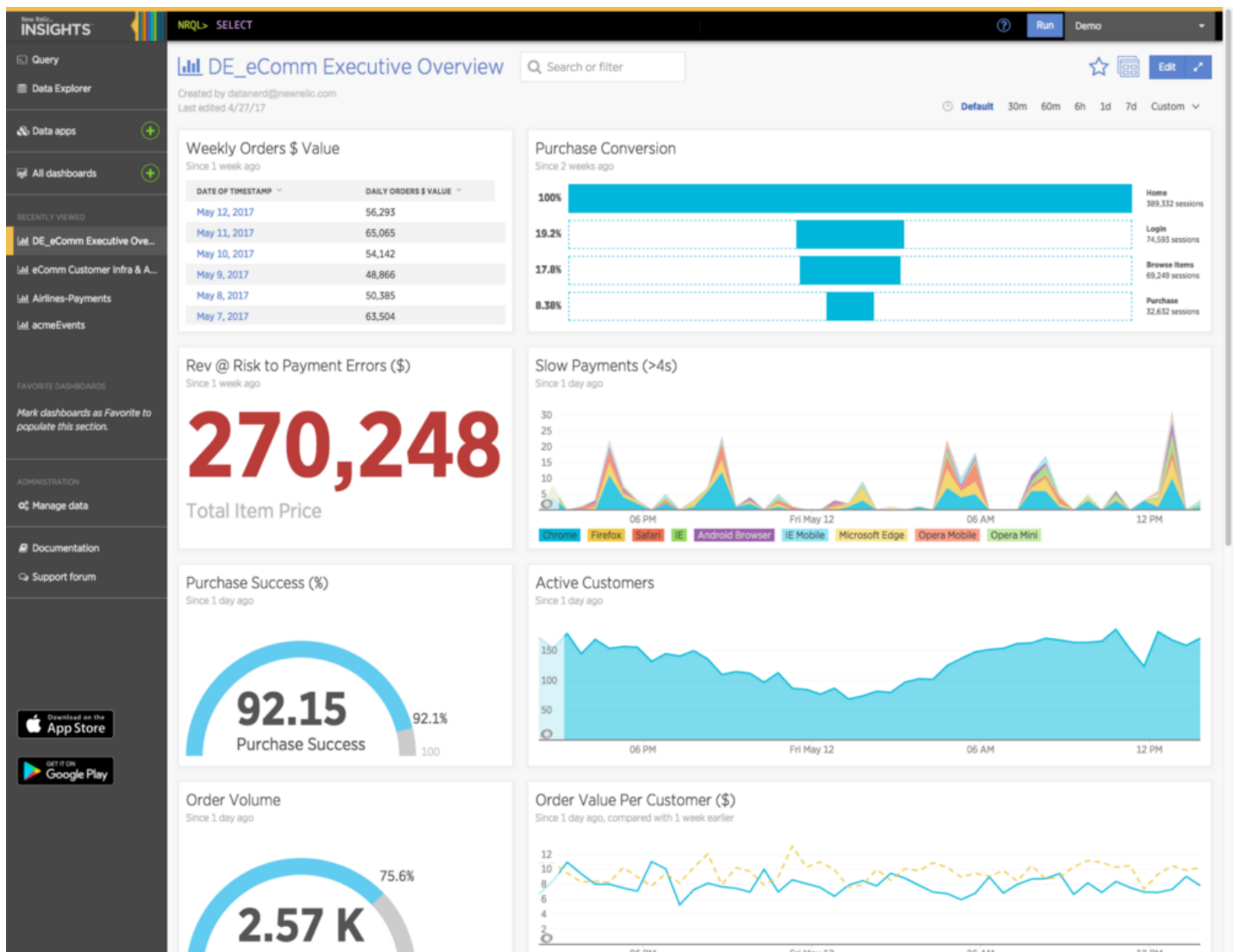


# New Relic — Enviando eventos personalizados



Wallace Silva May 25, 2020 · 7 min read

Nesse artigo, você verá como enviar eventos personalizados ao New Relic, utilizando Java, bem como consumi-los para criar Dashboards e alertas.



