New Relic®

Nerd Days
Fundamentals Track

New Relic®

These slides are available at http://nru.to/fun

Agenda (US)

9:00 am	Keynote: Grafana, Code for America Partnership; Lew and Bill co-presenter
10:00 am	Instrumenting your service using agents - Michael
11:00 am	Exploring your data using NRQL - Phil
12:00 pm	Lunch with Lew (AMA Session)
1:00 pm	Custom Instrumentation - Michael
2:00 pm	Alerts Best Practices - Phil
3:00 pm	True availability using Synthetics - Michael

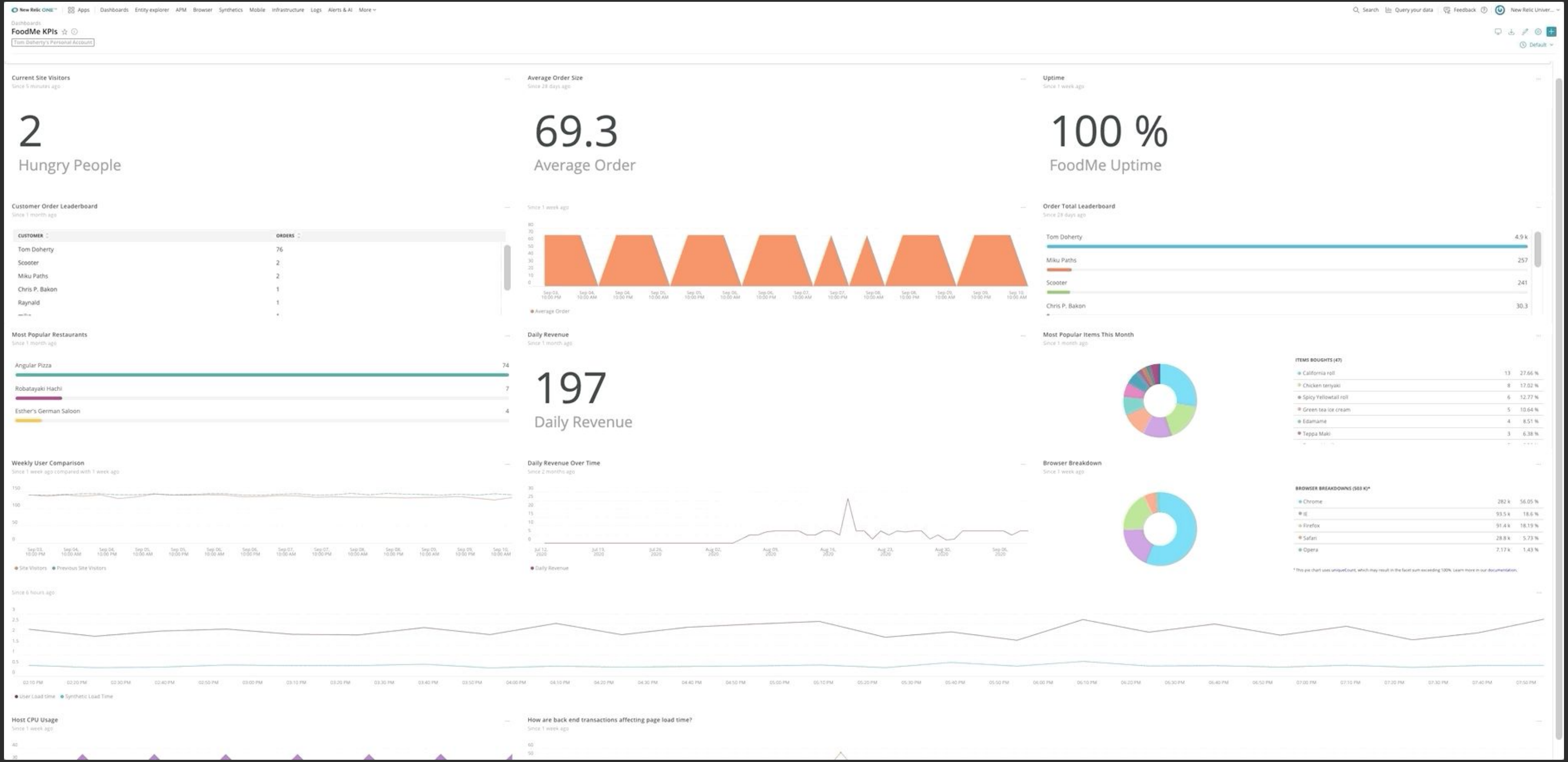
Agenda (EMEA)

9:00 am	**Keynote: Observability For Good** Lew Cirne
10:00 am	**Instrumenting your service using agents** Tom Doherty
11:00 am	**Exploring your data using NRQL** Liam Hurrell
12:00 pm	**Lunch Break**
12:30 pm	**Fireside Chat**
1:00 pm	**Custom Instrumentation** Tom Doherty
2:00 pm	**Alerts Best Practices** Liam Hurrell
3:00 pm	**True availability using Synthetics** Tom Doherty

Instrumenting your Service with Agents

Setting up Front and Back-end Monitoring

Dashboard Example



Dashboards

FoodMe KPIs

Tom Doherty's Personal Account

Current Site Visitors
Since 5 minutes ago

2
Hungry People

Average Order Size
Since 28 days ago

69.3
Average Order

Uptime
Since 1 week ago

100 %
FoodMe Uptime

Customer Order Leaderboard
Since 1 month ago

CUSTOMER	ORDERS
Tom Doherty	76
Scooter	2
Miku Paths	2
Chris P. Bakon	1
Raynald	1

Since 1 week ago

Order Total Leaderboard
Since 28 days ago

Tom Doherty	4.9 k
Miku Paths	257
Scooter	241
Chris P. Bakon	30.3

Most Popular Restaurants
Since 1 month ago

Angular Pizza	74
Robatayaki Hachi	7
Esther's German Saloon	4

Daily Revenue
Since 1 month ago

197
Daily Revenue

Most Popular Items This Month
Since 1 month ago

ITEMS BOUGHTS (47)		
● California roll	13	27.66 %
● Chicken teriyaki	8	17.02 %
● Spicy Yellowtail roll	6	12.77 %
● Green tea ice cream	5	10.64 %
● Edamame	4	8.51 %
● Teppa Maki	3	6.38 %

Weekly User Comparison
Since 1 week ago compared with 1 week ago

Daily Revenue Over Time
Since 2 months ago

Browser Breakdown
Since 1 week ago

BROWSER BREAKDOWNS (503 K)*		
● Chrome	282 k	56.05 %
● IE	93.5 k	18.6 %
● Firefox	91.4 k	18.19 %
● Safari	28.8 k	5.73 %
● Opera	7.17 k	1.43 %

