

RealTimeGet

<!-- Solr4.0

Realtime Get

- [Realtime Get](#)
- [Introduction](#)
- [Quick Start](#)

Introduction

For index updates to be visible (searchable), some kind of commit must reopen a searcher to a new point-in-time view of the index.

The **realtime get** feature allows retrieval (by unique-key) of the latest version of any documents without the associated cost of reopening a searcher. This is primarily useful when using Solr as a NoSQL data store and not just a search index.

Quick Start

Realtime-get currently relies on the update log feature, which is currently not enabled by default. It is enabled in the example that comes with Solr.

Open your solrconfig.xml file, and add the following snippet inside the updateHandler section (there should already be a commented out sample in the example solrconfig.xml)

```
<updateLog class="solr.FSUpdateLog">
  <str name="dir">${solr.data.dir}</str>
</updateLog>
```

The latest example solrconfig.xml should also have the **/get** request handler defined.

Start (or restart) the solr server, and then index a document:

```
curl 'http://localhost:8983/solr/update/json?commitWithin=10000000' -H 'Content-type:application/json' -d '[{
  "id" : "mydoc",
  "title" : "realtime-get test!"
}]'
```

We used commitWithin with a high value to avoid this document being committed and visible to normal search, so we have time to test out the realtime feature.

If we do a normal search, this document should not be found:

<http://localhost:8983/solr/select?q=id:mydoc>

```
...
"response":{"numFound":0,"start":0,"docs":[]}
```

However if we use the realtime-get handler exposed at **/get**, we should be able to retrieve that document:

<http://localhost:8983/solr/get?id=mydoc>

```
{ "doc": { "id": "mydoc", "title": "realtime-get test!" } }
```

You can also specify multiple documents at once via the **ids** parameter and a comma separated list of ids, or by using multiple **id** parameters. If you specify multiple ids, or use the **ids** parameter, the response will mimic a normal query response to make it easier for existing clients to parse. Since we've only indexed one document, the following equivalent examples just repeat the same id.

<http://localhost:8983/solr/get?ids=mydoc,mydoc>

<http://localhost:8983/solr/get?id=mydoc&id=mydoc>

```
{
  "response": {"numFound": 2, "start": 0, "docs": [
    {
      "id": "mydoc",
      "title": ["realtime-get test!"]},
    {
      "id": "mydoc",
      "title": ["realtime-get test!"]}
  ]}
}
```