Informações oficiais da New Relic

Se você usa o New Relic em sua aplicação para capturar informações de APM como tempo de resposta, uso de memória, CPU, etc. E também insights, sabe que essas informações são muito úteis para controlarmos a saúde de nossa aplicação. Porém, além desses dados enviados automaticamente pelo Agente do New Relic, podemos fazer o envio de dados personalizados, como tempo de execução de determinado método, número de erros,

arquivos que estão sendo importados, usuários sendo criados, enfim, qualquer tipo de informação que desejar.

Podemos enviar qualquer tipo de informação sobre nossos eventos ao New Relic, e o envio de eventos personalizados é algo bem simples. Para enviarmos eventos personalizados, devemos utilizar a Event API do New Relic, que nos permite com apenas poucas linhas de código fazer o envio dos eventos. Após enviarmos nossos eventos, os mesmos já estarão disponíveis nos Insights para consultas, e podemos montar dashboards personalizadas com os eventos enviados e o uso do NRQL.

## Porque utilizar eventos personalizados ?

Os eventos personalizados podem nos ser uteis como desejamos enviar informações específicas para determinada aplicação, as quais não são enviadas por padrão pelo New Relic, para que possamos manipular os dados, montar gráficos, alertas entre outros.

## Vamos começar

Vamos mostrar hoje, como fazer o uso do Event API com a linguagem de programação JAVA (com Gradle), mas vale lembrar que o Event API também tem suporte a linguagens como .NET, Go, C, NodeJs, Python, PHP, entre outras.

O exemplo será o evento bem simples, onde serão enviadas as pessoas cadastradas em um API, para gerarmos uma dashborad para saber a porcentagem de pessoas por idade. Por mais que seja trivial, o intuito do exemplo é mostrar como é fácil fazer o uso do Event API, pode ser que sua aplicação necessite de mais informações, e dados mais complexos, porém, o envio das informações será feito da mesma maneira mostrada no exemplo.

Primeiramente, precisamos de uma aplicação com o java agent New Relic já configurado, caso você ainda não tenha o New Relic em sua aplicação, seria interessante ler esse outro artigo que escrevi explicando um pouco sobre o que é o New Relic e seu uso básico.

### New Relic — Monitorando suas aplicações Java

Hoje, vou mostrar a vocês como com apenas alguns passos podemos utilizar essa poderosa ferramenta que é o New Relic...

medium.com



Se o agent já esta configurado, e sua aplicação já esta enviando informações ao APM do New Relic, vamos colocar a mão na massa. Como foi dito acima, a implementação é bem simples. Primeiro, precisamos importar a biblioteca do New Relic Event API.

```
implementation 'com.newrelic.agent.java:newrelic-api:5.12.0'
```

Após adicionada a dependência, basta criarmos nossos dados e eventos.

Os eventos tem a estrutura de um Map, sendo sua chave uma **String** e seu valor um **Object**. Podemos também ter eventos em cadeia, vai da necessidade de cada projeto.

```
HashMap<String, Object> personMap = new HashMap<>();
```

Após criarmos o Map, agora, é só ir adicionando as informações a ele. No nosso Map, o valor da chave do Map, que é uma String, será utilizado para realizarmos as consultas no NRQL.

```
personMap.put("name", personModel.getName());  
personMap.put("years", personModel.getYears());
```

Como podemos ver, nosso evento terá as informações `name`, que será o nome da pessoa, e `years`, que será a idade da pessoa.

Agora, com nossos dados prontos, basta agora enviarmos nosso evento ao New Relic. Para isso, precisamos apenas fazer o uso do método `recordCustomEvent`, que faz o envio ao New Relic. Ele espera como parâmetro, o nome do nosso evento, seguido do Map que contém suas informações.

```
NewRelic.getAgent().getInsights().recordCustomEvent("PersonsEvent", personMap);
```

Como podemos ver acima, criamos um evento chamado **PersonsEvent** que tem como valores o nosso Map **personMap**.