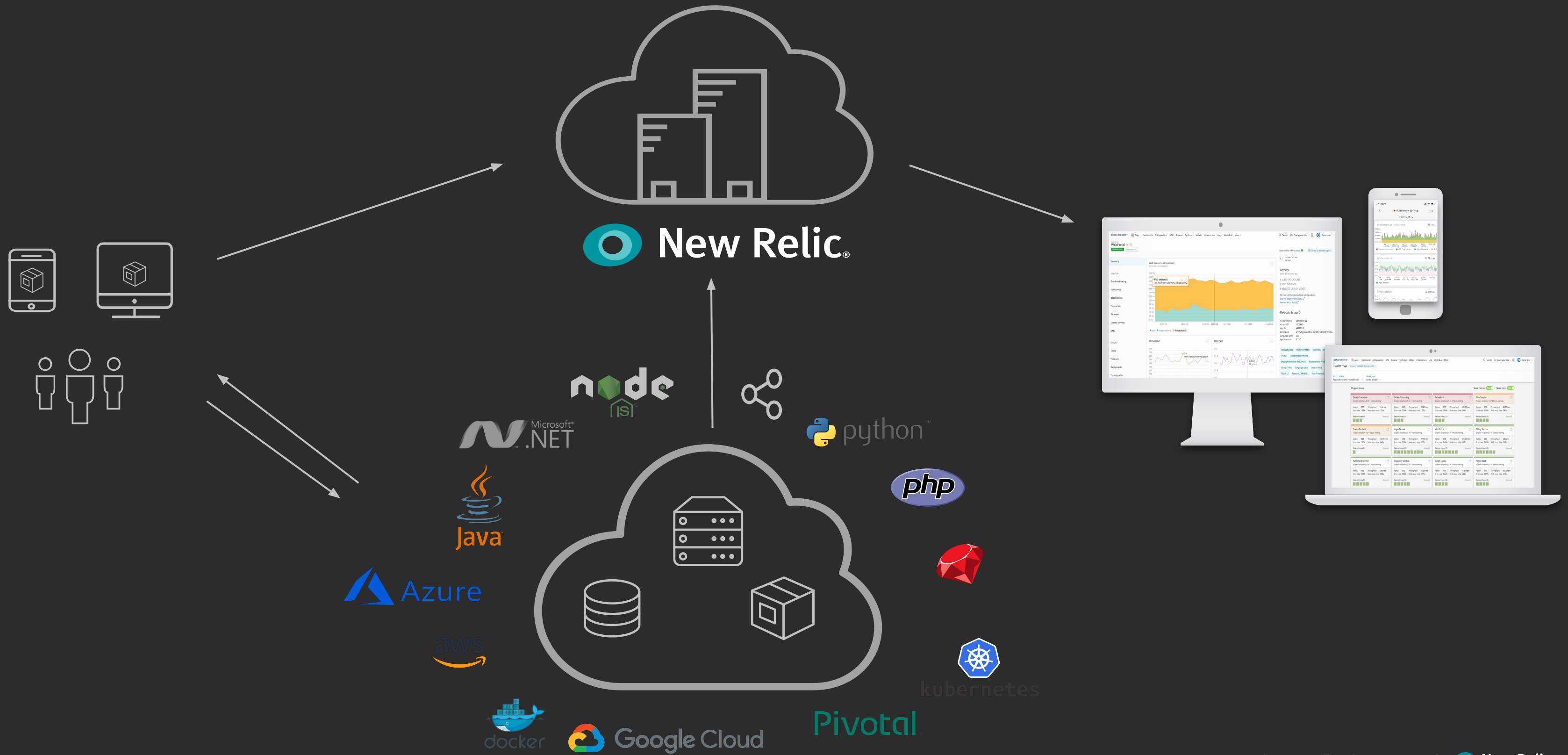
* This pie chart uses uniqueCount, which may result in the facet sum exceeding 100%. Learn more in our documentation.

Since 6 hours ago

Host CPU Usage
Since 1 week ago

How are back end transactions affecting page load time?
Since 1 week ago

Introduction to Agents



Let’s get an app reporting!

You will need:

1. A glitch account (free) www.glitch.com
2. A New Relic account www.newrelic.com
3. Your New Relic License key (which you can find in under the account settings drop down)

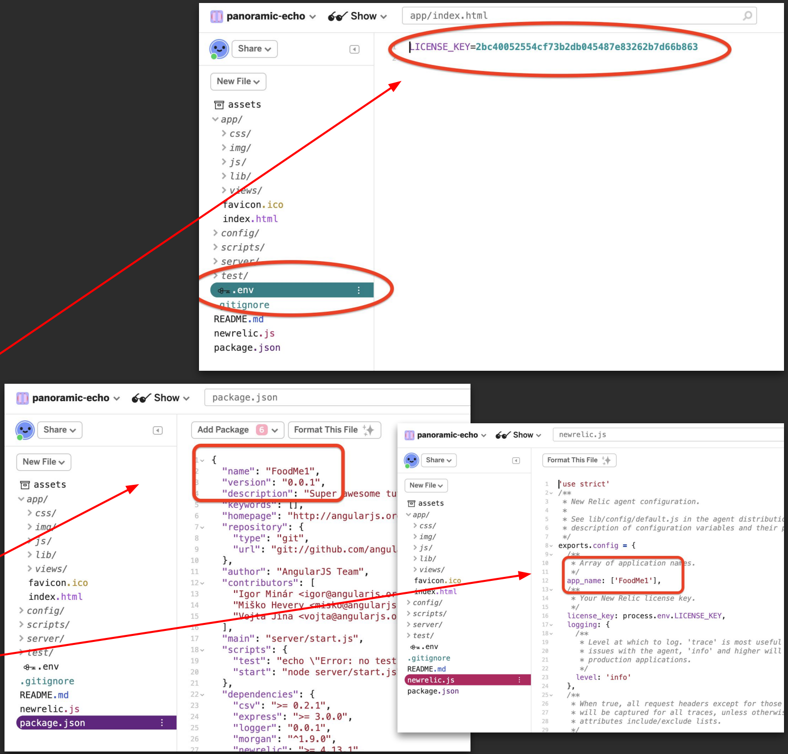
Let’s take 5 minutes to make sure you are ready to go.

5-7 mins

Lab: FoodMe step 1

Clone FoodMe into a Glitch account

1. Create an account at glitch.com
2. Visit the url: https://glitch.com/edit/#!/remix/nru-foodme to clone foodme into your Glitch account
3. Edit the `.env` file to contain your New Relic license key
4. Set the name of the application in the Newrelic.js file and package.json file

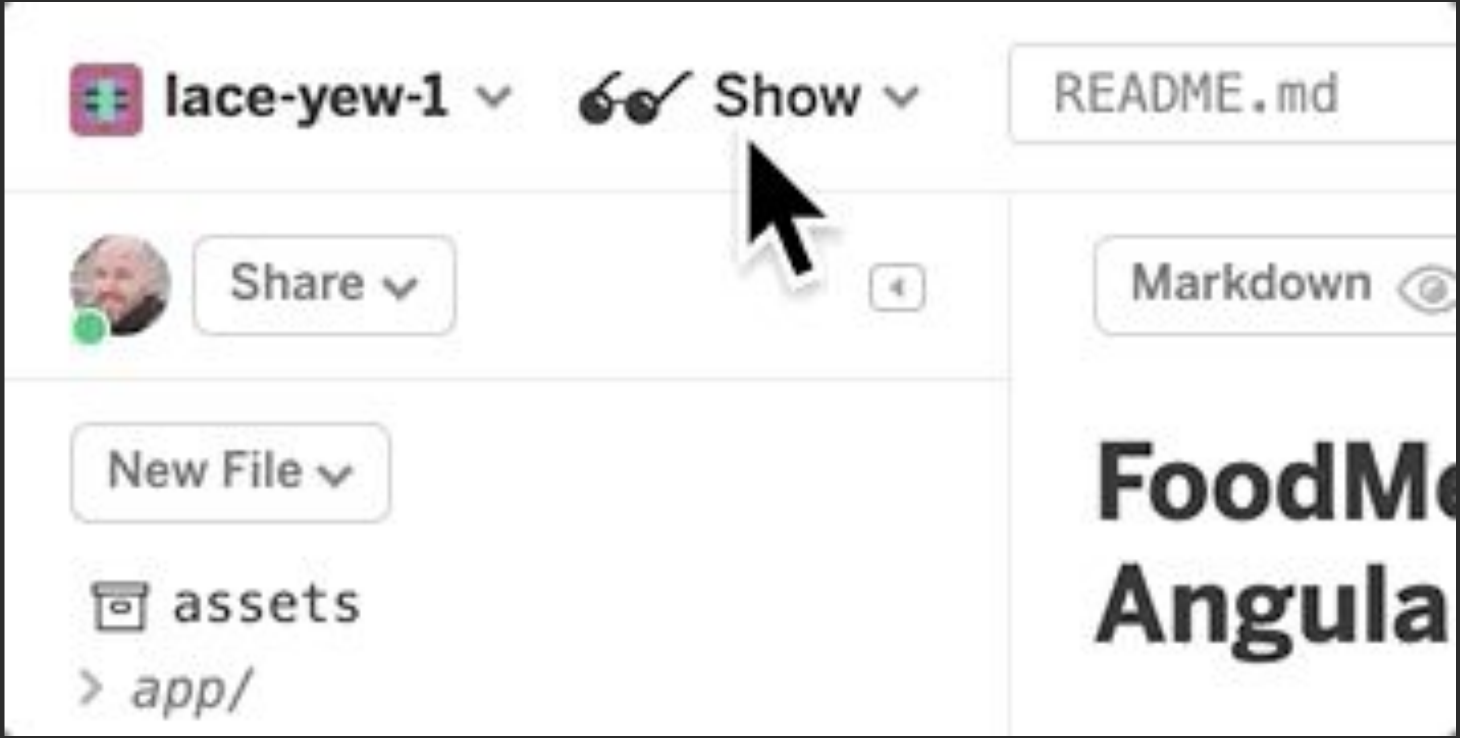


```
LICENSE_KEY=2bc40052554cf73b2db045487e83262b7d66b863
```

