

```

handleSelect=true will use consistent error handling for /select and /update
handleSelect=false will use solr1.1 style error formatting
-->
<requestDispatcher handleSelect="true">
  <!--Make sure your system has some authentication before enabling remote streaming! -->
  <requestParsers enableRemoteStreaming="false" multipartUploadLimitInKB="2048"/>
  <!-- Set HTTP caching related parameters (for proxy caches and clients).

  To get the behaviour of Solr 1.2 (ie: no caching related headers)
  use the never304="true" option and do not specify a value for
  <cacheControl>
  -->
  <!-- <httpCaching never304="true"> -->
  <httpCaching lastModifiedFrom="openTime" etagSeed="Solr">
    <!-- lastModFrom="openTime" is the default, the Last-Modified value
    (and validation against If-Modified-Since requests) will all be
    relative to when the current Searcher was opened.
    You can change it to lastModFrom="dirLastMod" if you want the
    value to exactly correspond to when the physical index was last
    modified.

    etagSeed="..." is an option you can change to force the ETag
    header (and validation against If-None-Match requests) to be
    different even if the index has not changed (ie: when making
    significant changes to your config file)

    lastModifiedFrom and etagSeed are both ignored if you use the
    never304="true" option.
    -->
    <!-- If you include a <cacheControl> directive, it will be used to
    generate a Cache-Control header, as well as an Expires header
    if the value contains "max-age="

    By default, no Cache-Control header is generated.

    You can use the <cacheControl> option even if you have set
    never304="true"
    -->
    <!-- <cacheControl>max-age=30, public</cacheControl> -->
  </httpCaching>
</requestDispatcher>
<!-- requestHandler plugins... incoming queries will be dispatched to the
correct handler based on the path or the qt (query type) param.
Names starting with a '/' are accessed with the a path equal to the
registered name. Names without a leading '/' are accessed with:
  http://host/app/select?qt=name
If no qt is defined, the requestHandler that declares default="true"
will be used.
-->
<requestHandler name="standard" class="solr.SearchHandler" default="true">
  <!-- default values for query parameters -->
  <lst name="defaults">
    <str name="echoParams">explicit</str>
    <!--
    <int name="rows">10</int>
    <str name="fl">*</str>
    <str name="version">2.2</str>
    -->
  </lst>
</requestHandler>
<!-- DisMaxRequestHandler allows easy searching across multiple fields
for simple user-entered phrases. It's implementation is now
just the standard SearchHandler with a default query type
of "dismax".
see http://wiki.apache.org/solr/DisMaxRequestHandler
-->
<requestHandler name="dismax" class="solr.SearchHandler">
  <lst name="defaults">
    <str name="defType">dismax</str>
    <str name="echoParams">explicit</str>
    <float name="tie">0.01</float>
    <str name="qf">
      text^0.5 features^1.0 name^1.2 sku^1.5 id^10.0 manu^1.1 cat^1.4
    </str>
    <str name="pf">
      text^0.2 features^1.1 name^1.5 manu^1.4 manu_exact^1.9
    </str>
    <str name="bf">
      ord(popularity)^0.5 recip(rord(price),1,1000,1000)^0.3
    </str>
    <str name="fl">
      id,name,price,score
    </str>
    <str name="mm">
      2<lt;-1 5<lt;-2 6<lt;90%
    </str>
    <int name="ps">100</int>
    <str name="q.alt">*</str>
    <!-- example highlighter config, enable per-query with hl=true -->
    <str name="hl.fl">text features name</str>
    <!-- for this field, we want no fragmenting, just highlighting -->
    <str name="f.name.hl.fragmentize">0</str>
    <!-- instructs Solr to return the field itself if no query terms are
    found -->
    <str name="f.name.hl.alternateField">name</str>
    <str name="f.text.hl.fragmenter">regex</str>

```