

1. Install PyDev in Eclipse (<<http://pydev.org/download.html>>, zip package at <<http://sourceforge.net/projects/pydev/files/pydev/>>) and Python 2.7.3 (<<http://python.org/download/>>) if you have not already done so.
2. Install Google App Engine SDK from <<https://developers.google.com/appengine/downloads>>.
3. Start a new PyDev project in Eclipse. Take note of the directory it is created in.
4. Create a new source folder (right click on the project and select “New > Source Folder”)
5. Download and unzip the zip file from <<https://github.com/django-nonrel/django-testapp>>, and copy the contents into the source folder you just created (This will be in “[name of the project]/[name of the source folder]” in your Eclipse workspace directory).
6. Download and unzip the following links, copying the corresponding unzipped directories into the source folder (this is the directory that now contains manage.py).

<i>Archive link</i>	<i>Directory to copy</i>
< https://github.com/django-nonrel/djangotoolbox >	djangotoolbox/
< https://github.com/django-nonrel/django-dbindexer >	dbindexer/
< https://github.com/django-nonrel/django-nonrel >	django/
< https://bitbucket.org/twanschik/django-autoload >	autoload/
< https://github.com/django-nonrel/django-permission-backend-nonrel >	permission_backend_nonrel/

7. Download and unzip the zip file from <<https://github.com/django-nonrel/djangoappengine>>, and copy the contents into a new directory djangoappengine/ in the source folder (this is the directory containing manage.py).
8. Refresh your project so it recognizes the new files and directories.
9. Right click on your project, go to the “PyDev” menu, and select “Set as Django project”.
10. Modify the settings.py file.
 1. Add a line after “'djangotoolbox',” in the INSTALLED_APPS list that reads “'permission_backend_nonrel',” and uncomment the line “”. Your INSTALLED_APPS should thus read something like:

```
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.contenttypes',
    'django.contrib.auth',
    'django.contrib.sessions',
    'djangotoolbox',
    'permission_backend_nonrel',
    'autoload',
    'dbindexer'

    # djangoappengine should come last, so it can override a few manage.py commands
    'djangoappengine',
)
```

2. After the TEMPLATE_CONTEXT_PROCESSORS section, add the following lines:

```
AUTHENTICATION_BACKENDS = (
    'permission_backend_nonrel.backends.NonrelPermissionBackend',
)
```

11. You can now start developing and deploying your code once you have set up your Google App Engine account. Sign up and create an application at <https://appengine.google.com/>. Make a note of the subdomain you registered.
12. Edit `app.yaml` in the root directory and modify the line starting with “application:” to end with the name of the subdomain you registered (rather than the default “ctst”).
13. You can now make your Django applications. Start by right clicking your main project folder (the one with the little “P” in the corner of the icon) and go to “Django > Create application...” You can call the application whatever you like, but I recommend “core” for the first application you make (the application name will become a Python package name).
14. Make sure to create a superuser before you test the app for the first time. Go to “Django > Custom command...” and add the command “createsuperuser” and run it.
15. *NOTE:* If you wish to use the Django admin/ path on your website (e.g. as described at <https://docs.djangoproject.com/en/dev/intro/tutorial02/>) you will need to add it to your URL configuration in `urls.py` in much the same way as described on that page (following the instructions in “Activating the admin site”), except that you will need to add the lines which would normally be uncommented. You will also need to modify your `settings.py` file to uncomment the line containing “‘django.contrib.admin’,”.
16. To test your app locally before deploying, right click on the main project folder (that would be the one with the little “P” in the corner of the icon) and go to “Django > Custom command...” Add a command “runserver”. Running this custom command will then set up a test server on your local computer!
17. Right click on the main project folder (that would be the one with the little “P” in the corner of the icon) and go to “Django > Custom command...” Add a command “deploy”. Running this custom command will then deploy your code to Google App Engine!
18. At this point you can commit your code to Git! But I would warn against committing the `settings.py` file (the one you edited) as is, as putting it as is into your version control is a security problem! If someone knows your `SECRET_KEY` they could use it to hijack your user's sessions (See: <https://code.djangoproject.com/wiki/SplitSettings> on details on what might be a good idea to move and how).