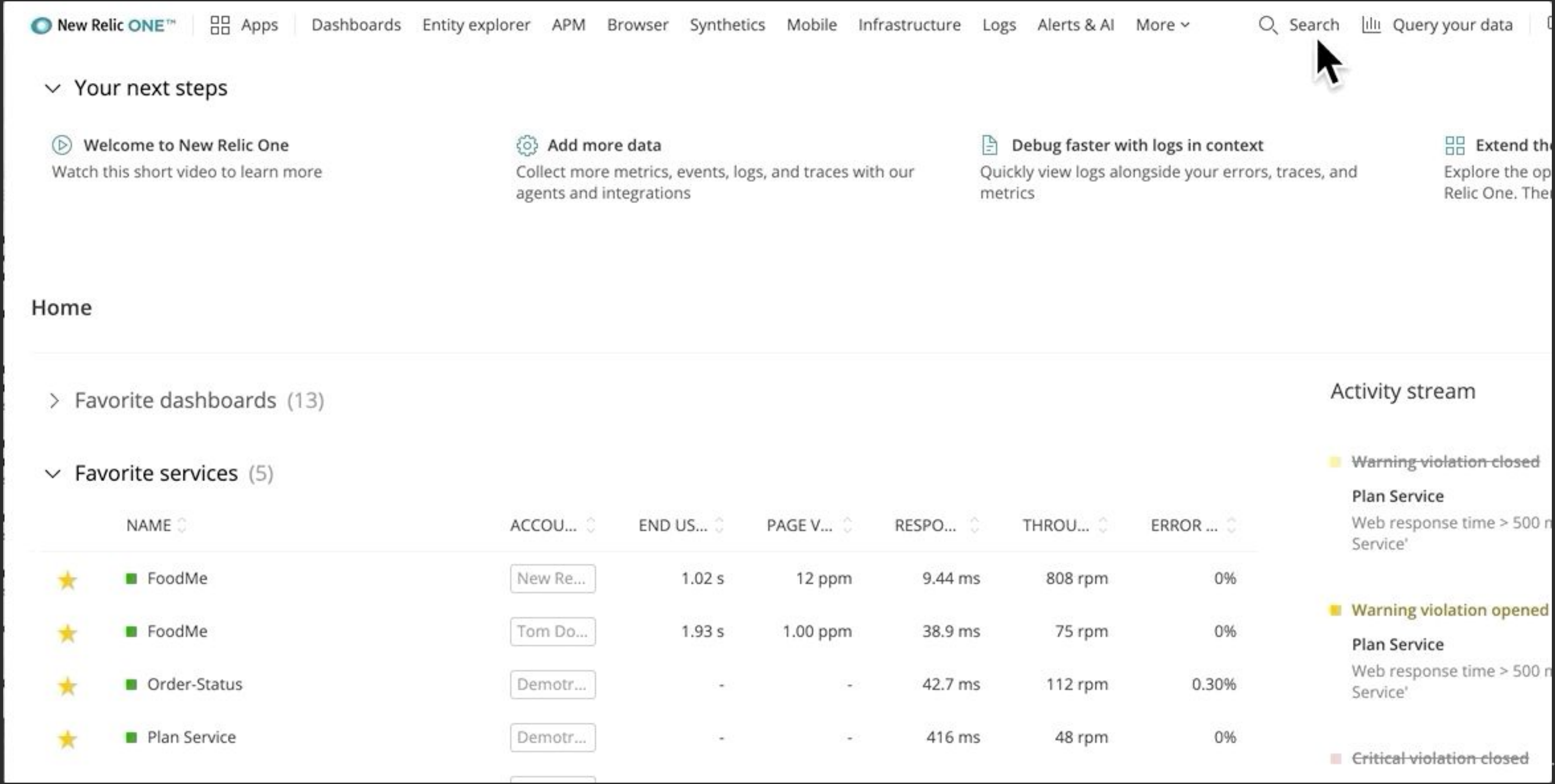
```
{
  "name": "FoodMe1",
  "version": "0.0.1",
  "description": "Super awesome tu
  "keywords": [],
  "homepage": "http://angularjs.or
  "repository": {
    "type": "git",
    "url": "git://github.com/angul
  },
  "author": "AngularJS Team",
  "contributors": [
    "Igor Minár <igor@angularjs.or
    "Miško Hevery <misko@angularjs
    "Vojta Jina <vojta@angularjs.o
  ],
  "main": "server/start.js",
  "scripts": {
    "test": "echo \"Error: no test
    "start": "node server/start.js
  },
  "dependencies": {
    "csv": ">= 0.2.1",
    "express": ">= 3.0.0",
    "logger": "0.0.1",
    "morgan": "^1.9.0",
    "newrelic": ">= 4.13.1"
```

```
'use strict'
/**
 * New Relic agent configuration.
 *
 * See lib/config/default.js in the agent distributi
 * description of configuration variables and their
 */
exports.config = {
  /**
   * Array of application names.
   */
  app_name: ['FoodMe1'],
  /**
   * Your New Relic license key.
   */
  license_key: process.env.LICENSE_KEY,
  logging: {
    /**
     * Level at which to log. 'trace' is most useful
     * issues with the agent, 'info' and higher will
     * production applications.
     */
    level: 'info'
  },
  /**
   * When true, all request headers except for those
   * will be captured for all traces, unless otherwis
   * attributes include/exclude lists.
   */
```

Run the Application to Generate Some Data by Placing Some Food Orders



Head to one.newrelic.com and search for your Service



Your next steps

Welcome to New Relic One
Watch this short video to learn more

Add more data
Collect more metrics, events, logs, and traces with our agents and integrations

Debug faster with logs in context
Quickly view logs alongside your errors, traces, and metrics

Extend th
Explore the op
Relic One. The

Home

Favorite dashboards (13)

Favorite services (5)

	NAME	ACCOU...	END US...	PAGE V...	RESPO...	THROU...	ERROR ...
★	🟩 FoodMe	New Re...	1.02 s	12 ppm	9.44 ms	808 rpm	0%
★	🟩 FoodMe	Tom Do...	1.93 s	1.00 ppm	38.9 ms	75 rpm	0%
★	🟩 Order-Status	Demotr...	-	-	42.7 ms	112 rpm	0.30%
★	🟩 Plan Service	Demotr...	-	-	416 ms	48 rpm	0%

Activity stream

~~Warning violation closed~~
Plan Service
Web response time > 500 r
Service'

Warning violation opened
Plan Service
Web response time > 500 r
Service'

~~Critical violation closed~~

5-7 mins

Lab: FoodMe step 2

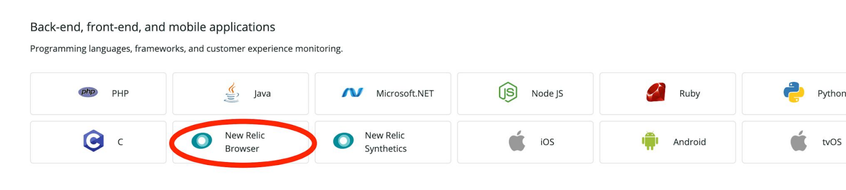
Deploying the Browser agent

1. In the “Add More Data” menu’, choose Browser, select your account, choose Copy/paste Javascript code and copy the Browser agent snippet.

