# SolrTutorial.com

## Solr in 5 minutes

Solr makes it easy to run a full-featured search server. In fact, its so easy, I'm going to show you how in 5 minutes!



## Installing Solr

For the purposes of this tutorial, I'll assume you're on a Linux or Mac environment.

You should also have **JDK 6 or above** installed.

```
wget http://download.nextag.com/apache/lucene/solr/5.3.0/solr-5.3.0.tgz
tar -zxvf solr-5.3.0.tgz
cd solr-5.3.0
```

## Starting Solr

Solr comes with an **example** directory which contains some sample files we can use. The available examples are:

```
    cloud        : SolrCloud example
    dih          : Data Import Handler (rdbms, mail, rss, tika)
    schemaless   : Schema-less example (schema is inferred from da
    techproducts : Kitchen sink example providing comprehensive ex
```

To run Solr with one of these examples, use **bin/solr -e [EXAMPLE]** where [EXAMPLE] is one of above.
e.g. **bin/solr -e dih**. Let's run the **techproducts** example.

```
bin/solr -e techproducts
```

You should see something like this in the terminal.

```
Creating Solr home directory /tmp/solrt/solr-5.3.0/example/techproducts/solr
```

```
Starting up Solr on port 8983 using command:
bin/solr start -p 8983 -
s"example/techproducts/solr"

Waiting up to 30 seconds to see Solr running on
port 8983 [/]
Started Solr server on port 8983 (pid=12281).
Happy searching!


Setup new core instance directory:
/tmp/solrt/solr-
5.3.0/example/techproducts/solr/techproducts

Creating new core 'techproducts' using command:
http://localhost:8983/solr/admin/cores?
action=CREATE&name=techproducts&instanceDir=techproducts

{
  "responseHeader":{
    "status":0,
    "QTime":2060},
  "core":"techproducts"}


Indexing tech product example docs
from /tmp/solrt/solr-5.3.0/example/exampledocs
SimplePostTool version 5.0.0
Posting files to [base] url
http://localhost:8983/solr/techproducts/update
using content-type application/xml...
POSTing file money.xml to [base]
POSTing file manufacturers.xml to [base]
POSTing file hd.xml to [base]
POSTing file sd500.xml to [base]
POSTing file solr.xml to [base]
POSTing file utf8-example.xml to [base]
POSTing file mp500.xml to [base]
POSTing file monitor2.xml to [base]
POSTing file vidcard.xml to [base]
POSTing file ipod_video.xml to [base]
POSTing file monitor.xml to [base]
POSTing file mem.xml to [base]
POSTing file ipod_other.xml to [base]
POSTing file gb18030-example.xml to [base]
14 files indexed.
COMMITting Solr index changes to
http://localhost:8983/solr/techproducts/update...
Time spent: 0:00:00.491

Solr techproducts example launched successfully.
Direct your Web browser to
http://localhost:8983/solr to visit the Solr
Admin UI
```

To verify that Solr is running, you can do this:

```
bin/solr status
```

Which should produce something like this:

```
Found 1 Solr nodes:

Solr process 12281 running on port 8983
{
  "solr_home":"/tmp/solrt/solr-
5.3.0/example/techproducts/solr/",
  "version":"5.3.0 1696229 - noble - 2015-08-17
17:10:43",
  "startTime":"2015-09-14T22:41:56.876Z",
  "uptime":"0 days, 0 hours, 1 minutes, 7
seconds",
  "memory":"32.7 MB (%6.7) of 490.7 MB"}
```

Solr is now running! You can now access the Solr Admin webapp by loading **http://localhost:8983/solr/** in your web browser.

## Indexing Data

The startup script already added some sample data to our Solr instance, but we're going to re-add some docs just to see how it all works.

The **example/exampledocs** folder contains some XML files we can use.

A quick glance at one of the XML files reveals that each Solr **document** consists of multiple **fields**. Each field has a **name** and a **value**. For example:

```
<add><doc>
  <field name="id">9885A004</field>
  <field name="name">Canon PowerShot
SD500</field>
  <field name="manu">Canon Inc.</field>
...
  <field name="inStock">true</field>
</doc></add>
```

The **post.jar** in the same folder provides a convenient way to add files to Solr.

```
cd example/exampledocs
java -Dc=techproducts -jar post.jar sd500.xml
```

That produces:

```
SimplePostTool version 5.0.0
Posting files to [base] url
http://localhost:8983/solr/techproducts/update
using content-type application/xml...
POSTing file sd500.xml to [base]
1 files indexed.
COMMITting Solr index changes to
http://localhost:8983/solr/techproducts/update...
Time spent: 0:00:00.186
```

This response tells us that the POST operation was successful.

Note that there are 2 main ways of adding data to Solr:

1. HTTP
2. Native client

We'll explore these in greater detail in a subsequent tutorial.

## Searching

Let's see if we can retrieve the document we just added.

Since Solr accepts HTTP requests, you can use your web browser to communicate with Solr: **http://localhost:8983/solr/techproducts/select?q=sd500&wt=json**

This returns the following JSON result:

```
{
    "responseHeader": {
        "status": 0,
        "QTime": 3,
        "params": {
            "q": "sd500",
            "wt": "json"
        }
    },
    "response": {
        "numFound": 1,
        "start": 0,
        "docs": [
            {
                "id": "9885A004",
                "name": "Canon PowerShot SD500",
                "manu": "Canon Inc.",
                "manu_id_s": "canon",
                "cat": [
                    "electronics",
                    "camera"
                ],
```

```
                "features": [
                    "3x zoop, 7.1 megapixel
Digital ELPH",
                    "movie clips up to 640x480
@30 fps",
                    "2.0\" TFT LCD, 118,000
pixels",
                    "built in flash, red-eye
reduction"
                ],
                "includes": "32MB SD card, USB
cable, AV cable, battery",
                "weight": 6.4,
                "price": 329.95,
                "price_c": "329.95,USD",
                "popularity": 7,
                "inStock": true,
                "manufacturedate_dt": "2006-02-
13T15:26:37Z",
                "store": "45.19614,-93.90341",
                "_version_": 1512330534874775600
            }
        ]
    }
}
```


Nice! A quick verification with **sd500.xml** should confirm that all is in order.

Let's now do some real searching.

Here's a demonstration of retrieving the **name** and **id** of all documents with **inStock = false**: **http://localhost:8983/solr/techproducts/select?q=inStock:false&wt=json&fl=id,name**


```
{
    "responseHeader": {
        "status": 0,
        "QTime": 3,
        "params": {
            "fl": "id,name",
            "q": "inStock:false",
            "wt": "json"
        }
    },
    "response": {
        "numFound": 4,
        "start": 0,
        "docs": [
            {
                "id": "EN7800GTX/2DHTV/256M",
                "name": "ASUS Extreme
N7800GTX/2DHTV (256 MB)"
```

```
            },
            {
                "id": "100-435805",
                "name": "ATI Radeon X1900 XTX 512
MB PCIE Video Card"
            },
            {
                "id": "F8V7067-APL-KIT",
                "name": "Belkin Mobile Power Cord
for iPod w/ Dock"
            },
            {
                "id": "IW-02",
                "name": "iPod & iPod Mini USB 2.0
Cable"
            }
        ]
    }
}
```

You'll learn more about the various URL query parameters in a separate tutorial.

## Shutdown

To shutdown Solr, use **bin/solr stop**. This will shutdown Solr cleanly.

Solr is fairly robust, so even in situations of OS or disk crashes, it is unlikely that Solr's index will become corrupted.