
- **Errors and Exceptions**

## IoT Monitoring Use Cases

IoT applications are typically embedded applications running on connected devices. This section focuses on how IoT Monitoring can collect and report data for the most common categories of IoT applications.

In the tabs below, you can view these IoT app categories and see how a typical example of each would use the three supported IoT Monitoring events to report data.

Retail

This category of IoT apps process orders, payments, and manage product inventory. Examples include point of sale (PoS) devices, smart shelves, and gateways (local servers that aggregate information within certain geography/region).

Device Example	Device Information	Example Network Requests	Custom Events Examples	Handled Errors	Exceptions/Crashes
Point of Sale	<ul style="list-style-type: none"> <li>• device ID</li> <li>• app version</li> <li>• hardware versions</li> <li>• firmware versions</li> </ul>	<ul style="list-style-type: none"> <li>• checkout</li> <li>• payment transaction</li> </ul>	<ul style="list-style-type: none"> <li>• total sales</li> <li>• number of items sold</li> <li>• discount applied</li> <li>• total checkout time</li> <li>• total processing time</li> </ul>	<ul style="list-style-type: none"> <li>• credit card declined</li> <li>• unable to read credit card</li> <li>• incorrect pin code</li> </ul>	<ul style="list-style-type: none"> <li>• crashes</li> <li>• reboots</li> <li>• frozen UI</li> <li>• timeouts</li> <li>• null point exceptions</li> </ul>

Media

IoT apps in this category enable users to seamlessly experience media content across different devices. An IoT app on a [set-top box \(STB\)](#) can quickly change the channel/content based on what the user was watching or listening to on their computer or phone.

Device Example	Device Information	Example Network Requests	Custom Events Examples	Handled Errors	Exceptions/Crashes
Audio Streaming	<ul style="list-style-type: none"> <li>• device ID</li> <li>• app version</li> <li>• hardware versions</li> <li>• firmware versions</li> </ul>	<ul style="list-style-type: none"> <li>• media play</li> <li>• pause</li> <li>• stop requests</li> <li>• ad performance</li> </ul>	<ul style="list-style-type: none"> <li>• media downloads</li> <li>• media skips</li> <li>• subscript details</li> <li>• total play time</li> </ul>	<ul style="list-style-type: none"> <li>• play errors</li> <li>• song not available</li> <li>• download errors</li> </ul>	<ul style="list-style-type: none"> <li>• crashes</li> <li>• reboots</li> <li>• frozen UI</li> <li>• timeouts</li> <li>• null point exceptions</li> </ul>

#### Connected Cars

Because automotive head units are often computers, IoT apps are used to manage user subscriptions and monitor the user experience.

Device Example	Device Information	Example Network Requests	Custom Events Examples	Handled Errors	Exceptions/Crashes
Infotainment System	<ul style="list-style-type: none"> <li>• device ID</li> <li>• app version</li> <li>• hardware versions</li> <li>• firmware versions</li> </ul>	<ul style="list-style-type: none"> <li>• media play</li> <li>• pause</li> <li>• stop requests</li> <li>• ad performance</li> </ul>	<ul style="list-style-type: none"> <li>• media downloads</li> <li>• media skips</li> <li>• subscript details</li> <li>• total play time</li> </ul>	<ul style="list-style-type: none"> <li>• application loading errors</li> <li>• connectivity errors</li> <li>• audio errors</li> <li>• launching errors</li> </ul>	<ul style="list-style-type: none"> <li>• crashes</li> <li>• reboots</li> <li>• frozen UI</li> <li>• timeouts</li> <li>• null point exceptions</li> </ul>

#### Smart Homes

These IoT apps improve the security as well as reduce the energy and maintenance costs of intelligent buildings and smart homes. Examples include home security devices and smart thermostats, smart keyless locks, and smart lights.

Device Example	Device Information	Example Network Requests	Custom Events Examples	Handled Errors	Exceptions/Crashes
----------------	--------------------	--------------------------	------------------------	----------------	--------------------

Home Security	<ul style="list-style-type: none"> <li>• device ID</li> <li>• app version</li> <li>• hardware versions</li> <li>• firmware versions</li> </ul>	<ul style="list-style-type: none"> <li>• home sensor status</li> <li>• communication between gateway devices and sensors</li> <li>• health alerts</li> </ul>	<ul style="list-style-type: none"> <li>• sensor health</li> <li>• number of alerts</li> <li>• connectivity status</li> <li>• total roundtrip time to trigger an alarm</li> <li>• total time to set security status</li> </ul>	<ul style="list-style-type: none"> <li>• device wake-up/sleep</li> <li>• alarm notification,</li> <li>• connectivity issues</li> </ul>	<ul style="list-style-type: none"> <li>• crashes</li> <li>• reboots</li> <li>• frozen UI</li> <li>• timeouts</li> <li>• null point exception</li> </ul>
---------------	--	--	---	--	---

Travel

This category of IoT apps monitors and improves the travel experience. You can find IoT apps used in airport kiosks, luggage tracking devices, and embedded in cabin seats to monitor passenger comfort.

