Episode 281

Foreman

Ryan Bates asked recently on Twitter about the best way to manage the background processes that a Rails application relies on while its in development mode. Several people suggested Foreman[1] and that's what we'll cover in this episode.

For example, to use Ryan's GoVsGo[2] application in development we need to start up a number of background processes beforehand. These include Beanstalkd, a `script/worker` process and a Faye server. We have to start up all of these every time we want to use the application in development. It would be much better if there was a simpler way to manage all of these processes and this is where Foreman comes in.

Installing and Using Foreman

Foreman is supplied as a gem and is installed in the usual way.

```
$ gem install foreman
```

Foreman reads the processes it's going to manage from file called `Procfile` in the Rails application's root directory so we'll need to create this. In this file we'll list the processes we want Foreman to manage. Each one is defined by a name followed by a colon and the process we want to run so we'll enter our three processes like this:

`/Procfile`

```
beanstalk:  beanstalkd
worker:     ./script/worker
faye:       rackup faye.ru -s thin -E production
```

Once we've written our `Procfile` we can check that it's valid by running the `foreman check` command.

```
$ foreman check
valid procfile detected (beanstalk, worker, faye)
```

If we use tabs to separate the name from the command instead of spaces we'll see deprecation errors so we'll need to bear that in mind when writing our `Procfile`.

---

[1] https://github.com/ddollar/foreman

[2] https://github.com/ryanb/govsgo

If our file is reported as being valid we can try running Foreman with the `foreman start` command.

```
$ foreman start
20:06:35 beanstalk.1  | started with pid 23140
20:06:35 worker.1     | started with pid 23141
20:06:35 faye.1       | started with pid 23143
20:06:36 faye.1       | >> Thin web server (v1.2.11 codename Bat-
Shit Crazy)
20:06:36 faye.1       | >> Maximum connections set to 1024
20:06:36 faye.1       | >> Listening on 0.0.0.0:9292, CTRL+C to
stop
20:06:38 worker.1     | [2011-08-29 20:06:38 +0100] Working 1
jobs: [ Game.move ]
```

When we do each of the processes we've specified in the `Procfile` will be run and be listed along with its output in the terminal. If we type CTRL+C in that terminal window each process is stopped.

Foreman has a useful manual page[3] that documents its features and the options that can be passed in and we'll take a look at some of those now. If we just want to start up one process we can pass its name in directly.

```
$ foreman start faye
20:35:41 faye.1       | started with pid 23322
20:35:41 faye.1       | >> Thin web server (v1.2.11 codename Bat-
Shit Crazy)
20:35:41 faye.1       | >> Maximum connections set to 1024
20:35:41 faye.1       | >> Listening on 0.0.0.0:9292, CTRL+C to
stop
```

If we want to start up multiple copies of a process we can do so by using the `-c` flag. For example, we can start up 4 `worker` processes by running `foreman start -c worker=4`.

[3] http://ddollar.github.com/foreman/

```
$ foreman start -c worker=4
20:39:41 beanstalk.1  | started with pid 23366
20:39:41 worker.1     | started with pid 23368
20:39:41 worker.2     | started with pid 23370
20:39:41 worker.3     | started with pid 23372
20:39:41 worker.4     | started with pid 23374
20:39:41 faye.1       | started with pid 23376
20:39:42 faye.1       | >> Thin web server (v1.2.11 codename Bat-
Shit Crazy)
20:39:42 faye.1       | >> Maximum connections set to 1024
20:39:42 faye.1       | >> Listening on 0.0.0.0:9292, CTRL+C to
stop
20:39:46 worker.3     | [2011-08-29 20:39:46 +0100] Working 1
jobs: [ Game.move ]
20:39:46 worker.1     | [2011-08-29 20:39:46 +0100] Working 1
jobs: [ Game.move ]
20:39:46 worker.2     | [2011-08-29 20:39:46 +0100] Working 1
jobs: [ Game.move ]
20:39:47 worker.4     | [2011-08-29 20:39:47 +0100] Working 1
jobs: [ Game.move ]
```

This will start up all of the processes once, except for `worker` which will start four times.

Exporting Processes

Foreman has an `export` command which is useful for exporting the list of processes in a format that we can use on a production server. It currently supports `inittab` and `upstart` formats. If we run `foreman export upstart .` Foreman will write a number of upstart configuration files to the current directory. If we take a look at one of them we'll see that it will execute a command to start `beanstalkd` and handle logging.

/govsgo-beanstalk-1.conf

```
start on starting govsgo-beanstalk
stop on stopping govsgo-beanstalk
respawn

exec su - govsgo -c 'cd /Users/eifion/govsgo; export PORT=5000;
beanstalkd >> /var/log/govsgo/beanstalk-1.log 2>&1'
```

That's it for this episode on Foreman. It's been a shorter than usual episode but Foreman is a simple solution for a specific problem and does its job well. If you have to start up a number of processes to get your Rails application into a working state in development mode then Foreman is well worth considering.