MAPREDUCE - API

http://www.tutorialspoint.com/map_reduce/map_reduce_api.htm

In this chapter, we will take a close look at the classes and their methods that are involved in the operations of MapReduce programming. We will primarily keep our focus on the following −

- JobContext Interface
- Job Class
- Mapper Class
- Reducer Class

JobContext Interface

The JobContext interface is the super interface for all the classes, which defines different jobs in MapReduce. It gives you a read-only view of the job that is provided to the tasks while they are running.

The following are the sub-interfaces of JobContext interface.

S.No.	Subinterface Description
1.	**MapContext<KEYIN, VALUEIN, KEYOUT, VALUEOUT>** Defines the context that is given to the Mapper.
2.	**ReduceContext<KEYIN, VALUEIN, KEYOUT, VALUEOUT>** Defines the context that is passed to the Reducer.

Job class is the main class that implements the JobContext interface.

Job Class

The Job class is the most important class in the MapReduce API. It allows the user to configure the job, submit it, control its execution, and query the state. The set methods only work until the job is submitted, afterwards they will throw an IllegalStateException.

Normally, the user creates the application, describes the various facets of the job, and then submits the job and monitors its progress.

Here is an example of how to submit a job −

```
// Create a new Job
Job job = new Job(new Configuration());
job.setJarByClass(MyJob.class);

// Specify various job-specific parameters
job.setJobName("myjob");
job.setInputPath(new Path("in"));
job.setOutputPath(new Path("out"));

job.setMapperClass(MyJob.MyMapper.class);
job.setReducerClass(MyJob.MyReducer.class);

// Submit the job, then poll for progress until the job is complete
job.waitForCompletion(true);
```

Constructors

Following are the constructor summary of Job class.

S.No	Constructor Summary
1	**Job**
2	**Job***Configurationconf*
3	**Job***Configurationconf, StringjobName*

Methods

Some of the important methods of Job class are as follows −

S.No	Method Description
1	**getJobName** User-specified job name.
2	**getJobState** Returns the current state of the Job.
3	**isComplete** Checks if the job is finished or not.
4	**setInputFormatClass** Sets the InputFormat for the job.
5	**setJobName*Stringname*** Sets the user-specified job name.
6	**setOutputFormatClass** Sets the Output Format for the job.
7	**setMapperClass*Class*** Sets the Mapper for the job.
8	**setReducerClass*Class*** Sets the Reducer for the job.
9	**setPartitionerClass*Class*** Sets the Partitioner for the job.
10	**setCombinerClass*Class*** Sets the Combiner for the job.

Mapper Class

The Mapper class defines the Map job. Maps input key-value pairs to a set of intermediate key-value pairs. Maps are the individual tasks that transform the input records into intermediate records. The transformed intermediate records need not be of the same type as the input records. A given input pair may map to zero or many output pairs.

Method

map is the most prominent method of the Mapper class. The syntax is defined below −

```
map(KEYIN key, VALUEIN value, org.apache.hadoop.mapreduce.Mapper.Context context)
```

This method is called once for each key-value pair in the input split.

Reducer Class

The Reducer class defines the Reduce job in MapReduce. It reduces a set of intermediate values that share a key to a smaller set of values. Reducer implementations can access the Configuration for a job via the JobContext.getConfiguration method. A Reducer has three primary phases − Shuffle, Sort, and Reduce.

- **Shuffle** − The Reducer copies the sorted output from each Mapper using HTTP across the network.

- **Sort** − The framework merge-sorts the Reducer inputs by keys $sincedifferentMappersmayhaveoutputthesamekey$. The shuffle and sort phases occur simultaneously, i.e., while outputs are being fetched, they are merged.

- **Reduce** − In this phase the reduce $Object, Iterable, Context$ method is called for each <key, $collectionofvalues$> in the sorted inputs.

Method

reduce is the most prominent method of the Reducer class. The syntax is defined below −

```
reduce(KEYIN key, Iterable<VALUEIN> values, org.apache.hadoop.mapreduce.Reducer.Context
context)
```

This method is called once for each key on the collection of key-value pairs.

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js