

Java Map Interface

A map contains values on the basis of key i.e. key and value pair. Each key and value pair is known as an entry. Map contains only unique keys.

Map is useful if you have to search, update or delete elements on the basis of key.

Useful methods of Map interface

Method	Description
Object put(Object key, Object value)	It is used to insert an entry in this map.
void putAll(Map map)	It is used to insert the specified map in this map.
Object remove(Object key)	It is used to delete an entry for the specified key.
Object get(Object key)	It is used to return the value for the specified key.
boolean containsKey(Object key)	It is used to search the specified key from this map.
Set keySet()	It is used to return the Set view containing all the keys.
Set entrySet()	It is used to return the Set view containing all the keys and values.

Map.Entry Interface

Entry is the sub interface of Map. So we will be accessed it by Map.Entry name. It provides methods to get key and value.

Methods of Map.Entry interface

Method	Description
Object getKey()	It is used to obtain key.
Object getValue()	It is used to obtain value.

Java Map Example: Generic (New Style)

```
import java.util.*;

class MapInterfaceExample{

    public static void main(String args[]){

        Map<Integer,String> map=new HashMap<Integer,String>();

        map.put(100,"Amit");

        map.put(101,"Vijay");

        map.put(102,"Rahul");

        for(Map.Entry m:map.entrySet()){

            System.out.println(m.getKey()+" "+m.getValue());

        }

    }

}
```

```
}
```

Output:

```
102 Rahul  
100 Amit  
101 Vijay
```

Java Map Example: Non-Generic (Old Style)

```
//Non-generic  
import java.util.*;  
public class MapExample1 {  
  public static void main(String[] args) {  
    Map map=new HashMap();  
    //Adding elements to map  
    map.put(1,"Amit");  
    map.put(5,"Rahul");  
    map.put(2,"Jai");  
    map.put(6,"Amit");  
    //Traversing Map  
    Set set=map.entrySet();//Converting to Set so that we can traverse  
    Iterator itr=set.iterator();  
    while(itr.hasNext()){  
      //Converting to Map.Entry so that we can get key and value separately  
      Map.Entry entry=(Map.Entry)itr.next();  
      System.out.println(entry.getKey()+" "+entry.getValue());  
    }  
  }  
}
```

Output:

```
1 Amit  
2 Jai  
5 Rahul  
6 Amit
```

[← prev](#)[next →](#)

Share this page



Latest 4 Tutorials



CouchDB



Docker



Rails



RichFaces