package com.twitter.product\_mixer.core.feature.featuremap.asyncfeaturemap

import com.fasterxml.jackson.core.JsonGenerator

import com.fasterxml.jackson.databind.JsonSerializer

import com.fasterxml.jackson.databind.SerializerProvider

/\*\*

\* Since an [[AsyncFeatureMap]] is typically incomplete, and by the time it's serialized, all the [[com.twitter.product\_mixer.core.feature.Feature]]s

\* it will typically be completed and part of the Query or Candidate's individual [[com.twitter.product\_mixer.core.feature.Feature]]s

\* we instead opt to provide a summary of the Features which would be hydrated using [[AsyncFeatureMap.features]]

\*

\* This indicates which [[com.twitter.product\_mixer.core.feature.Feature]]s will be ready at which Steps

\* and which [[com.twitter.product\_mixer.core.functional\_component.feature\_hydrator.FeatureHydrator]]

\* are responsible for those [[com.twitter.product\_mixer.core.feature.Feature]]

\*

\* @note changes to serialization logic can have serious performance implications given how hot the

\* serialization path is. Consider benchmarking changes with [[com.twitter.product\_mixer.core.benchmark.AsyncQueryFeatureMapSerializationBenchmark]]

\*/

private[asyncfeaturemap] class AsyncFeatureMapSerializer() extends JsonSerializer[AsyncFeatureMap] {

override def serialize(

asyncFeatureMap: AsyncFeatureMap,

gen: JsonGenerator,

serializers: SerializerProvider

): Unit = {

gen.writeStartObject()

asyncFeatureMap.features.foreach {

case (stepIdentifier, featureHydrators) =>

gen.writeObjectFieldStart(stepIdentifier.toString)

featureHydrators.foreach {

case (hydratorIdentifier, featuresFromHydrator) =>

gen.writeArrayFieldStart(hydratorIdentifier.toString)

featuresFromHydrator.foreach(feature => gen.writeString(feature.toString))

gen.writeEndArray()

}

gen.writeEndObject()

}

gen.writeEndObject()

}

}