

Getting Started with Haystack

Search is a topic of ever increasing importance. Users increasingly rely on search to separate signal from noise and find what they're looking for quickly. In addition, search can provide insight into what things are popular (many searches), what things are difficult to find on the site and ways you can improve the site better.

To this end, Haystack tries to make integrating custom search as easy as possible while being flexible/powerful enough to handle more advanced use cases.

Haystack is a reusable app (that is, it relies only on it's own code and focuses on providing just search) that plays nicely with both apps you control as well as third-party apps (such as `django.contrib.*`) without having to modify the sources.

Haystack also does pluggable backends (much like Django's database layer), so virtually all of the code you write ought to be portable between which ever search engine you choose.

Note

If you hit a stumbling block, there is both a [mailing list](#) and [#haystack on irc.freenode.net](#) to get help.

This tutorial assumes that you have a basic familiarity with the various major parts of Django (models/forms/views/settings/URLconfs) and tailored to the typical use case. There are shortcuts available as well as hooks for much more advanced setups, but those will not be covered here.

For example purposes, we'll be adding search functionality to a simple note-taking application. Here is `myapp/models.py`:

```
from django.db import models
from django.contrib.auth.models import User

class Note(models.Model):
    user = models.ForeignKey(User)
    pub_date = models.DateTimeField()
    title = models.CharField(max_length=200)
    body = models.TextField()

    def __unicode__(self):
        return self.title
```