

Set: A Collection That Refuses Duplicates

A Set holds each item once, and add returns false to report a duplicate -- exactly what a crawler or a dedup check needs, without a separate lookup first.

Automation Foundations · ashishautomationlabs.store

A Set holds items with no duplicates and no positions. Adding something already present simply does nothing -- and add tells you which happened by returning true or false:

```
Set<String> visited = new HashSet<>();
System.out.println("added /login : " + visited.add("/login"));
System.out.println("added /cart : " + visited.add("/cart"));
System.out.println("added /login : " + visited.add("/login"));
System.out.println("size : " + visited.size());
System.out.println("contains /cart: " + visited.contains("/cart"));

added /login : true
added /cart : true
added /login : false
size : 2
contains /cart: true
```

Three adds, size two, and the third add returned false to say "already there." That return value is genuinely useful: it is how a crawler decides whether a page is new without a separate check.

Use a Set when the question is "have I seen this?" or "what are the distinct values?" Use a List when order matters or duplicates are meaningful. A list of every test run belongs in a List; the set of distinct failure messages belongs in a Set.

Is a Set just a Map with no values? Very nearly, and that's a useful way to remember it -- HashSet is built on a HashMap internally, storing your items as keys. That is why both refuse duplicates by the same rule, and why both depend on hashCode.

Both containers are everywhere in a framework. Configuration is a Map: environment to URL, browser name to driver setup, severity to SLA hours. Request headers and API parameters are Maps of name to value. A run summary is a Map of test name to result, built with the exact `getOrDefault` counting line from the previous article. Sets carry the distinct failure reasons in a report, the pages a crawler has already visited, and the unique defect ids so the same failure is filed once.

Both containers have quietly depended on one thing this whole time without saying so: hashCode. What happens when it's missing is the sharpest demonstration in this entire book of why Chapter 7 insisted on it.