2. Paste the snippet into the \`app/index.html\` file (just after the head)

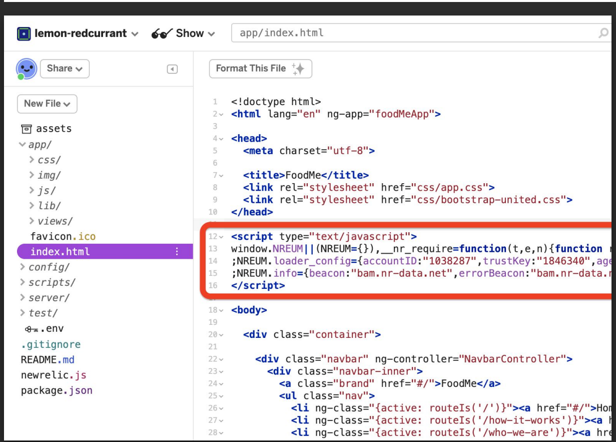
3. In the “Name Your App” section, select “Yes” and search for your Application name

4. Click the Show > In a New Window button in the upper-left corner of the screen to open the app in a new browser window.



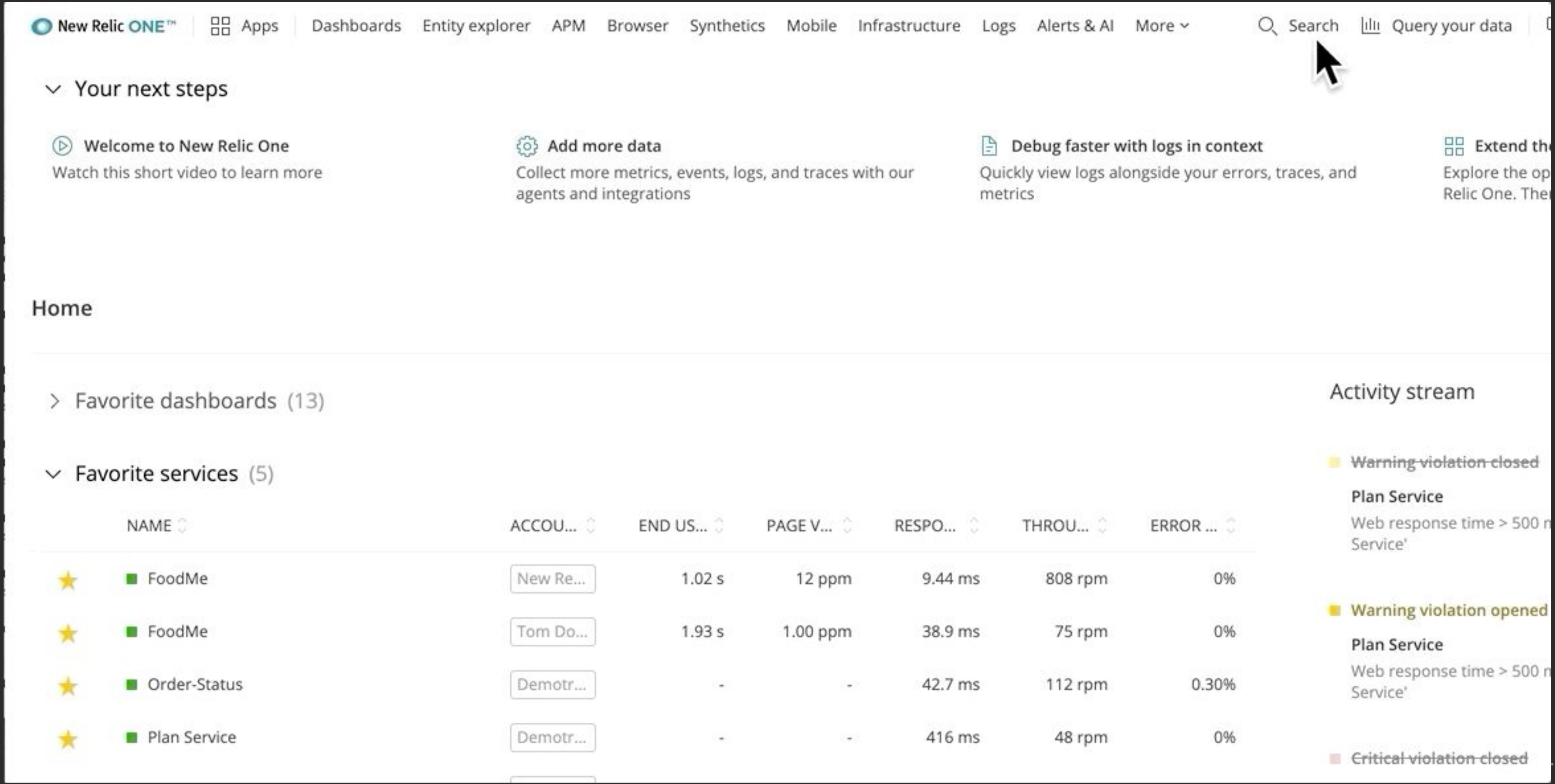
Back-end, front-end, and mobile applications

Programming languages, frameworks, and customer experience monitoring.



```
<!doctype html>
<html lang="en" ng-app="foodMeApp">

<head>
  <meta charset="utf-8">

  <title>FoodMe</title>
  <link rel="stylesheet" href="css/app.css">
  <link rel="stylesheet" href="css/bootstrap-united.css">
</head>

<script type="text/javascript">
window.NREUM||(NREUM={}),__nr_require=function(t,e,n){function
;NREUM.loader_config={accountID:"1038287",trustKey:"1846340",age
;NREUM.info={beacon:"bam.nr-data.net",errorBeacon:"bam.nr-data.
</script>

<body>

  <div class="container">

    <div class="navbar" ng-controller="NavbarController">
      <div class="navbar-inner">
        <a class="brand" href="#/">FoodMe</a>
        <ul class="nav">
          <li ng-class="{active: routeIs('/')}"><a href="#/">Hon
          <li ng-class="{active: routeIs('/how-it-works')}"><a h
          <li ng-class="{active: routeIs('/who-we-are')}"><a hre
```

Head to one.newrelic.com and search for your Browser Application



Your next steps

Welcome to New Relic One
Watch this short video to learn more

Add more data
Collect more metrics, events, logs, and traces with our agents and integrations

Debug faster with logs in context
Quickly view logs alongside your errors, traces, and metrics

Extend th
Explore the op
Relic One. The

Home

Favorite dashboards (13)

Favorite services (5)

	NAME	ACCOU...	END US...	PAGE V...	RESPO...	THROU...	ERROR ...
★	🟩 FoodMe	New Re...	1.02 s	12 ppm	9.44 ms	808 rpm	0%
★	🟩 FoodMe	Tom Do...	1.93 s	1.00 ppm	38.9 ms	75 rpm	0%
★	🟩 Order-Status	Demotr...	-	-	42.7 ms	112 rpm	0.30%
★	🟩 Plan Service	Demotr...	-	-	416 ms	48 rpm	0%

Activity stream

~~Warning violation closed~~
Plan Service
Web response time > 500 r
Service'

Warning violation opened
Plan Service
Web response time > 500 r
Service'

~~Critical violation closed~~

Exploring your Data with NRQL

Querying and Building Visualisations

New Relic.

