Passerelle - Development #36215

jobs, pouvoir renseigner une date/heure minimale d'exécution

18 septembre 2019 10:10 - Frédéric Péters

Statut:	Fermé	**Début:**	18 septembre 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

(parce qu'après avoir refait cron(1), autant refaire at(1) aussi)

En fait l'idée c'est pouvoir planifier un rejeu, qui ne soit pas immédiat.

Révisions associées

Révision 8f39fbc3 - 01 octobre 2019 10:31 - Benjamin Dauvergne

misc: remove unused statement in jobs() (#36215)

Should have been removed in a39595b.

Révision 95904dba - 01 octobre 2019 10:31 - Benjamin Dauvergne

misc: prevent locking all jobs (#36215)

.first() does list(qs)[:1] which will select all target jobs,
we must add a LIMIT 1 before .first() to lock only the job we are
looking for. Ordering is necessary as .first() will do an
.order_by('pk') on an unordered queryset to get a deterministic result
in all cases and ordering a sliced queryset is not possible.

Révision d629c50e - 01 octobre 2019 10:31 - Benjamin Dauvergne

misc: add after_timestamp to run Job later (#36215)

The after_timestamp can be set:
- when adding the job with:

```
self.add_job(..., after_timestamp=datetime(...))
```

- when skipping a job with:

```
raise SkipJob(after_timestamp=datetime(...))
```

Historique

#1 - 21 septembre 2019 11:17 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#2 - 21 septembre 2019 11:17 - Benjamin Dauvergne

- Fichier 0001-misc-prevent-locking-all-jobs-36215.patch ajouté

- Fichier 0002-misc-add-after_timestamp-to-run-Job-later-36215.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#3 - 21 septembre 2019 12:34 - Frédéric Péters

Est-ce que le after_timestamp pourrait aussi accepter un timedelta (qui se ferait par rapport à .now()) ?

#4 - 21 septembre 2019 14:04 - Benjamin Dauvergne

- Fichier 0001-misc-prevent-locking-all-jobs-36215.patch ajouté

- Fichier 0002-misc-add-after_timestamp-to-run-Job-later-36215.patch ajouté

Interdiff:

```
diff --git a/passerelle/base/models.py b/passerelle/base/models.py
index e38efe3b..d866addf 100644
--- a/passerelle/base/models.py
+++ b/passerelle/base/models.py
@@ -523,7 +523,7 @@ class BaseResource(models.Model):
                 getattr(self, job.method_name)(**job.parameters)
             except SkipJob as e:
                 job.status = 'registered'
-                job.after_timestamp = e.after_timestamp
+                job.set_after_timestamp(e.after_timestamp)
                 skipped_jobs.append(job.id)
             except Exception as e:
                 self.handle_job_error(job, sys.exc_info())
@@ -538,8 +538,8 @@ class BaseResource(models.Model):
                   resource_pk=self.pk,
                   method_name=method_name,
                   natural_id=natural_id,
-                  after_timestamp=after_timestamp,
                   parameters=kwargs)
+        job.set_after_timestamp(after_timestamp)
         job.save()
         return job

@@ -679,6 +679,16 @@ class Job(models.Model):
             )
     status_details = jsonfield.JSONField(default={})

+    def set_after_timestamp(self, value):
+        if isinstance(value, datetime.datetime):
+            self.after_timestamp = value
+        elif isinstance(value, six.integer_types + (float,)):
+            self.after_timestamp = timezone.now() + datetime.timedelta(seconds=value)
+        elif isinstance(value, datetime.timedelta):
+            self.after_timestamp = timezone.now() + value
+        else:
+            self.after_timestamp = value
+

 class ResourceLog(models.Model):
     timestamp = models.DateTimeField(auto_now_add=True)
diff --git a/tests/test_jobs.py b/tests/test_jobs.py
index e932909f..c79d6413 100644
--- a/tests/test_jobs.py
+++ b/tests/test_jobs.py
@@ -1,5 +1,6 @@
 # -*- coding: utf-8 -*-

+import datetime
 import os

 import mock
@@ -40,15 +41,46 @@ def test_jobs(mocked_get, app, base_adresse, freezer):
     assert Job.objects.get(id=job.id).status == 'registered'

     # use after_timestamp with SkipJob
-    mocked_get.side_effect = None
     freezer.move_to('2019-01-01 00:00:00')
-    Job.objects.filter(id=job.id).update(after_timestamp='2019-01-02 00:00:00')
+    mocked_get.side_effect = SkipJob(after_timestamp='2019-01-02 00:00:00')
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'registered'
+    mocked_get.side_effect = None
+    freezer.move_to('2019-01-01 12:00:00')
     base_adresse.jobs()
     assert Job.objects.get(id=job.id).status == 'registered'
     freezer.move_to('2019-01-02 01:00:00')
     base_adresse.jobs()
     assert Job.objects.get(id=job.id).status == 'completed'

+    # use after_timestamp with SkipJob and seconds
+    job = base_adresse.add_job('update_streets_data')
```

```
+    freezer.move_to('2019-01-01 00:00:00')
+    mocked_get.side_effect = SkipJob(after_timestamp=3600)
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'registered'
+    mocked_get.side_effect = None
+    freezer.move_to('2019-01-01 00:30:00')
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'registered'
+    freezer.move_to('2019-01-01 01:01:00')
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'completed'
+
+    # use after_timestamp with SkipJob and timedelta
+    job = base_adresse.add_job('update_streets_data')
+    freezer.move_to('2019-01-01 00:00:00')
+    mocked_get.side_effect = SkipJob(after_timestamp=datetime.timedelta(seconds=3600))
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'registered'
+    mocked_get.side_effect = None
+    freezer.move_to('2019-01-01 00:30:00')
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'registered'
+    freezer.move_to('2019-01-01 01:01:00')
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'completed'
+
     # use after_timestamp with add_job
     freezer.move_to('2019-01-01 00:00:00')
     job = base_adresse.add_job('update_streets_data', after_timestamp='2019-01-02 00:00:00')
@@ -58,6 +90,24 @@ def test_jobs(mocked_get, app, base_adresse, freezer):
     base_adresse.jobs()
     assert Job.objects.get(id=job.id).status == 'completed'

+    # use after_timestamp with add_job and seconds
+    freezer.move_to('2019-01-01 00:00:00')
+    job = base_adresse.add_job('update_streets_data', after_timestamp=3600)
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'registered'
+    freezer.move_to('2019-01-01 01:01:00')
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'completed'
+
+    # use after_timestamp with add_job and seconds
+    freezer.move_to('2019-01-01 00:00:00')
+    job = base_adresse.add_job('update_streets_data', after_timestamp=datetime.timedelta(seconds=3600))
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'registered'
+    freezer.move_to('2019-01-01 01:01:00')
+    base_adresse.jobs()
+    assert Job.objects.get(id=job.id).status == 'completed'
+
     # don't run jobs if connector is down
     StreetModel.objects.all().delete()
     with mock.patch('passerelle.apps.base_adresse.models.BaseAdresse.down') as down:
```

