

---

# **django-newrelic-plugin-agent Documentation**

***Release 0.2.0***

**Ryan Bales**

November 29, 2016



<b>1</b>	<b>Usage</b>	<b>3</b>
1.1	Configuration . . . . .	3
1.2	Add a Plugin Component . . . . .	3
1.3	Adding Metric Values to be Sent to New Relic . . . . .	3



This project provides an interface for publishing arbitrary metrics to New Relic.



## 1.1 Configuration

Register Django settings

```
>>> NEWRELIC_PLUGIN_AGENT = {  
...     # number of milliseconds between retries to obtain lock on metric timeslice  
...     # lock acquired for atomic metric computation/publication operations  
...     'TIMESLICE_LOCK_RETRY_DELAY_MS': 1000,  
...     # license key for newrelic account  
...     'NEWRELIC_LICENSE_KEY': 'keyboardcat',  
... }
```

Set metric push task to run as often as you like. New Relic recommends 60 seconds.

```
>>> from djcelery.models import PeriodicTask, IntervalSchedule  
>>> interval_schedule = IntervalSchedule.from_schedule(schedule(timedelta(seconds=60)))  
>>> interval_schedule.save()  
>>> PeriodicTask.object.create(  
...     name='PushMetricTimeslice', interval=interval_schedule, enabled=True,  
...     task='newrelic_plugin_agent.tasks.PushMetricTimeslicesTask')
```

## 1.2 Add a Plugin Component

A component can be thought of as an aspect of your application/stack that you want to monitor.

```
>>> from newrelic_plugin_agent.models import NewRelicComponent  
>>> # create a component  
>>> component = NewRelicComponent.objects.create(  
...     name='AccountActivity', guid='com.your_company_name.account_activity')
```

## 1.3 Adding Metric Values to be Sent to New Relic

Push a metric value by forking an async job to add the value to the metric queue

```
>>> from newrelic_plugin_agent.tasks import PushMetricValueTask  
>>> # created a new account, track frequency  
>>> PushMetricValueTask.delay(component, 'AccountsCreated', 1)
```