Training account credentials

Log out of other New Relic accounts (or open private browser window)

URL: https://one.newrelic.com/

Email: learn@newrelicuniversity.com

Password: pointless-meeting

Telemetry data in NRDB

- Different data types can be queried with a consistent SQL like query language: NRQL
- Event-type data is collected from across your stack
- Can include custom events
- Each event collects metadata (attributes) about the event

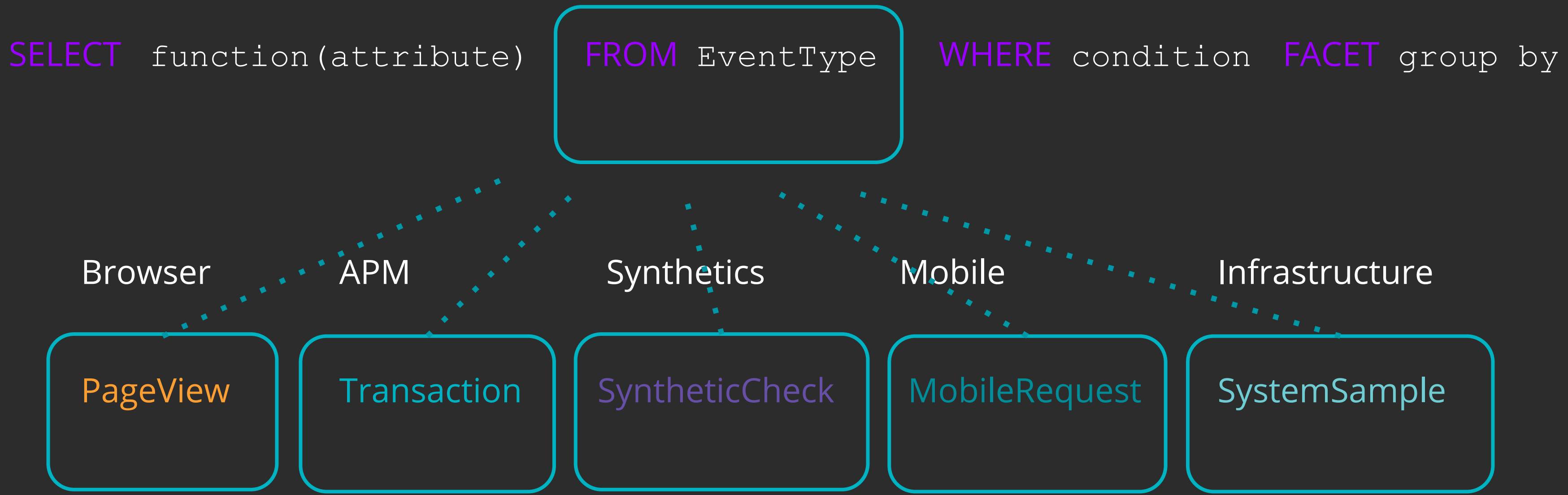
Example

Event type: Transaction

Metadata: Transaction Type - Response Code - Request URI - Duration - App ID

NRQL Syntax

Build queries around your chosen data type



```
SELECT function(attribute) FROM EventType WHERE condition FACET group by
```

Browser

PageView

APM

Transaction

Synthetics

SyntheticCheck

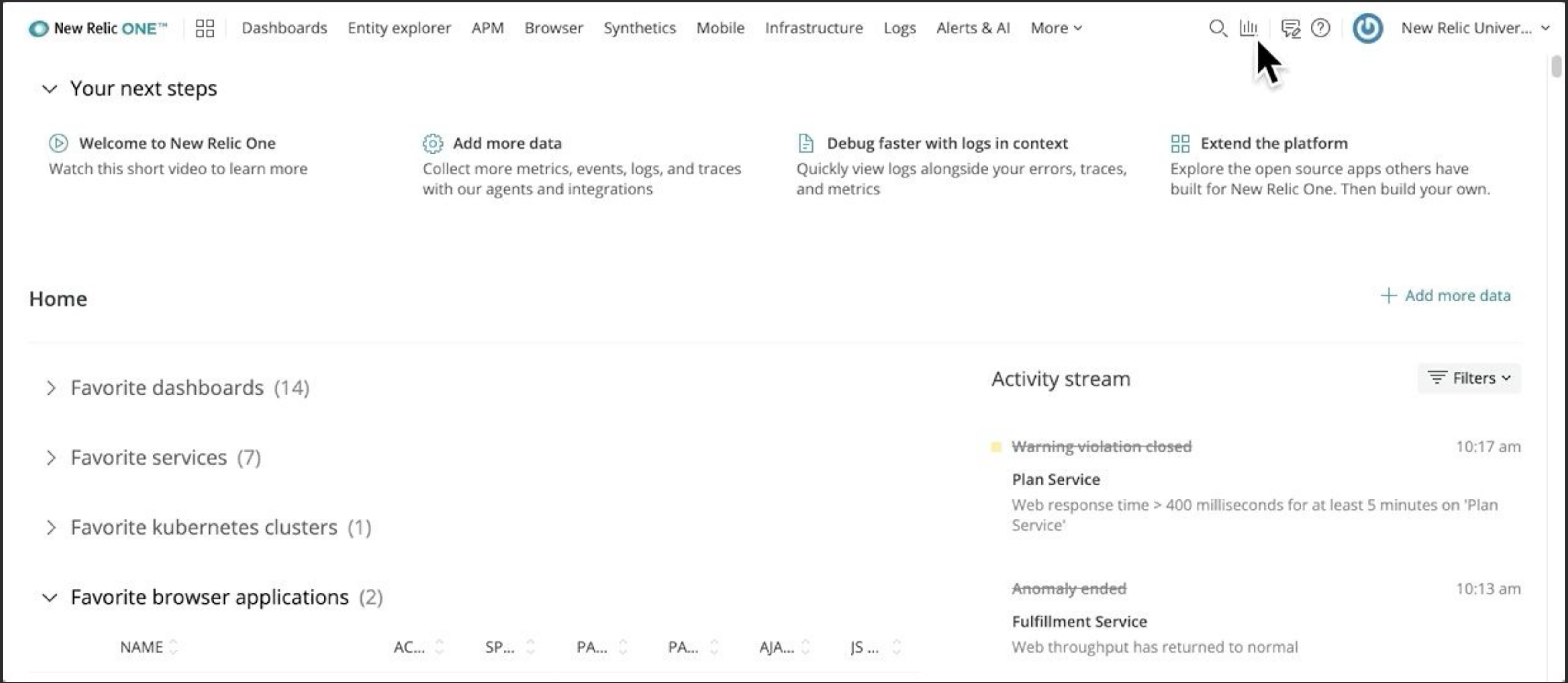
Mobile

MobileRequest

Infrastructure

SystemSample

Let’s explore our data!



Your next steps

Welcome to New Relic One
Watch this short video to learn more

Add more data
Collect more metrics, events, logs, and traces with our agents and integrations

Debug faster with logs in context
Quickly view logs alongside your errors, traces, and metrics

Extend the platform
Explore the open source apps others have built for New Relic One. Then build your own.