#5 - 01 octobre 2019 08:47 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

Quand même vraiment pas fan de cette pratique de réindenter de manière toute différente, et encore moins de tout taper entre parenthèses. For the record donc, à qui voudrait relire le patch dans le futur,

```
+                        Q(after_timestamp__isnull=True) | Q(after_timestamp__lt=timezone.now()),
```

Anyway.

#6 - 01 octobre 2019 10:41 - Benjamin Dauvergne

- Fichier 0003-misc-add-after_timestamp-to-run-Job-later-36215.patch ajouté

- Fichier 0001-misc-remove-unused-statement-in-jobs-36215.patch ajouté

- Fichier 0002-misc-prevent-locking-all-jobs-36215.patch ajouté

- Statut changé de Solution validée à Solution proposée

Voilà, juste que c'est pas PEP8 et juste assez illisible de ne pas commencer les lignes par un opérateur, même si les patchs minimaux c'est bien aussi.

#7 - 01 octobre 2019 13:49 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

(je ne vérifie pas)

#8 - 01 octobre 2019 13:54 - Benjamin Dauvergne

- Statut changé de Solution validée à Résolu (à déployer)

```
commit d629c50e2eccb271d9a84ac6d7120fc3ca29b645
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Sat Sep 21 11:06:39 2019 +0200

    misc: add after_timestamp to run Job later (#36215)

    The after_timestamp can be set:
    - when adding the job with:

        self.add_job(..., after_timestamp=datetime(...))

    - when skipping a job with:

        raise SkipJob(after_timestamp=datetime(...))

commit 95904dbaae1fd7933e2b6ec6304aeb58c14a7ffd
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Sat Sep 21 10:51:21 2019 +0200

    misc: prevent locking all jobs (#36215)

    .first() does list(qs)[:1] which will select all target jobs,
    we must add a LIMIT 1 before .first() to lock only the job we are
    looking for. Ordering is necessary as .first() will do an
    .order_by('pk') on an unordered queryset to get a deterministic result
    in all cases and ordering a sliced queryset is not possible.

commit 8f39fbc38ca19a3d7740743b21845eba32048762
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Tue Oct 1 10:30:13 2019 +0200

    misc: remove unused statement in jobs() (#36215)

    Should have been removed in a39595b.
```

#9 - 04 octobre 2019 16:15 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Solution déployée

Fichiers

0001-misc-prevent-locking-all-jobs-36215.patch	1,23 ko	21 septembre 2019	Benjamin Dauvergne
0002-misc-add-after_timestamp-to-run-Job-later-36215.patch	6,04 ko	21 septembre 2019	Benjamin Dauvergne
0001-misc-prevent-locking-all-jobs-36215.patch	1,23 ko	21 septembre 2019	Benjamin Dauvergne
0002-misc-add-after_timestamp-to-run-Job-later-36215.patch	8,88 ko	21 septembre 2019	Benjamin Dauvergne
0003-misc-add-after_timestamp-to-run-Job-later-36215.patch	8,49 ko	01 octobre 2019	Benjamin Dauvergne
0001-misc-remove-unused-statement-in-jobs-36215.patch	895 octets	01 octobre 2019	Benjamin Dauvergne
0002-misc-prevent-locking-all-jobs-36215.patch	1,44 ko	01 octobre 2019	Benjamin Dauvergne