Pronto! Simples né. Nossa aplicação já está fazendo o envio dos eventos personalizados ao New Relic.

## Informações Adicionais

Podemos também, fazer algumas configurações, como diminuir o número de eventos personalizados enviados por minuto, e também, desativar o envio de eventos personalizados.

Todas essas configurações são feitas no arquivo ***newrelic.yml***, que é responsável por conter todas as configurações do agente New Relic para nossa aplicação.

Diminuir o número de eventos personalizados.

```
custom_insights_events:  
  max_samples_stored: 3000
```

No exemplo de configuração acima, limitamos o número de eventos personalizados a 3K por minuto.

Podemos também, desativar o envio de mensagens personalizadas.

```
custom_insights_events:  
  enabled: false
```

## Ponto de atenção

*O New Relic envia suas informações normais a cada 5 segundos. O envio de eventos personalizados pode acabar sobrecarregando a memória do agente JAVA, por isso, é limitado a 10.000, caso esse número se exceda, os eventos podem ser perdidos. Também, informações com mais de 1 MB não serão gravadas.*

## Capturando nossos eventos

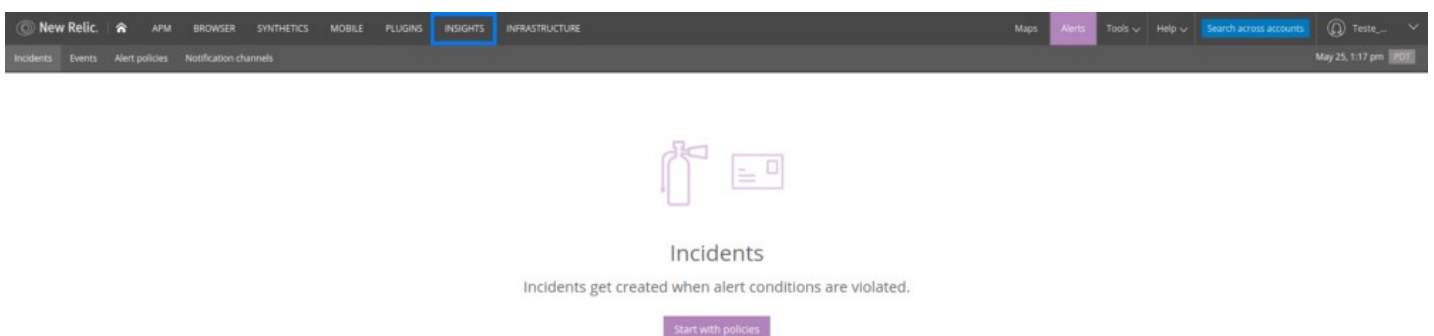
Agora que já estamos enviando nossos eventos personalizados ao New Relic, podemos montar nossas Dashboards e alertas. Vamos criar dois gráficos, um com a média de idades, e outro agrupando a porcentagem de usuários por idade. Também, faremos um alerta caso usuários com menos de 18 anos sejam cadastrados.

