package com.twitter.tsp.utils

import com.twitter.bijection.Injection

import java.io.ByteArrayInputStream

import java.io.ByteArrayOutputStream

import java.io.ObjectInputStream

import java.io.ObjectOutputStream

import java.io.Serializable

import scala.util.Try

/\*\*

\* @tparam T must be a serializable class

\*/

case class SeqObjectInjection[T <: Serializable]() extends Injection[Seq[T], Array[Byte]] {

override def apply(seq: Seq[T]): Array[Byte] = {

val byteStream = new ByteArrayOutputStream()

val outputStream = new ObjectOutputStream(byteStream)

outputStream.writeObject(seq)

outputStream.close()

byteStream.toByteArray

}

override def invert(bytes: Array[Byte]): Try[Seq[T]] = {

Try {

val inputStream = new ObjectInputStream(new ByteArrayInputStream(bytes))

val seq = inputStream.readObject().asInstanceOf[Seq[T]]

inputStream.close()

seq

}

}

}