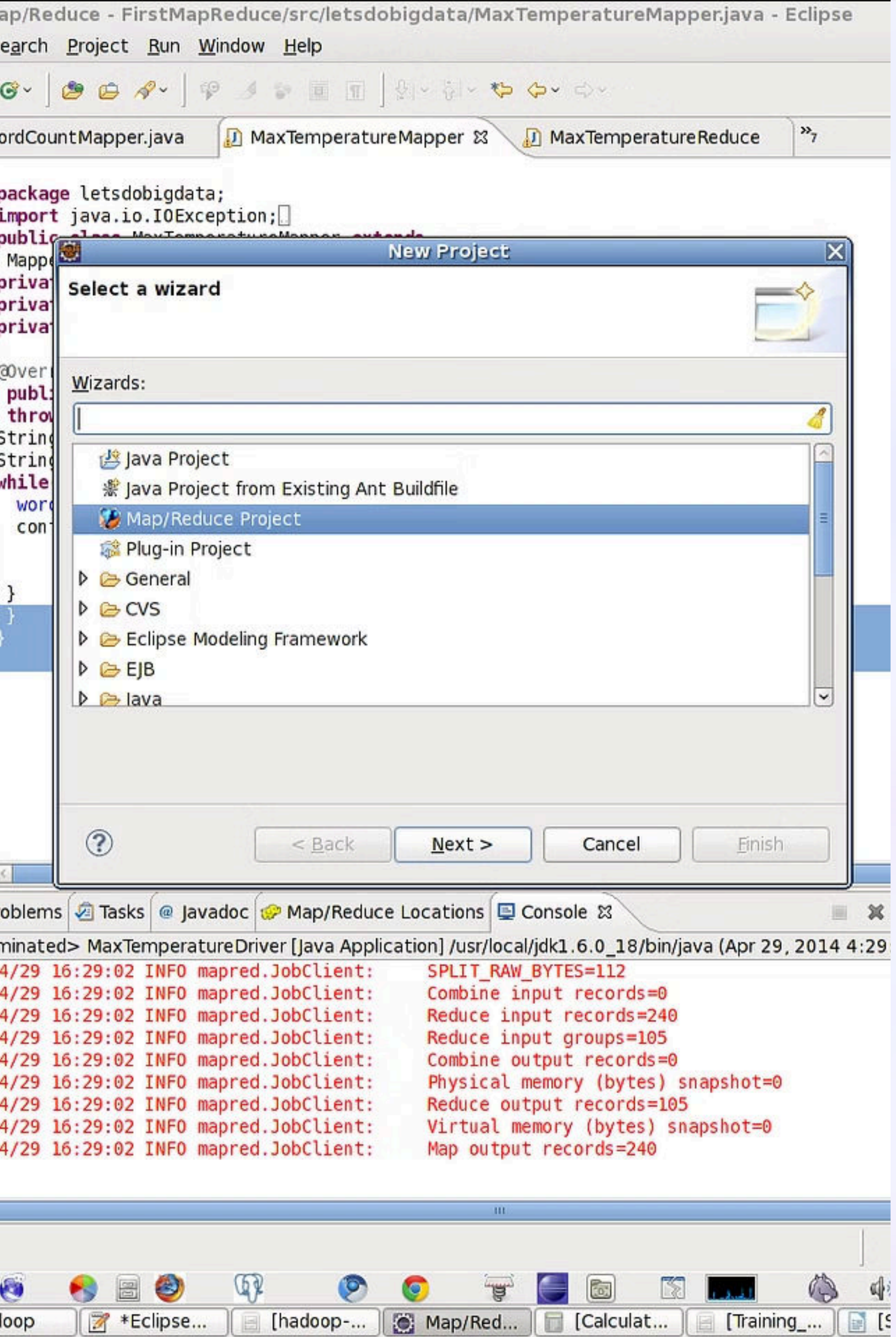


```
package letsdobigdata;
import java.io.IOException;
```

```
minated> MaxTemperatureDriver [Java Application] /usr/local/jdk1.6.0_18/bin/java (Apr 29, 2014 4:29
4/29 16:29:02 INFO mapred.JobClient:     SPLIT_RAW_BYTES=112
4/29 16:29:02 INFO mapred.JobClient:     Combine input records=0
4/29 16:29:02 INFO mapred.JobClient:     Reduce input records=240
4/29 16:29:02 INFO mapred.JobClient:     Reduce input groups=105
4/29 16:29:02 INFO mapred.JobClient:     Combine output records=0
4/29 16:29:02 INFO mapred.JobClient:     Physical memory (bytes) snapshot=0
4/29 16:29:02 INFO mapred.JobClient:     Reduce output records=105
4/29 16:29:02 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=0
4/29 16:29:02 INFO mapred.JobClient:     Map output records=240
```