## Criando a Dashboard



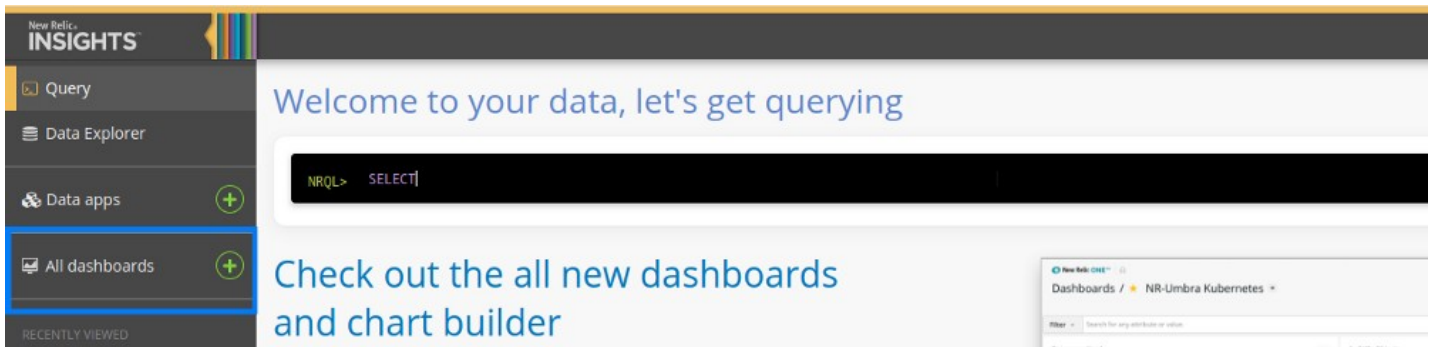
Nossos insights

Primeiro, vamos criar insights, devemos acessar a página de insights.



Seleciona a aba de insights

Após selecionar a aba de insights, no canto superior esquerdo, teremos a opção “All Dashboards”, vamos selecioná-la.



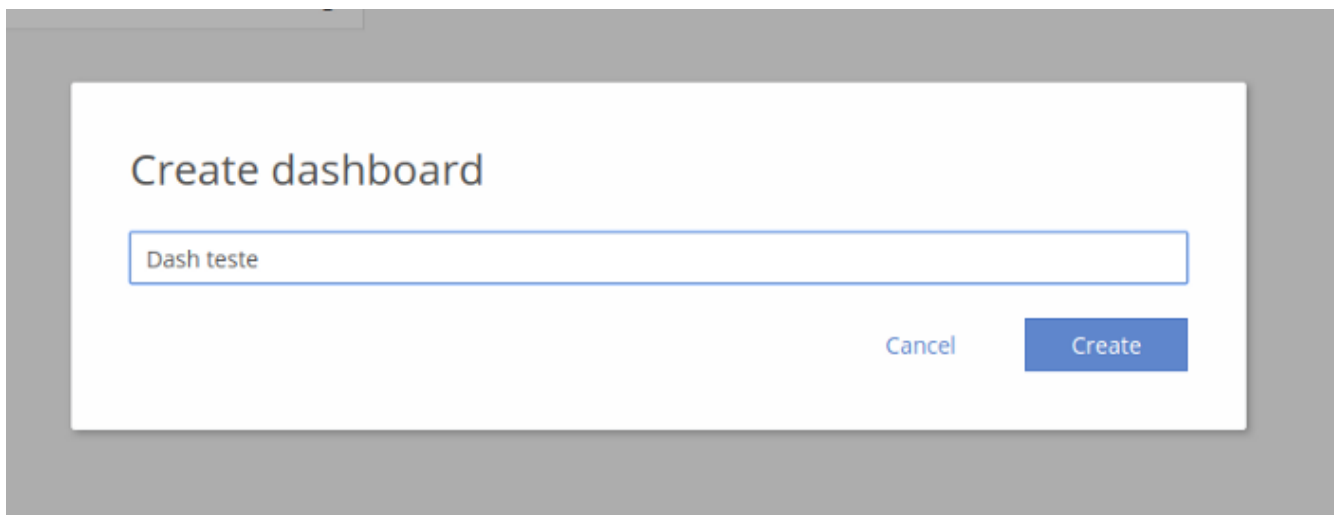
All Dashboards

Agora, no canto superior direito, temos a opção para criar nova dashboard “Create a Dashboard”.



