
The Mechanical MOOC Documentation

Release 2.0

P2PU

November 18, 2014

1 Quickstart	3
1.1 What's next?	3
2 Design your course	5
3 Setting up the Mechanical MOOC	7
3.1 Setting up email	7
3.2 Setting up Heroku scheduler	7
3.3 Setup a domain for Heroku	8
3.4 Setup a domain for mailgun	8
3.5 Create a new sequence	8
4 Customize site	9
5 Configure class photo (optional)	11
5.1 Configure Amazon Web Services	11
5.2 Configure getting profile data from Google+ and Twitter	11
5.3 Sending an email to everyone who signed up with a link to the class photo	11
6 Grouping users	13
7 Creating, scheduling and sending emails	15
8 Dump data after the MOOC	17

Contents:

Quickstart

The Mechanical MOOC is a Django application. This quickstart will show you how to deploy the Mechanical MOOC to [heroku](#). To complete this quickstart you will need to use the linux command line.

Set up a [heroku account](#) and install the heroku toolbelt.

Get the code from [github](#):

```
git clone https://github.com/p2pu/mechanical-mooc
```

change into the source directory and run:

```
heroku create
```

Set the django SECRET_KEY by running:

```
heroku config:set SECRET_KEY="`uuidgen | md5pass`"
```

Next push the application code to heroku:

```
git push heroku master
```

Then add the free development database and create the database tables:

```
heroku addons:add heroku-postgresql:dev  
heroku run python manage.py syncdb --migrate
```

During the database creation, you need to create a superuser. This is the user that you will use to log in and create and send emails. Be sure to remember the credentials and don't make them admin:password!!

And finally, to open the Mechanical MOOC in your browser run:

```
heroku open
```

1.1 What's next?

- *Complete the Mechanical MOOC setup*
- *Design your course*
- *Share your course idea and ask questions on the P2PU forum*
- *Update website and messaging copy*
- *Configure class photo (optional)*

Design your course

The Mechanical MOOC is built around the idea that the web is a pretty good platform and that we don't need to reinvent the wheel. Because of that approach using the Mechanical MOOC for a course will require using multiple tools for the different pieces of a big open online course.

The landing page. The Mechanical MOOC gives you a place where you can tell people what your course is about. If you use the recommended deployment method this page should be able to hold up to reasonably high spikes in traffic.

Signup. This is also part of the Mechanical MOOC. Users sign up right on the landing page, fill in some info if you want and then they are watching Vanilla Ice! Nice!

Sending emails. The next bit the Mechanical MOOC does for you is send out emails to everyone who signed up for a course. You can author your emails in a WYSIWYG editor and schedule them to be delivered in the future.

The content. While it is possible to put some content on the Mechanical MOOC itself, this hasn't been the way we used it, most content lived on other platforms. For the content you need to host, you can use a tool like WordPress or GithubPage.

The community. The Mechanical MOOC gives you some tools to help users establish identity in your course in the form of a collaborative class photo. But hosting the community is best done using an existing social media site. We've used Google+, Twitter, SoundCloud and OpenStudy in the past. The right tool will depend on the community you are targeting and what they are comfortable with. You may end up using more than one tool.

There are many other good tools that you can use that will depend on the way your course is set up and the topic of your course. We've used online sound editing tools, online coding sites and a few other tools in a bunch of different courses.

If running your course like this makes sense to you, you should proceed with setting up the Mechanical MOOC. You can find instructions on how to do that here. If you need assistance with this, you can reach out to us for help, advice on [this thread in the P2PU forum](#).

Setting up the Mechanical MOOC

If you followed the instructions in the *quick start* you will have a basic running example of the Mechanical MOOC, but you won't be able to send out emails, it will run at a subdomain at heroku and all the content will still be the example content.

3.1 Setting up email

For sending email and managing mailing lists we use *mailgun*. You need to sign up for a mailgun account and get the API keys.

Set the API keys in the mechanical MOOC by running:

```
heroku config:set MAILGUN_API_KEY=<api-key>
heroku config:set MAILGUN_API_DOMAIN=<your-domain>
```

You need to replace *<api-key>* with the API Key you see at <https://mailgun.com/cp>.