Device Example	Device Information	Example Network Requests	Custom Events Examples	Handled Errors	Exceptions/Crashes
Airport Kiosk	<ul style="list-style-type: none"> <li>• device ID</li> <li>• app version</li> <li>• hardware versions</li> <li>• firmware versions</li> </ul>	<ul style="list-style-type: none"> <li>• fetching passenger records</li> <li>• payment processing</li> </ul>	<ul style="list-style-type: none"> <li>• number of users completing a transaction</li> <li>• number of issued boarding passes</li> <li>• total user activity time</li> <li>• total time spent on each screen</li> </ul>	<ul style="list-style-type: none"> <li>• passenger records not found</li> <li>• printer error</li> <li>• connectivity issues</li> </ul>	<ul style="list-style-type: none"> <li>• crashes</li> <li>• reboots</li> <li>• frozen UI</li> <li>• timeouts</li> <li>• null point exceptions</li> </ul>

## IoT SDKs

IoT Monitoring provides SDKs for C/C++ and Java languages. Some of the features of IoT SDKs are:

- APIs for reporting instrumentation data through custom, network, and error events
- Serialization of all events into JSON format before sending them to EUM Server
- API to plugin application HTTPS stack into SDK which will be used to communicate with EUM Server
- Business Transaction (BT) correlation
- Auto-disable when monitoring is turned off from the Controller UI

For languages other than C++ and Java, you can use the REST API to send custom, network request, and error events.

## IoT Monitoring UI

IoT Monitoring also provides the following features through the Controller UI:

- Dashboards for monitoring devices, network requests, as well as errors and exceptions
- Predefined and custom widgets to visualize reported application data
- Analytics for device application data through [AppDynamics Application Analytics](#)
- Configuration for naming and excluding network requests

## IoT Monitoring Workflow

The IoT SDKs and REST API enable you to report application data for a large number of different kinds of devices. This flexibility allows for a great deal of customization in the kind of the data you report, the quantity of data you report, and the timing of when the data is sent to the EUM Server.

To monitor customized data, you will need to better understand the data generated by your application and your device instrumentation. The goal of this page is to guide you through the process of understanding your application data, instrumenting your device application, and creating custom dashboards, so you can monitor device performance and activity.

It is recommended that you follow the steps below as a guideline to monitor your devices:

1. [Define your monitoring goals.](#)
2. [Identify what data needs to be captured.](#)
3. [Determine which events to use to report the data.](#)
4. [Obtain an EUM App Key.](#)
5. [Report events.](#)
6. [Monitor the availability, usage, and performance of your devices.](#)
7. [Build widgets.](#)
8. [Diagnose problems through filters.](#)
9. [Improve monitoring and troubleshoot.](#)

## Define Monitoring Goals

AppDynamics IoT Monitoring enables you to monitor the availability, performance, and usage of your device. You should devise a plan that prioritizes your monitoring goals, considering the needs of all stakeholders (developers, DevOps, business units), and lists the resources that need to be monitored. Also, you should consider who will be instrumenting the device application, who will be performing the monitoring, and the parties who should be notified when something goes wrong.

## Identify Data to Capture

Next, you'll need to analyze your device to identify what application data can be used to meet your monitoring goals. For example, if your device is a car, you would want to look at network requests to check device availability or report custom data for the wear and tear

on its components or road conditions.

## Determine Which Event Types to Use to Report Data

You report app data through the three types of events shown in the table below. Based on this information, map the most suitable event types to the data you identified and your monitoring goals.

Event Type	Monitoring Objective
Custom	Usage
	Business
Network Request	Availability
	Performance
	Usage
Error Events	Performance

## Obtain an EUM App Key

Follow the instructions provided in [Get an EUM App Key](#).

## Report Events

You can use the IoT SDKs or REST API to capture and report key data points using one of the event types. If you're using a language other than C/C++ or Java, use the IoT REST API, which gives you the flexibility to report data from any platform.

Follow the getting started instructions to learn how to install the SDK and instrument your IoT application:

- [Instrument Applications with the IoT C/C++ SDK](#)
- [Instrument Applications with the IoT Java SDK](#)

## Monitor Your IoT Application

To monitor your IoT applications, you will need to understand the function and purpose of the three IoT dashboards. See [Monitor Applications with the IoT Dashboards](#) for a brief introduction of each dashboard and usage instructions.

## Build Widgets Based on Your Data Models

Each dashboard has a set of predefined widgets for basic performance and activity monitoring. You will need to create your own widgets, however, to optimize the monitoring capabilities of **IoT Monitoring**. Based on the events you reported, create custom widgets to monitor device activity and performance. See the section [Build Custom Dashboard Widgets](#) to learn how to build widgets with the **IoT Widget Builder Wizard**.

## Diagnose Problems

When you do discover deviations from expected behavior, you can then add criteria to filter results. This enables you to narrow the results to certain devices or possible causes.

## Troubleshoot Issues and Improve Monitoring

For performance issues, you can open the device details to analyze errors and slow network requests. The device details allow you to trace the error event you are interested in and even download the stack trace. The developer can then use this information to determine the root cause of the problem and devise a possible solution. Going back to our point-to-sale device example, you might discover that most of the failed payments were due to network request errors.

You can use the usage information to improve your monitoring or the devices themselves. For example, an inventory management application indicates that several stores sell a smaller quantity of a certain product. It turns out that this product is often out of stock at these stores. You could improve the monitoring of your IoT app by using custom events to report when a product is sold out and then creating custom widgets to let you know which stores need to restock.

## IoT Analytics

All IoT data is processed and stored by the AppDynamics Platform Events Service. A separate product component, [AppDynamics Business iQ](#), has an event type called Connected Device Data. This event type is based on the same Events Service and uses the same data. Viewing this event type from the Analytics UI offers additional capabilities, including:

- Additional predefined widgets, such as the funnel widget
- ADQL for searching the data
- Creating custom widgets
- Manipulating multiple dashboard types
- Longer retention time for data storage

IoT Analytics does *not* require a license separate from the IoT Monitoring license.



To view Analytics data for connected devices, users need **Connected Devices Permissions** for the application (identified by a specific App Key).

---

## IoT Licenses and Limits

Currently, AppDynamics only offers the **IoT PRO** license. To learn about the license entitlements and restrictions, contact your AppDynamics account representative.