Create a Dashboard

Agora, damos um nome a nossa Dashboard, e clicamos em “Create”

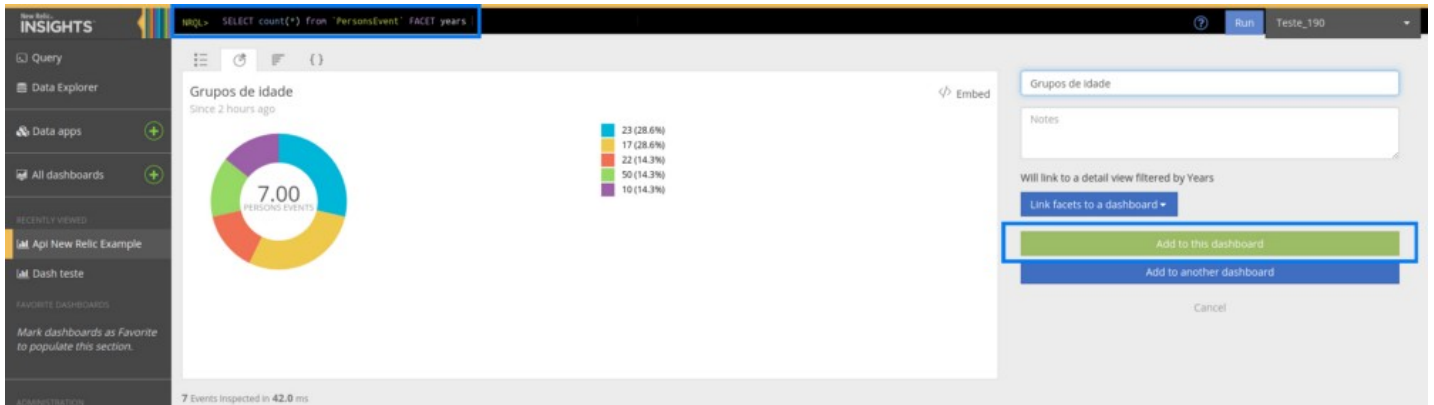


Nome da Dashboard

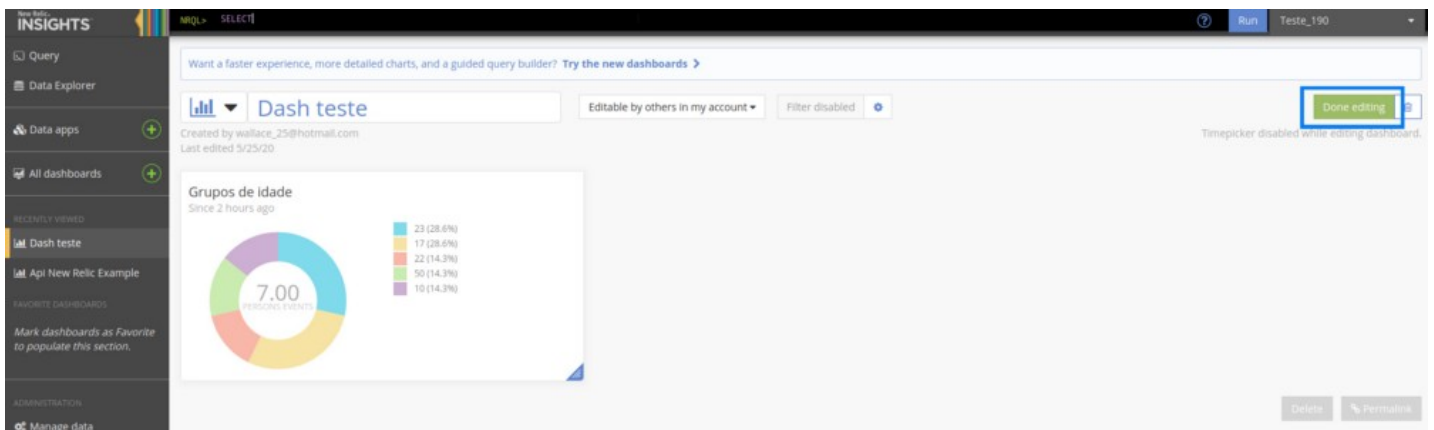
Pronto. Temos nossa dashboard criada. Agora, vamos criar alguns gráficos e adicionar a ela.

No topo da página, temos um campo escrito “NRQL>”, que é destinado para escrevermos nossas queries de consulta de nossos eventos. Podemos criar nossa query, e ao executar, será mostrado o resultado, podemos dar um titulo a ele, e depois, adicionar a nossa dash.

```
SELECT count(*) from `PersonsEvent` FACET years
```



Agora, basta clicarmos em “Done Editing”, e nossa dashboard esta pronta.