Introduction to MapReduce paradigm in Java

In Java, the MapReduce paradigm, popularized by frameworks like Apache Hadoop, is a powerful model for processing large datasets in parallel.

KA **by Kakumanu Kalyani**

Key concepts for implementing MapReduce

Data Partitioning

Divide input data into smaller independent chunks to be processed in parallel.

Reduce Function

Aggregates the map outputs, typically providing a summary view of the data.

Shuffle and Sort

Organizes the map outputs and prepares them for the reduce stage.

Distributed Computing

Utilizes multiple machines to perform computations on vast amounts of data.

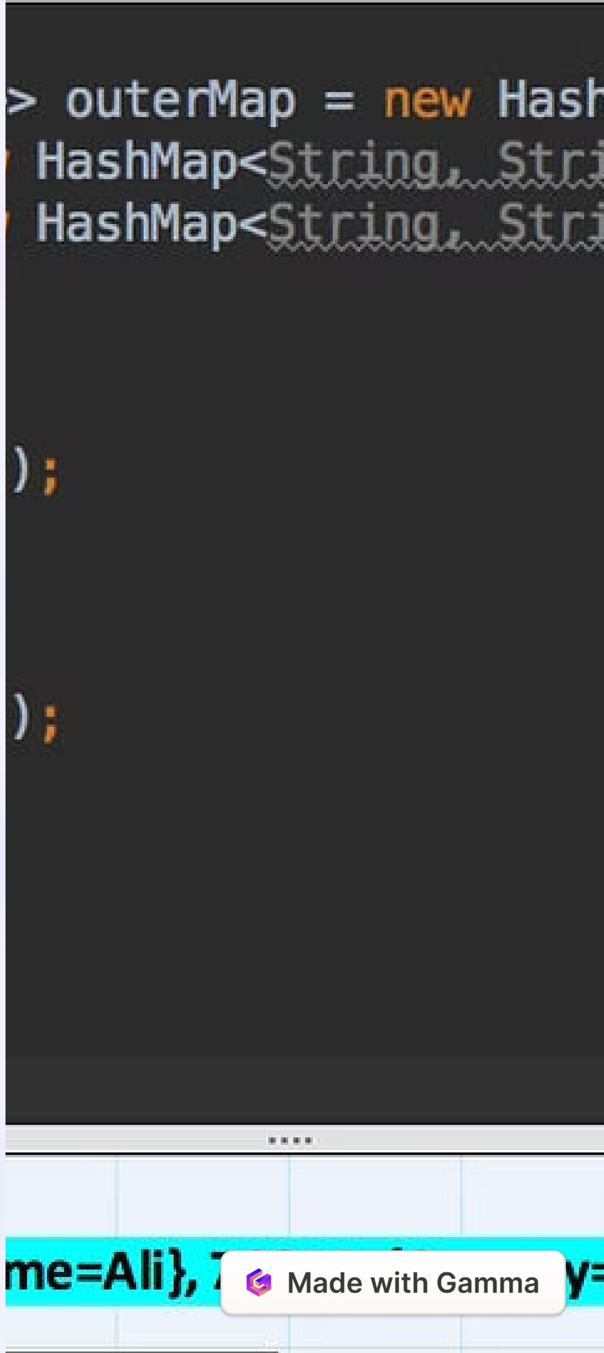
Map Function: definition and purpose

1 What is a Map Function?

The Map function processes input data and transforms it into key-value pairs.

