TECHNICAL BRIEF

SOLR-18350 — Stable Schema Access During Request Processing

Issue: Many callers of SolrCore#getLatestSchema should use alternatives
Branch: solr-18350-stable-schema
Status: Ready for review; CI awaiting Apache Solr maintainer approval
Commit: 87d6dee889
Pull request: https://github.com/apache/solr/pull/4817
Prepared By: Shrey Narayan
Organization: NextBricks
Date: August 26, 2026

Outcome Thirty-six inappropriate getLatestSchema() references were removed across request-, parser-, searcher-, update-, and language-model paths. The patch now keeps each request or searcher paired with its stable schema snapshot.

Executive summary

SolrCore#getLatestSchema() can change while a request is executing. SolrQueryRequest#getSchema() is intentionally a request-lifetime snapshot, while SolrIndexSearcher#getSchema() is the schema associated with that searcher. Mixing these sources can make one operation interpret fields using different schema generations.

This change replaces only callers that already have a request or searcher in scope. Schema mutation, core initialization, managed-resource setup, and other paths that genuinely need the live schema remain unchanged.

Measure	Result
Production files changed	15
Test files changed	1 Mockito fixture
Latest-schema references removed	36
Focused tests passing	269 unique tests
Formatting / affected checks	Passing

Technical invariant

- Use req.getSchema() when processing a SolrQueryRequest; it remains stable for that request.
- Use searcher.getSchema() when reading data through a SolrIndexSearcher; it matches that searcher.
- Use core.getLatestSchema() only when the operation explicitly needs the core's current live schema.

```
// Request-scoped code
final SchemaField field = req.getSchema().getField(fieldName);

// Searcher-scoped code
final SchemaField field = searcher.getSchema().getField(fieldName);
```

Code changes

Area	Representative files	Change
API documentation	SolrCore.java	Documents the request and searcher alternatives directly on getLatestSchema().
Schema API	SchemaHandler.java	Uses the request snapshot for schema reads.
Realtime get / facets	RealTimeGetComponent.java; SimpleFacets.java	Uses request or searcher schemas according to the object performing the work.
Query parsing	Payload*; ValueSourceParser; vector parsers	Uses req.getSchema() consistently during parsing.
Update processing	ContentHash; version constraints; tolerant updates	Uses request/command schema snapshots.
Language models	Text-to-vector and enrichment paths	Uses the request schema for field validation and lookup.
Tests	ContentHashVersionProcessorTest.java	Updates the mock request to expose its schema snapshot.

Javadoc added

```
Request processing code should normally use SolrQueryRequest#getSchema()
so that the schema remains stable for the lifetime of the request.
Code operating on a SolrIndexSearcher should normally use
SolrIndexSearcher#getSchema() so that the schema matches the searcher.
```

Intentional exclusions

The patch is deliberately not a repository-wide textual replacement. These categories retain getLatestSchema():

- SchemaManager and schema-mutation retry logic, which must observe and install the newest schema.

- Core construction, searcher creation, codec setup, and plugin initialization paths without a request/searcher snapshot.
- Public core-only helpers where changing the API contract would exceed this focused cleanup.
- Managed-resource and schema-refresh code whose purpose is to inspect live state.

Reviewer note These exclusions are the primary guard against an overbroad mechanical refactor. Every replacement in the patch has an in-scope request or searcher alternative.

Verification

Check	Result	Evidence
./gradlew tidy	PASS	Solr formatting applied successfully.
Affected module checks	PASS	:solr:core:check and :solr:modules:language-models:check with tests excluded; 136 tasks.
Focused core tests	PASS	181 unique tests. Initial run exposed one obsolete mock; fixture corrected and failing class rerun with the same seed.
Language-model tests	PASS	88 tests across query parser, update processor, and factory suites.
Diff validation	PASS	git diff --check reports no whitespace errors.
Whole-repository check	ENVIRONMENT	Code checks progressed successfully; Antora failed because Node split the workspace path at the embedded space.

Focused test commands

```
./gradlew :solr:core:test --tests '<14 affected core suites>'
./gradlew :solr:modules:language-models:test --tests '<5 affected suites>'
./gradlew tidy :solr:core:check :solr:modules:language-models:check -x test
```

Risk assessment

Risk is low. This is an accessor-selection cleanup with no API removal and no schema mutation behavior change. The main semantic risk—pairing indexed data with a newer unrelated schema—is reduced. Existing request and searcher lifetimes already provide the required snapshots.

The change is also reviewable: 16 files, 48 insertions, and 52 deletions, with most edits being one-line accessor substitutions. The single test edit aligns a Mockito fixture with the real SolrQueryRequest contract.