Sucesso! Temos uma Dashboard criada com dados de nossa evento personalizado. Crie novos gráficos, séries temporais, fique a vontade para explorar e aprender mais.

## Criando alertas

Além de termos representações visuais dos nossos dados de eventos customizados utilizando as Dashboards, podemos também programar alertas, para caso aconteça algo com nossa aplicação, o determinado dado foi enviado. Nesse exemplo, vamos criar um alerta, caso alguma pessoa com menos de 18 anos for cadastrada.

Primeiramente, devemos clicar na aba “Alerts”.



## Incidents

Incidents get created when alert conditions are violated.

Start with policies

Selecionar a opção “Alerts”

Em seguida, devemos selecionar as politicas de alerta “Alerts Polices”.



## Incidents

Incidents get created when alert conditions are violated.

Start with policies

Alert Policies

Agora, devemos criar uma nova policita de alerta. Devemos clicar em “Create Policy”.



## Create an alert policy

Create policy

Create policy



Agora, devemos criar um nome para nossa política de alerta, e também as preferências de incidente, e salvar.

Por padrão, já é criado o canal de notificação por e-mail, onde, o dono da conta do New Relic receberá um e-mail caso ocorra um incidente.

## Create alert policy

**ALERT POLICY NAME**  
Give your policy a concise and descriptive name.

**INCIDENT PREFERENCE**  
Specify how incidents should be created when conditions in this alert policy are violated. (Notifications are sent only when an incident opens, is acknowledged, and closes.)  

☒ **By policy**  
All violations **within this policy will be grouped** into a single incident; only one open incident at a time for this alert policy

☐ **By condition**  
All violations **within a condition in this policy will be grouped** into a single incident; only one incident at a time per alert condition

☐ **By condition and entity**  
An incident will open **every time an entity violates** a condition in this policy

[Learn more about incident preference](#)

**NOTIFICATION CHANNELS**  
You will receive notifications on open, acknowledged and closed incidents for this policy via the channel below. (You can add more channels later on the Alert policy page > Notification channels tab.)

## New Alert Policy


Agora, com a política de alerta criada, podemos criar uma nova condição de alerta. Clicando no botão “Crie a condição”.

Teste

Incident preference: **By policy** Delete this policy

0 Alert conditions 1 Notification channel

Last modified 1:18 pm by Wallace Silva



This policy doesn't have any conditions  
Alert conditions are the criteria for creating incidents.  
Notifications are sent when incidents are created.

[Create a condition](#)

## Create a condition

E em nossa condição, devemos selecionar a opção “NRLQ”, pois, se tratará de uma query

que trará as informações. E clicar em “Next, define thresholds”.