And you need to replace *<your-domain>* with the domain you are using for mailgun. If you have just set up mailgun, chances are that this will be one of the sandbox domains: *something.mailgun.org*.

Lastly you need to set the default email address that all emails to users will come from. Run:

```
heroku config:set DEFAULT_FROM_EMAIL=mooc@example.net
```

3.2 Setting up Heroku scheduler

Enable the Heroku scheduler to send out the welcome email to new users who signed up and to send the emails that you schedule.

Tasks use an Heroku add-on that runs some python scripts. Run:

```
heroku addons:add scheduler:standard
```

to enable the add-on.

Then add the tasks - log into your application dashboard. *heroku addons:open scheduler* will open the dashboard in your web browser. Through the web form, add the following scripts:

- `python manage.py handle_new_signups`
- `python manage.py send_scheduled_mail`

The first task should run every 10 minutes and the second task should run every hour.

3.3 Setup a domain for Heroku

Because of the way that Heroku works, it is best to run the Mechanical MOOC at a subdomain (remember that `www.` is also a subdomain if another subdomain doesn't make sense). You can follow [these](#) instructions on how to configure a custom domain. This will be the domain where you send users to sign up.

Configure the Mechanical MOOC for your domain by running:

```
heroku config:set MOOC_DOMAIN=<your-domain>
```

3.4 Setup a domain for mailgun

Mailgun needs a custom domain to [function properly](#). You can follow the instructions there to setup the domain. Once done, run:

```
heroku config:set MAILGUN_API_DOMAIN=<your-domain>
```

with your proper domain.

3.5 Create a new sequence

A sequence is a single run of your course, obviously your course will be a resounding success and you will want to run many more! To create a sequence run the following commands:

```
heroku run python manage.py shell
> from sequence import models
> from datetime import datetime
> start_date = datetime(2013, 12, 1)
> signup_close_date = datetime(2013, 11, 23)
> models.create_sequence(start_date, signup_close_date)
> exit()
```

This will automatically create a mailgun campaign and a single mailgun mailing list to which new sign up will be added.

Customize site

To customize the landing page, you need to edit `templates/signup/index.html`. This page includes the form used during signup. You can customize the questions without needing to modify the Python code, but be sure to put in unique names for every field you add and to validate anything that needs validation on the client side. The only field that you cannot remove is the timezone field.

To customize the welcome email sent to users, edit the three files in `templates/signup/emails/`.

The content that lives in the about and FAQ links at the top lives in `templates/about.html` and `templates/faq.html` respectively.

To edit the topnav, update:

```
<div class="nav-collapse collapse">
  <ul class="nav main-menu">
    <li><a class="home" href="{% url 'home' %}"><i class="icon-home"></i></a></li>
    <li><a href="{% url 'faq' %}">FAQ</a></li>
    <li><a href="http://mechanicalmooc.wordpress.com">Blog</a></li>
    <li><a href="{% url 'about' %}">About</a></li>
  </ul>
</div>
```

in `templates/base.html`.

Configure class photo (optional)

The MechanicalMOOC supports a light weight method for users to put a face to their participation.

5.1 Configure Amazon Web Services

To use this feature you need an Amazon Web Services account since images are uploaded to Amazon S3. Once you created the account, create a keypair with access to the AWS S3 bucket that you want to use. See the [documentation over at Amazon](#) to see how to do this.

You also need to create a new bucket where the images will be uploaded to.

Once done, run:

```
heroku config:set AWS_ACCESS_KEY_ID=<aws_access_key>
heroku config:set AWS_SECRET_ACCESS_KEY=<aws_secret_key>
heroku config:set AWS_S3_BUCKET=<bucket-name>
```

5.2 Configure getting profile data from Google+ and Twitter

Next you need to obtain app keys for Twitter and Google+. For twitter you can follow [these instructions](#). For Google+, see [this](#).

Once you've obtained the needed credentials, run:

```
heroku config:set TWITTER_ACCESS_TOKEN=<access-token>
heroku config:set TWITTER_ACCESS_TOKEN_SECRET=<access-token-secret>
heroku config:set GOOGLE_PLUS_API_KEY=<api-key>
```

5.3 Sending an email to everyone who signed up with a link to the class photo

Need to explain, but code lives [here](#).

Grouping users

Creating, scheduling and sending emails

Dump data after the MOOC