Home

+ Add more data

Favorite dashboards (14)

Favorite services (7)

Favorite kubernetes clusters (1)

Favorite browser applications (2)

NAME	AC...	SP...	PA...	PA...	AJA...	JS ...

Activity stream

Filters

~~Warning violation closed~~ 10:17 am

Plan Service

Web response time > 400 milliseconds for at least 5 minutes on 'Plan Service'

~~Anomaly ended~~ 10:13 am

Fulfillment Service

Web throughput has returned to normal

10 mins

Lab: Data Explorer

Explore and build queries quickly and easily with the Data Explorer

1. Build 3 queries by selecting the following options.
   Try different visualisation (Bar chart, Pie Chart etc)

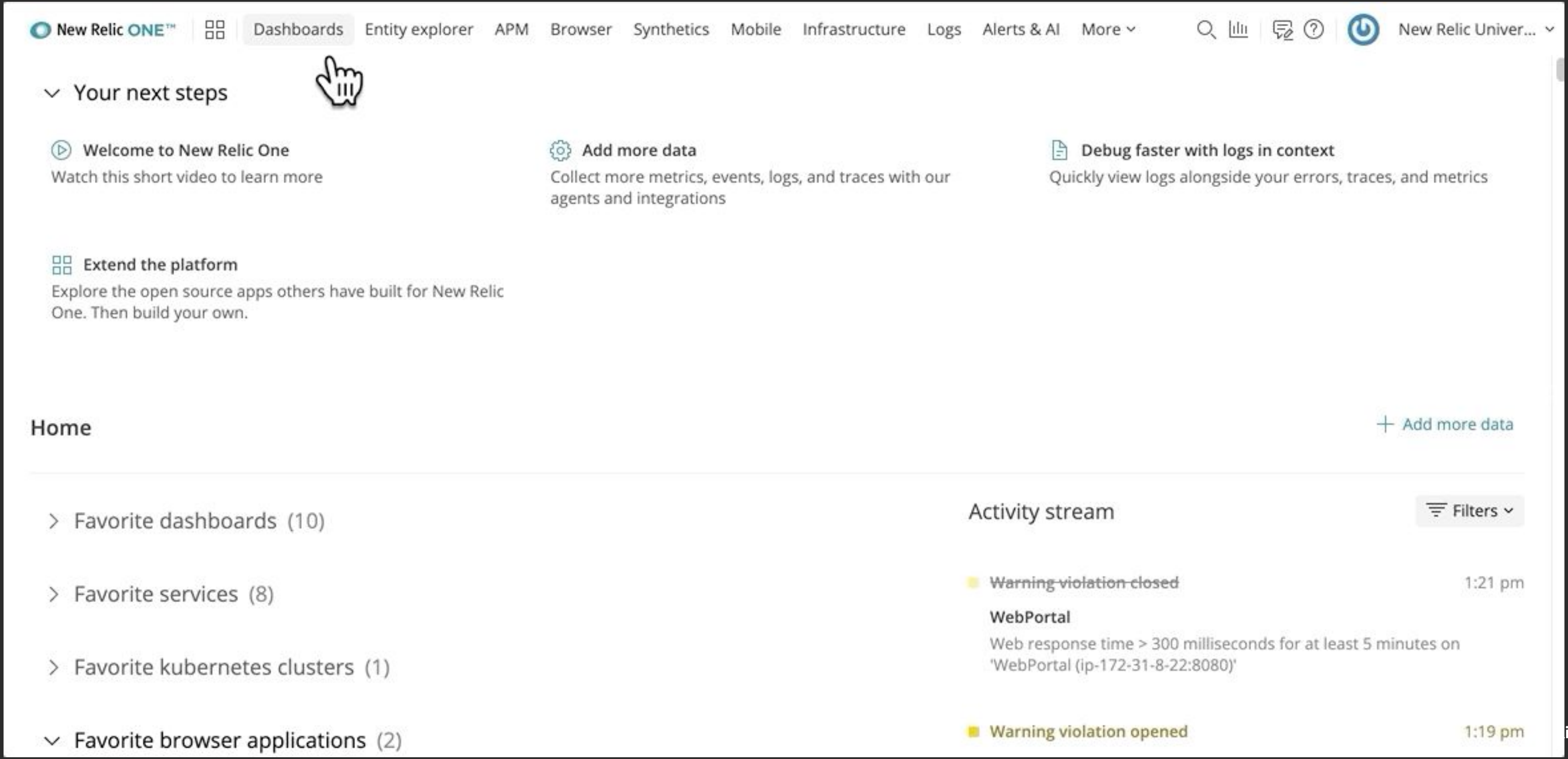
   - Event type: **Transaction**, Plot **Count (*)** Dimension **Name**
   - Event type: **Transaction**, Plot **Duration Max** Dimension **Name**
   - Event type: **PageView**, Plot **FirstContentfulPaint Max** Dimension **PageUrl**

2. Try the *RAW Data* option and explore the attributes for the **PageView** Event Type

3. Mouse over the NRQL query you have built, and click to 'Edit in Query Builder'

   - Try changing the time period. Example: `SINCE 1 day ago`
   - Try changing the FACET. Example: `userAgentName`

Let’s create a dashboard!



Your next steps

Welcome to New Relic One
Watch this short video to learn more

Add more data
Collect more metrics, events, logs, and traces with our agents and integrations

Debug faster with logs in context
Quickly view logs alongside your errors, traces, and metrics

Extend the platform
Explore the open source apps others have built for New Relic One. Then build your own.

Home

+ Add more data

Favorite dashboards (10)

Favorite services (8)

Favorite kubernetes clusters (1)

Favorite browser applications (2)