### Teste

New condition

ⓧ Cancel

1. Categorize

Select a product

APM

Browser

Mobile

Plugins

Synthetics

NRQL

Infrastructure

Next, define thresholds

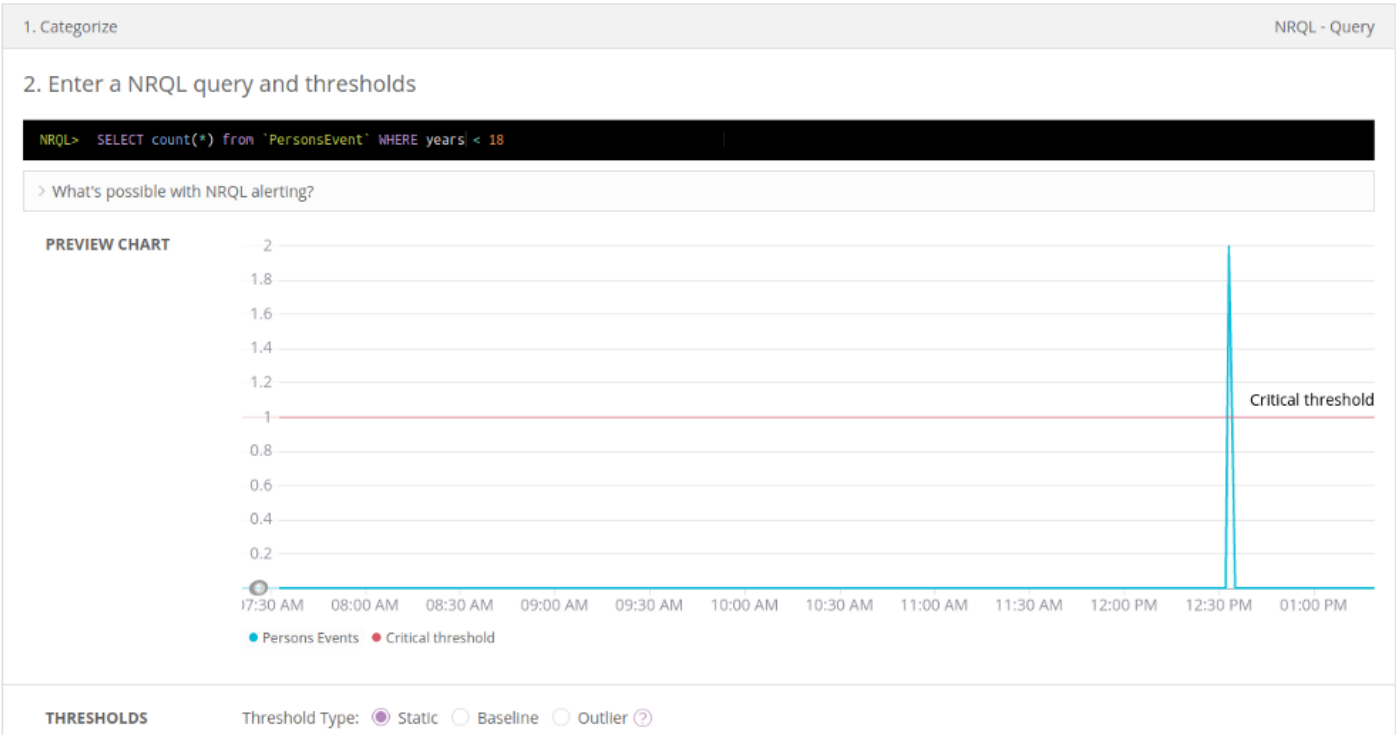
2. Enter a NRQL query and thresholds

### NRQL

Agora, essa é uma etapa muito importante, pois, é nela que colocaremos nossa query, e definiremos os limites do alerta, tempo para que o incidente seja fechado automaticamente, etc.

A query que será utilizada no exemplo é:

```
SELECT count(*) from `PersonsEvent` WHERE years < 18
```



When the

### Configurações do alerta

No exemplo acima, definimos o Threshold como 1 por pelo menos um minuto. Assim, caso 1 usuário com menos de 18 anos seja cadastrado em 5 minutos de análise, recebemos um e-mail, e um incidente será aberto. Agora, é só salvar, e nosso alerta será criado.

1 Alert condition

1 Notification channel

NRQL QUERY

Novo usuário menor de idade

Last modified 1:24 pm by Wallace Silva

NRQL> SELECT count(\*) from `PersonsEvent` WHERE years < 18

`PersonsEvent` query result is > 1 unit for at least 5 mins

Add a warning threshold

### Lista de alertas criados

É isso pessoal. Nesse artigo, vimos como é fácil criar eventos personalizados no New Relic. Bem como consumi-los para Insights e alertas. Por mais que foi uma implementação trivial, com o exemplo abortado, você conseguirá iniciar o uso e também, evoluir para uma implementação maior, pois o princípio é o mesmo.

O código fonte do exemplo mostrado pode ser encontrado em <https://github.com/wallacehenriquesilva/new-relic-api-test/blob/master/src/main/java/br/com/wallace/newrelic/api/services/PersonService.java>

Espero que tenham gostado, ótimos estudos!