2 Purpose of Map Function

It distributes the processing of data across a set of nodes and produces intermediate key-value pairs.


```
> outerMap = new Hash
 HashMap<String, Stri
 HashMap<String, Stri


);



);
```
me=Ali},

Made with Gamma

How Map Function works in MapReduce

1

Input Data

Raw data is divided into logical partitions called input splits.

2

Map Function Processing

The map function processes each input split and produces intermediate key-value pairs.

3

Context Emission

Key-value pairs generated by the map function are sorted and grouped by key for the reduce stage.

Examples of Map Function in Java

Word Count Example

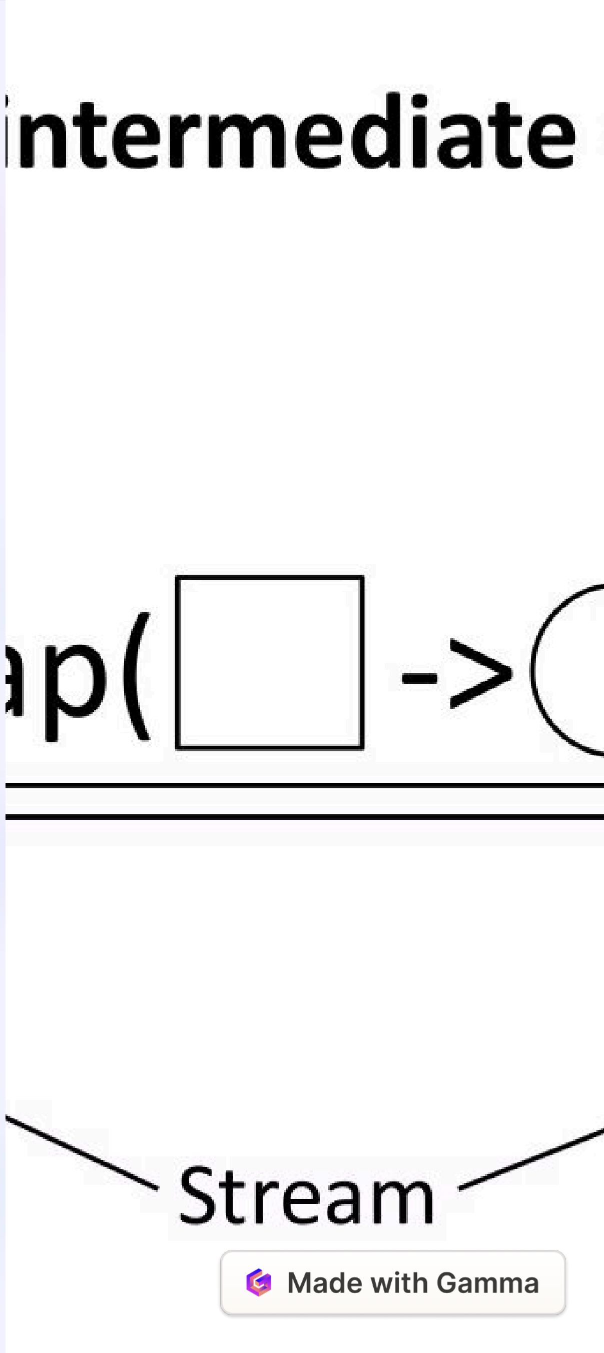
A classic example that demonstrates the fundamental workings of the Map function.

