

## could not find implicit value for evidence parameter of type org.apache.flink.api.common.typeinfo.TypeInformation[Int]

大多数刚刚使用Apache Flink的人很可能在编译写好的程序时遇到如下的错误：

```
Error:(15, 26) could not find implicit value for evidence parameter of type org.apache.flink.api.common.typeinfo.TypeInformation[Int]
  socketStockStream.map(_toInt).print()
                        ^
```



如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

完整的代码如下：

```
package com.iteblog.streaming

import org.apache.flink.streaming.api.scala.{DataStream, StreamExecutionEnvironment}

////////////////////////////////////
User: 过往记忆
Date: 2017-03-01
Time: 22:59
blog: https://www.iteblog.com
本文地址：https://www.iteblog.com/archives/2047
过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
过往记忆博客微信公共帐号：iteblog_hadoop
////////////////////////////////////
```

```
object Iteblog{
  def main(args: Array[String]) {
    val env = StreamExecutionEnvironment.getExecutionEnvironment
    val socketStockStream:DataStream[String] = env.socketTextStream("www.iteblog.com", 9999
  )
    socketStockStream.map(_.toInt).print()
    env.execute("Stock stream")
  }
}
```

这种异常的发生通常是因为程序需要一个隐式参数(implicit parameter)，我们可以看看上面程序用到的 map 在Flink中的实现：

```
def map[R: TypeInformation](fun: T => R): DataStream[R] = {
  if (fun == null) {
    throw new NullPointerException("Map function must not be null.")
  }
  val cleanFun = clean(fun)
  val mapper = new MapFunction[T, R] {
    def map(in: T): R = cleanFun(in)
  }

  map(mapper)
}
```

在 map 的定义中有个 [R: TypeInformation]，但是我们程序并没有指定任何有关隐式参数的定义，这时候编译代码无法创建TypeInformation，所以出现上面提到的异常信息。解决这个问题有以下两种方法

(1)、我们可以直接在代码里面加上以下的代码：

```
implicit val typeInfo = TypeInformation.of(classOf[Int])
```

然后再去编译代码就不会出现上面的异常。

(2)、但是这并不是Flink推荐我们去做的，推荐的做法是在代码中引入一下包：

```
import org.apache.flink.streaming.api.scala._
```

如果数据是有限的（静态数据集），我们可以引入以下包：

```
import org.apache.flink.api.scala._
```

然后即可解决上面的异常信息。

本博客文章除特别声明，全部都是原创！  
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。  
本文链接: **【】**（**）**