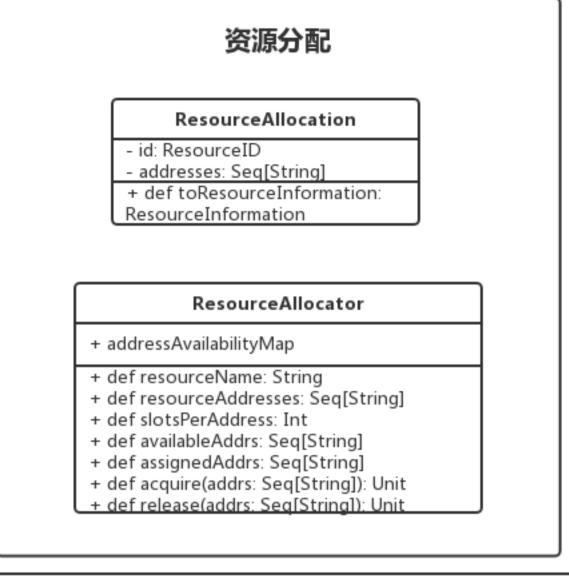
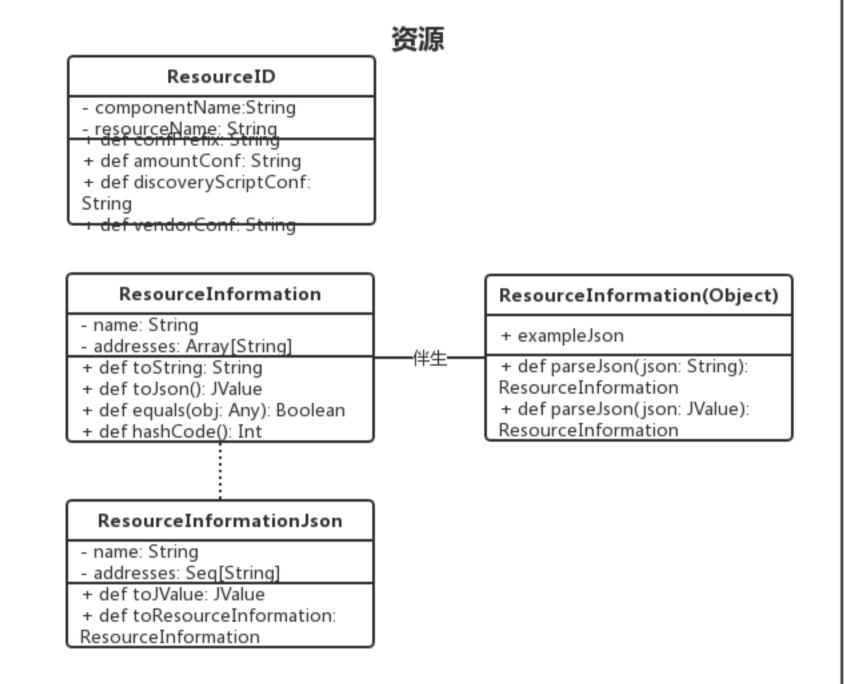


ResourceProfile + _id=ResourceProfile.getNextProfileId + _taskResources:HashMap + _executorResources:HashMap + defid: Int = _id + def taskResources: Map[String, TaskResourceRequest] + def executorResources: Map[String, ExecutorResourceRequest] + def taskResourcesJMap: JMap[String, TaskResourceRequest] + def executorResourcesJMap: JMap[String, ExecutorResourceRequest] + def reset(): Unit + def require(requests: ExecutorResourceRequests): this.type + def require(requests: TaskResourceRequests): this.type def toString(): String ResourceProfile(Object) + UNKNOWN_RESOURCE_PROFILE_ID=-1 + DEFAULT_RESOURCE_PROFILE_ID + CPUS="cpus + CORES="cores" + MEMORY="memory" + OVERHEAD_MEM="memoryOverhead" + PYSPARK_MEM="pyspark.memory" + nextProfileId + defaultProfileRefnew + def getNextProfileId: Int + def resetDefaultProfile(conf: SparkConf): Unit + def getOrCreateDefaultProfile(conf: SparkConf): ResourceProfile + def addDefaultTaskResources(rprof: ResourceProfile, conf: SparkConf): Unit + def addDefaultExecutorResources(rprof: ResourceProfile, conf: SparkConf): Unit





ResourceUtils

- + DISCOVERY_SCRIPT
- + VENDOR
- + AMOUNT
- + GPU + FPGA
- + RESOURCE_PREFIX
- + RESOURCE_DOT
- + def parseResourceRequest(sparkConf: SparkConf, resourceId: ResourceID): ResourceRequest
- + def listResourceIds(sparkConf: SparkConf, componentName: String): Seq[ResourceID] + def parseAllResourceRequests(sparkConf: SparkConf,componentName: String): Seq[ResourceRequest]
- + def resourcesMeetRequirements(
- resourcesFree: Map[String, Int],resourceRequirements: Seq[ResourceRequirement]): Boolean
- + def withResourcesJson[T](resourcesFile: String)(extract: String => Seq[T]): Seq[T]
- + def parseAllocatedFromJsonFile(resourcesFile: String): Seq[ResourceAllocation]
- + def parseAllocatedOrDiscoverResources(sparkConf: SparkConf,componentName: String, resourcesFileOpt: Option[String]): Seq[ResourceAllocation]
- + def parseResourceRequirements(sparkConf: SparkConf,
- componentName:String:Seq[ResourceRequirement]
- + def assertResourceAllocationMeetsRequest(allocation: ResourceAllocation,
- request: ResourceRequest): Unit
- + def assertAllResourceAllocationsMeetRequests(allocations: Seq[ResourceAllocation], requests: Seq[ResourceRequest]): Unit
- + def logResourceInfo(componentName: String, resources: Map[String, ResourceInformation])
- + def getOrDiscoverAllResources(sparkConf: SparkConf,componentName: String,
- resourcesFileOpt: Option[String]): Map[String, ResourceInformation] + def discoverResource(resourceRequest: ResourceRequest): ResourceInformation