Activity stream

Filters

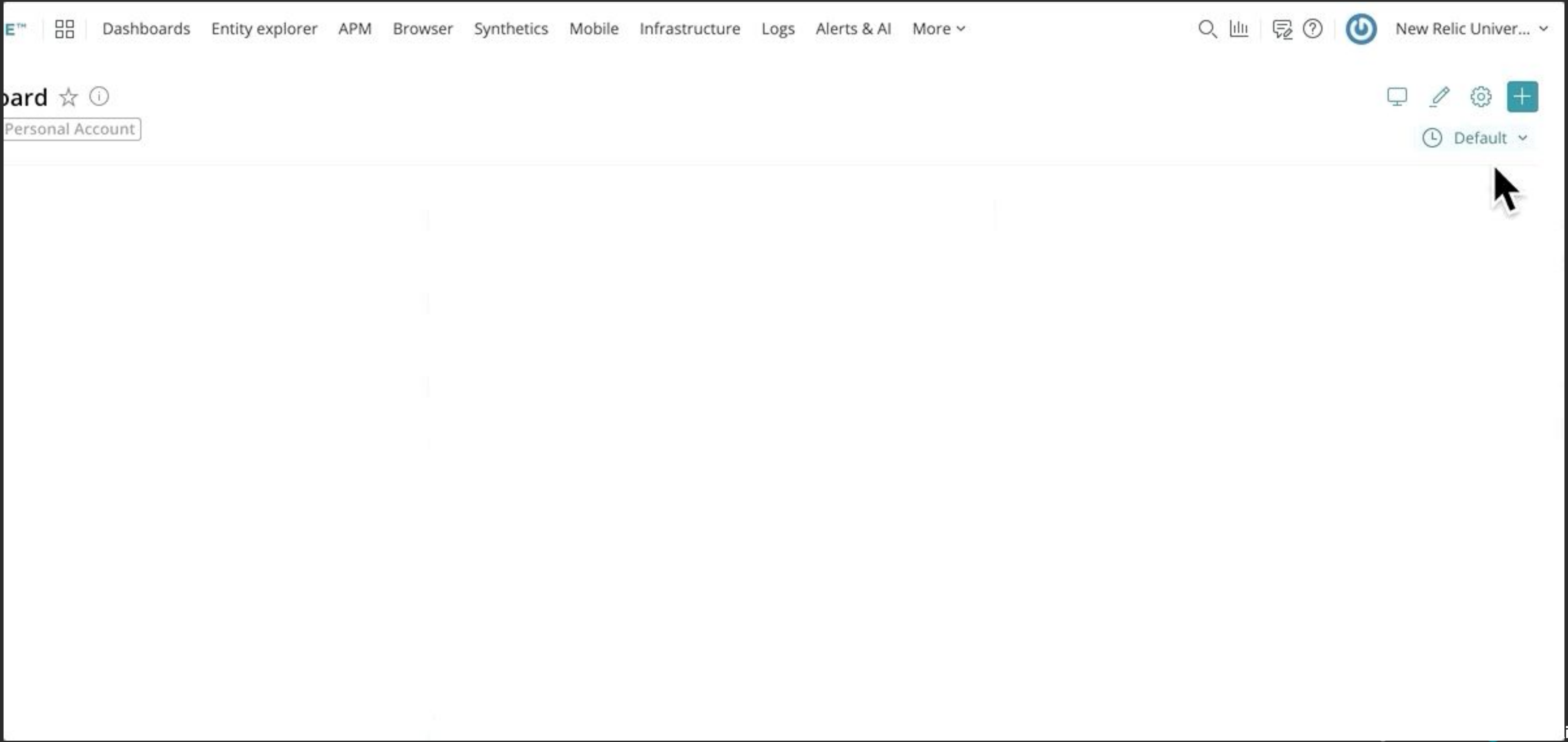
~~Warning violation closed~~ 1:21 pm

WebPortal

Web response time > 300 milliseconds for at least 5 minutes on 'WebPortal (ip-172-31-8-22:8080)'

Warning violation opened 1:19 pm

Let’s write some queries!



Lab: NRQL Query Builder - Your first query

1 Type this query in the Chart Builder and view the results:

```
SELECT * FROM Transaction
```

Aggregate your data

Average() Max() Min() Count()

```
SELECT count(*) FROM Transaction
```

Throughput for my application

```
SELECT max(duration) FROM Transaction
```

Slowest transactions for my application

Group your data

FACET

```
SELECT count(*) FROM Transaction  FACET name
```

Transactions grouped by name

```
SELECT count(*) FROM Transaction  FACET httpResponseCode
```

Transactions grouped by response codes

Query part of your data

WHERE

```
SELECT average(duration) FROM Transaction  WHERE appName = 'FoodMe'
```

Average duration for a specific application

```
SELECT count(*) FROM Transaction  WHERE httpResponseCode != '200'
FACET httpResponseCode
```

Count all transactions with different response codes

10 mins

Lab: Starting Queries

Add these Transaction queries to your dashboard:

1 FROM ___________ SELECT _______________

From the `Transaction` *event type, count all* `count(*)` *the Transactions*

2 FROM ___________ SELECT _____________ FACET ____________

From `Transaction`*, count all the Transactions grouped* `FACET` *by* `name`

3 FROM ____________ SELECT ___________WHERE _______ FACET _____

Max duration of Transactions where the application name is 'FoodMe' grouped by name

10 mins

Lab: Starting Queries

Add these PageView queries to your dashboard:

1 FROM ____________ SELECT ________________

Count all `Count(*)` *the* `PageView`*s on your site*

2 FROM ____________ SELECT ______________ FACET _____________

Count all `Pageviews` *grouped by country*

3 FROM _____________ SELECT ____________WHERE ________ FACET ______

Select Max duration of Pageviews from a specific country grouped by Page url

10 mins

Lab: More Advanced Queries

Add these queries to your dashboard:

1. Show the 3 slowest transactions

2. Compare active unique sessions on your site in the last 5 minutes with 5 minutes earlier

3. Show a pie chart of the worst page loads grouped by browser (userAgentName), city, and country

Custom Instrumentation

Adding Custom Data

New Relic.

Adding Custom Data

1. We will add custom attributes on the server side to gather details about our food order

```
newrelic.addCustomAttributes()
```

2. Then we will add a PageAction event on the client side, which will also enable us to capture attributes about food orders

```
newrelic.addPageAction(‘orderItem’)
```

5-7 mins

Lab: FoodMe step 3

Adding Custom Data

Modify Server side - index.js

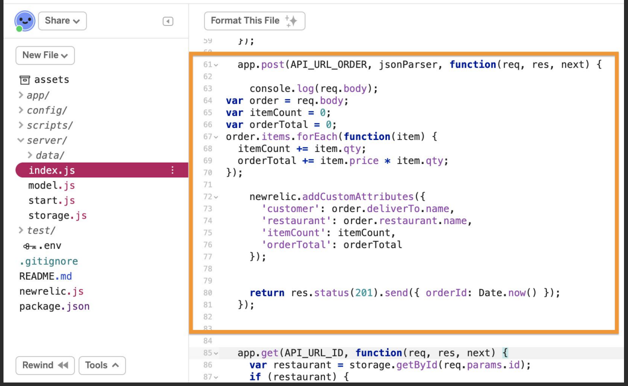
Attributes: Customer, restaurant, itemCount, orderTotal

- First we iterate through each item in our order and use the quantity and price for our **itemCount** and **orderTotal**.

```
 var order = req.body;
   var itemCount = 0;
   var orderTotal = 0;
   order.items.forEach(function(item){
   itemCount += item.qty;
   orderTotal += item.price * item.qty;
   });
```

- Then we need to send this to New Relic. We just need 1 line of code. Which is our agent API call: **newrelic.addCustomAttributes()**

```
newrelic.addCustomAttributes({
'customer': order.deliveryTo.name,
'restaurant': order.restaurant.name,
'itemCount': itemCount,
'orderTotal': orderTotal
});
```



```
  });

  app.post(API_URL_ORDER, jsonParser, function(req, res, next) {

    console.log(req.body);
var order = req.body;
var itemCount = 0;
var orderTotal = 0;
order.items.forEach(function(item) {
  itemCount += item.qty;
  orderTotal += item.price * item.qty;
});

    newrelic.addCustomAttributes({
      'customer': order.deliverTo.name,
      'restaurant': order.restaurant.name,
      'itemCount': itemCount,
      'orderTotal': orderTotal
    });


    return res.status(201).send({ orderId: Date.now() });
  });



  app.get(API_URL_ID, function(req, res, next) {
    var restaurant = storage.getById(req.params.id);
    if (restaurant) {
```

5-7 mins

Lab: FoodMe step 4

Adding a PageAction Event

Modify Client side - app/js/services/cart.js

- We want to answer questions like ‘Which items are most popular?’ - which we could do using our attributes from the Transaction event. But we could also gather this info from the client side; So for each item in the cart, we want to create an event which captures the restaurant name, the item name and the quantity.

- We are recording this using an API call from our JavaScript API to create a PageAction.

Find the submitOrder function and add the teal lines of code after:

```
self.submitOrder = function() {
    if (self.items.length){
```

```
self.items.forEach(
function(item) {
newrelic.addPageAction(‘orderItem’), {
restaurant: self.restaurant.name,
item: item.name,
qty: item.qty
});
});
```

And before the post to Order API:

```
return $http.post('/api/order', {
```

Time for lunch (not really)

Now you're collecting custom data from your FoodMe application you can start to query that data to understand your customers and your business as well as the application performance. But first, let's generate some data.