Inverted Index Example

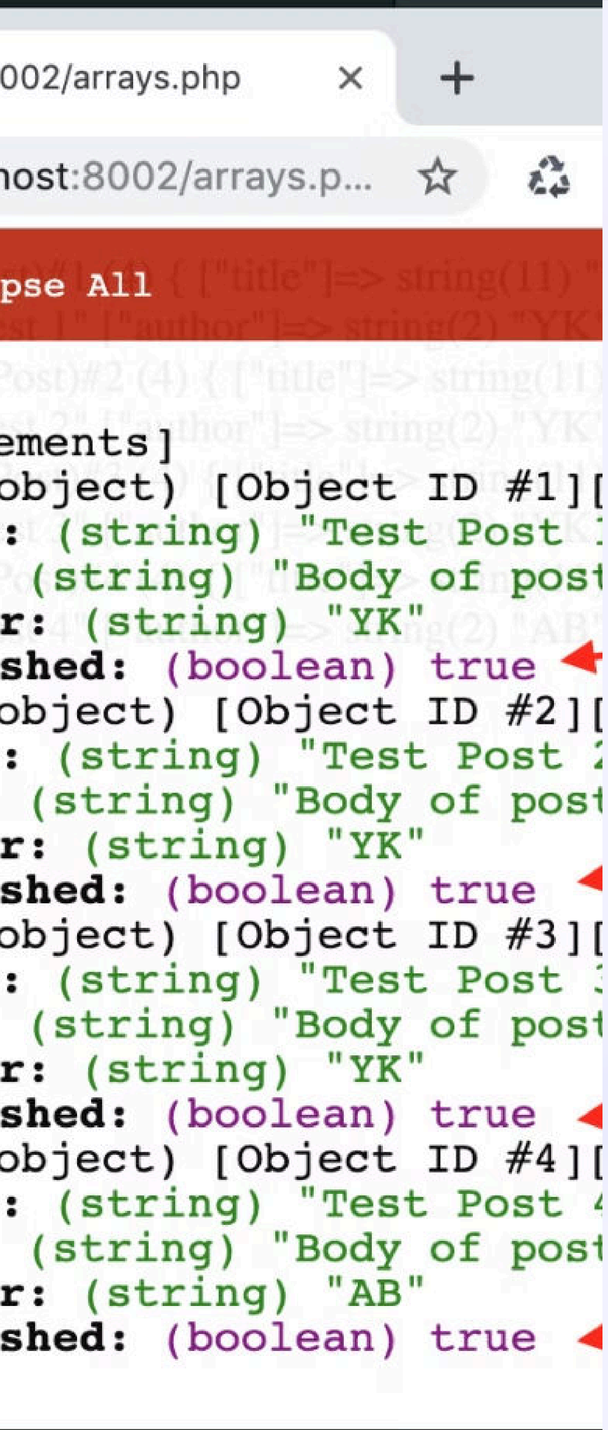
An example that maps words from documents to their locations, enabling efficient search.

Url Access Example

An example mapping URL access logs to analyze website traffic patterns.



Made with Gamma



```
002/arrays.php
host:8002/arrays.p...
pse All
ements]
object) [Object ID #1][
: (string) "Test Post
 (string) "Body of post
r: (string) "YK"
shed: (boolean) true
object) [Object ID #2][
: (string) "Test Post
 (string) "Body of post
r: (string) "YK"
shed: (boolean) true
object) [Object ID #3][
: (string) "Test Post
 (string) "Body of post
r: (string) "YK"
shed: (boolean) true
object) [Object ID #4][
: (string) "Test Post
 (string) "Body of post
r: (string) "AB"
shed: (boolean) true
```

