

# MakingApiConsistent

## Lucene's API is inconsistent.

It would be great, if some people could unite to solve this problem.

### Proposed steps to achieve API consistency:

- 1) Identify categories of inconsistencies and define solutions
- 2) Identify concrete inconsistencies including their occurrences
- 3) Examine, which components would be affected by possible refactoring
- 4) Propose patches for refactoring

---

## Let's go

---

### 1) Identify categories of inconsistencies and define solutions

#### Containers don't use Iterable<...> (or more concrete interfaces)

Example:

`IndexSearcher`: "int maxDoc()" and "doc(int i)"

Why is it bad?

Usage of containers must be intuitive and consistent.

Solution:

Make use of Java's collection interfaces such as: Iterable<...>, Collection<...>

#### Public accessible fields

Example:

`ScoreDoc`: "public float score" and "public int doc"

Why is it bad?

People shouldn't have to memorize, which functionality is exposed by public fields, and which by public methods.

Solution:

No public fields. Look at every serious API. There are no public fields!

#### Some methods which return values are named something() others are named getSomething()

Example:

`Fieldable`:

without get: `Reader readerValue()`, `byte[] binaryValue()`, `String stringValue()`, ...

with get: `byte[] getBinaryValue()`, `int getBinaryLength()`, ...

Why is it bad?

People shouldn't have to memorize, which method starts with 'get' or 'set', which without 'get' or 'set'.

Solution:

Methods which somehow represent a property should use names such as `getSomething` / `setSomething`

Boolean methods could use start with `isSomething` / `setSomething`

#### Some methods don't follow standard Java naming guidelines

Example:

`QueryParser`:

`ReInit(...)`, `Term(...)`, `TopLevel(...)` - look to me like constructors!!!

`enable_tracing()`, `disable_tracing()` - oh my!!!

Why is it bad?

People shouldn't have to memorize, which method starts with capital letter, and which doesn't.

Solution:

All such methods and fields MUST be renamed use Java's recommended naming scheme.

## **2) Identify concrete inconsistencies including their occurrences**

TODO

---

## **3) Examine, which components would be affected by possible refactoring**

TODO

---

## **4) Propose patches for refactoring**

TODO

---

former bug report, which led to creation of this wiki page: <https://issues.apache.org/jira/browse/LUCENE-1439>