1 Place some orders in your FoodMe application to generate data

2 Feel free to share around your app to crowdsource some data

15-20 mins

Lab: Query your FoodMe App

Create a Dashboard in New Relic One to answer these questions for your online FoodMe App

Transaction event type

- **Total orders in past 24 hours?**
  - (Count all Transactions WHERE Appname is ‘FoodMe’ AND name LIKE '%api/order')
- **Which are the most popular Restaurants?**
  - (as above) ...FACET restaurant
- **Which are the most active Customers?**
  - (Count orders group by customer) ..FACET customer
- **Which Customers have the highest ‘orderTotal’?**
  - (sum the custom attribute ‘orderTotal’ group by customer)
  - SELECT sum(orderTotal)...
- **What’s the average order size?**
  - SELECT average(orderTotal)
- **What’s the average order size over time?** (TIMESERIES) for the past 24 hours

PageAction event type

- **Which are the most popular items?** (Sum the total quantity of orders from front end using FROM **pageAction** event type WHERE actionName = ‘orderitem’ and group by item)
  - SELECT sum(qty) ... WHERE actionName = 'orderItem' .. .

Alerts Best Practices

Building Alert Policies that don’t keep you up at night

New Relic.

Alerting - Best Practises

Getting started:
https://blog.newrelic.com/product-news/alerts-getting-started-best-practices/

Effective Alerting Strategy:
https://newrelic.com/resource/effective-alerting-guide

1 Make Alerts actionable

2 Alert as close to the source as possible

3 Test to start
or use Baseline Metrics & adjust over time

4 Avoid too low thresholds (cause alert fatigue)

5 Disable alert conditions, for example while testing others

5-7 mins

Lab: Create Alert Policies based on established thresholds

You have set up SLOs and SLA thresholds. Your CTO wants you to be proactive and add alerts that will notify you if the FoodMe application has degraded performance and is not meeting agreed levels (SLAs)

1 **Create a policy** to contain a series of conditions that monitor overall user experience.

Use the following naming convention for the Policy: *'service-name'* User-Experience *'your-team-name'*

2 As it will have a range of conditions across the stack, you want to group any issues that are triggered together so that you can review and resolve them in one place if performance degrades.

Which incident preference option should you choose?

____

3 Add the following conditions:

- **APM > Backend Apdex drop below 0.85 for 5 minutes**. Add a warning of your choice
- **Browser > Pageload time (2 seconds)** Add a warning of your choice

Note: Use explicit condition names for the following conditions eg. 'service-name' 'condition' 'value exceeded'

5-7 mins

Lab: Alerting on a new application or unpredictable behaviour

The front end developers have started making use of AJAX calls to keep the client-server connections to a minimum and improve perceived performance.

The team is still experimenting with AJAX and want to keep doing so. There is no clear performance threshold yet but they still want to be notified when the AJAX response time deviates from the historical average.

1 Add an alert condition to your “User-Experience” policy you have set up already.

Select the **Browser category > ‘Metric Baseline’** condition type.

Choose **AJAX Response Time > Upper Only > for 3 minutes**

- choose a sensitivity setting with the **slider**

NRQL Alerts

Query results

- Queries must return a number
- The alert condition works by evaluating that returned number against thresholds you set

Threshold Types

Static

- Condition based on the value returned

Baseline

- self-adjusting condition - based on the past behavior of the monitored values

Outlier

- Looks for values that are outliers from the same metric time series as its peers

5-7 mins

Lab: Create Alert Policies using NRQL

Query your Custom Data and get alerted on it

You want to be prepared for busy patterns of sales on your FoodMe delivery site. You decide to use the NRQL Alert to trigger an incident when the average order value exceeds a set value.

1 Add an alert condition to the User-Experience policy you have set up already.

Select the NRQL category, and use the following NRQL query:

```
SELECT average(orderTotal) AS 'Average
Order' FROM Transaction WHERE appName =
'my-FoodMe-app-name' AND name LIKE
'%api/order' FACET appName, name
```

2 Use the Static Threshold type:

- when query returns a value > above
- 50 > at least once in > 3 minutes

Add the condition name:

- Order Value Average > 50 FoodMe App

Set Violation time limit:

- 300 seconds

10 mins

Lab: Notify the right team

Use the Webhook Notification Channel

Now that the alert conditions have been configured, it's time to set up the notification channels.
For this specific Alert Policy, the Operations team needs to integrate the alerts with their Third Party Tool so you plan to use the **Webhook** channel option.

1 Choose an Operations `'teamName'` *(family-friendly !)*

2 Create a webhook notification channel that will send notifications to https://webhook.nru.to/

- Alerts > Notification Channels > + New Notification
- Channel Channel Type > Webhook
- Channel Name: 'My *Operations team name*' Webhook
- Base Url: https://webhook.nru.to/

3 Customise the webhook JSON Payload by adding an additional key value pair for your `'teamName'` :

- `"teamName": "MyOpsTeamName-FoodMe",`
- Save / Create Channel
- Send a 'Notification Test' - Check for '200'
- Check the Third Party Tool (indicator page!) https://alert-indicator.nru.to/

4 Start adding Food Delivery orders of over $50 value on your Glitch site

- Does your indicator change colour??
- Once it changes, Query your data and see what the average order value is. You can try out higher values...

True Availability with Synthetics

Going beyond uptime with Scripted Browsers

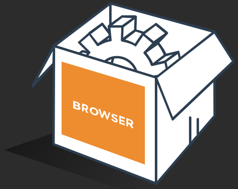
New Relic.

RUM and Synthetics work best together

Proactive
Find problems before users see them

Consistent, predictable
Good for alerting

Synthetics



Browser

Wide variety
browsers devices locations

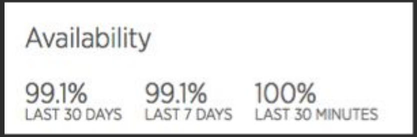
Valuable real-world performance
debugging info

Synthetics

Use cases

Ping Monitor

Track up/down availability



Availability

99.1%
LAST 30 DAYS

99.1%
LAST 7 DAYS

100%
LAST 30 MINUTES

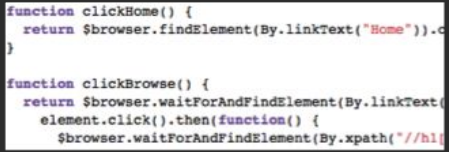
Simple Browser Monitor

Analyze page load performance

Screenshot of result

Scripted Browser Monitor

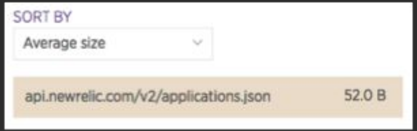
Advanced tests for complex workflows



```
function clickHome() {
  return $browser.findElement(By.linkText("Home")).c
}

function clickBrowse() {
  return $browser.waitForAndFindElement(By.linkText(
    element.click().then(function() {
      $browser.waitForAndFindElement(By.xpath("//h1[
```

API Monitor

Ensure key APIs are operational



SORT BY

Average size

api.newrelic.com/v2/applications.json 52.0 B

20-30 mins

Lab: Synthetics

Creating Monitors

- Create Ping Monitor for a site of your choice
- Create a Simple Browser check for a site of your choice
- Create a Scripted Browser Monitor to verify users can access your FoodMe app and place a food order (use Selenium Chrome extension and New Relic Synthetics exporter or Katalon Script recorder or copy/paste an example from the docs and write your own script)

Alerting and Dashboarding your Monitors

- Add a chart with the uptime of your Ping monitor to a dashboard
- Add a chart which allows you to compare user load time vs synthetic load time for your Simple Browser
- Create an alert condition to notify you if the success rate of your Scripted Browser drops below 95% (`SELECT percentage(count(*), WHERE result =...`)

Thank You

learn.newrelic.com

New Relic.