Best practices for writing Map Function

1. **Input Validation**

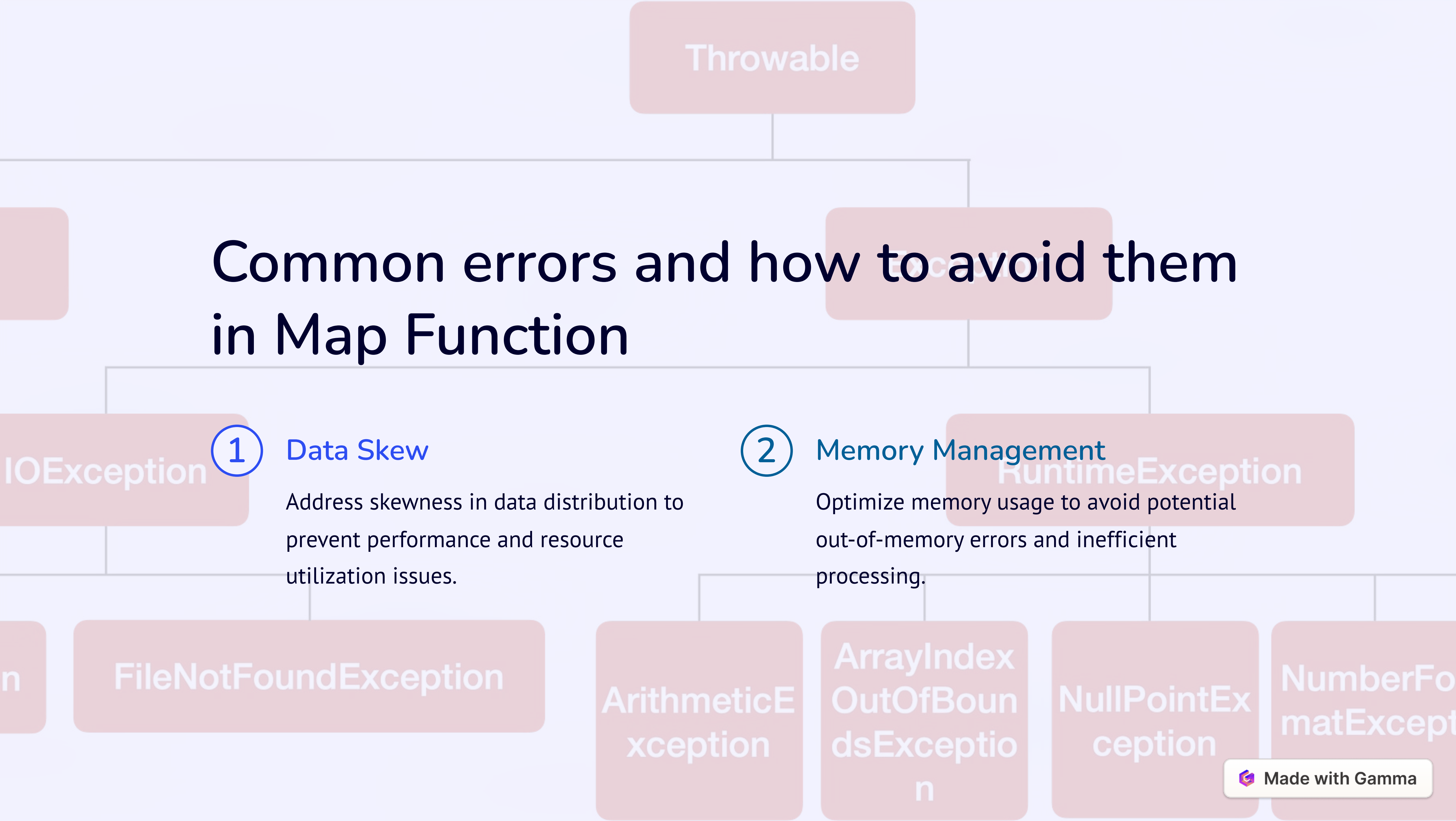
   Ensure that the input data is validated and properly handled for potential errors.

2. **Optimizing Algorithms**

   Choose efficient algorithms and data structures to maximize the performance of the map function.

3. **Error Handling**

   Implement robust error-handling mechanisms to gracefully handle exceptions and failures.



Common errors and how to avoid them in Map Function

1. **Data Skew**

   Address skewness in data distribution to prevent performance and resource utilization issues.

2. **Memory Management**

   Optimize memory usage to avoid potential out-of-memory errors and inefficient processing.

Made with Gamma

Conclusion and next steps

3

Review Practices

Review best practices and identify enhancements to the map function implementation.

25K

Community Engagement

Engage with the broader developer community to gain insights and contribute to MapReduce advancements.