package com.twitter.servo.repository

import scala.collection.SeqProxy

/\*\*

\* RichQuery is a mixin trait for KeyValueRepository query objects that are more complex

\* than Seq[K]. It extends SeqProxy to satisfy servo's requirements but provides Product-based

\* implementations of equals and toString. (The query object is expected to be a case class

\* and therefore implement Product.)

\*/

trait RichQuery[K] extends SeqProxy[K] with Product {

// Compare to other RichQuery instances via Product; otherwise allow any sequence to

// match our proxied Seq (thereby matching the semantics of a case class that simply

// extends SeqProxy).

override def equals(any: Any) = {

any match {

case null => false

case other: RichQuery[\_] =>

(

this.productArity == other.productArity &&

this.productIterator.zip(other.productIterator).foldLeft(true) {

case (ok, (e1, e2)) =>

ok && e1 == e2

}

)

case other => other.equals(this)

}

}

// Produce reasonable string for testing

override def toString = "%s(%s)".format(this.productPrefix, this.productIterator.mkString(","))

}