SDE

- # instance: SDE
 # db: mysqli
 # numQueries: int
 # timeQueries: float
- + instance(db: mysqli) : SDE# __construct(db: mysqli) : SDE+ query(sql: string) : mysql_result+ multiQuery(sql: string) : bool
- + flushDbResults() : void
 + commit() : bool
 + rollback() : bool
 # addQueryTime() : void
- # addQueryTime() : void
 + getStats() : array
- + makeUpsertQuery(table: string, insert: array, update: array) : string
- + makeUdateQuery(table: string, update: array, where: array): string
- + sanitizeString(string: string) : string
- + sanitizeAndEnguoteString(string: string): string

Util

- + quantitiesToReadable(val: int) : string
- + secondsToReadable(fseconds: int) : string

FitParser

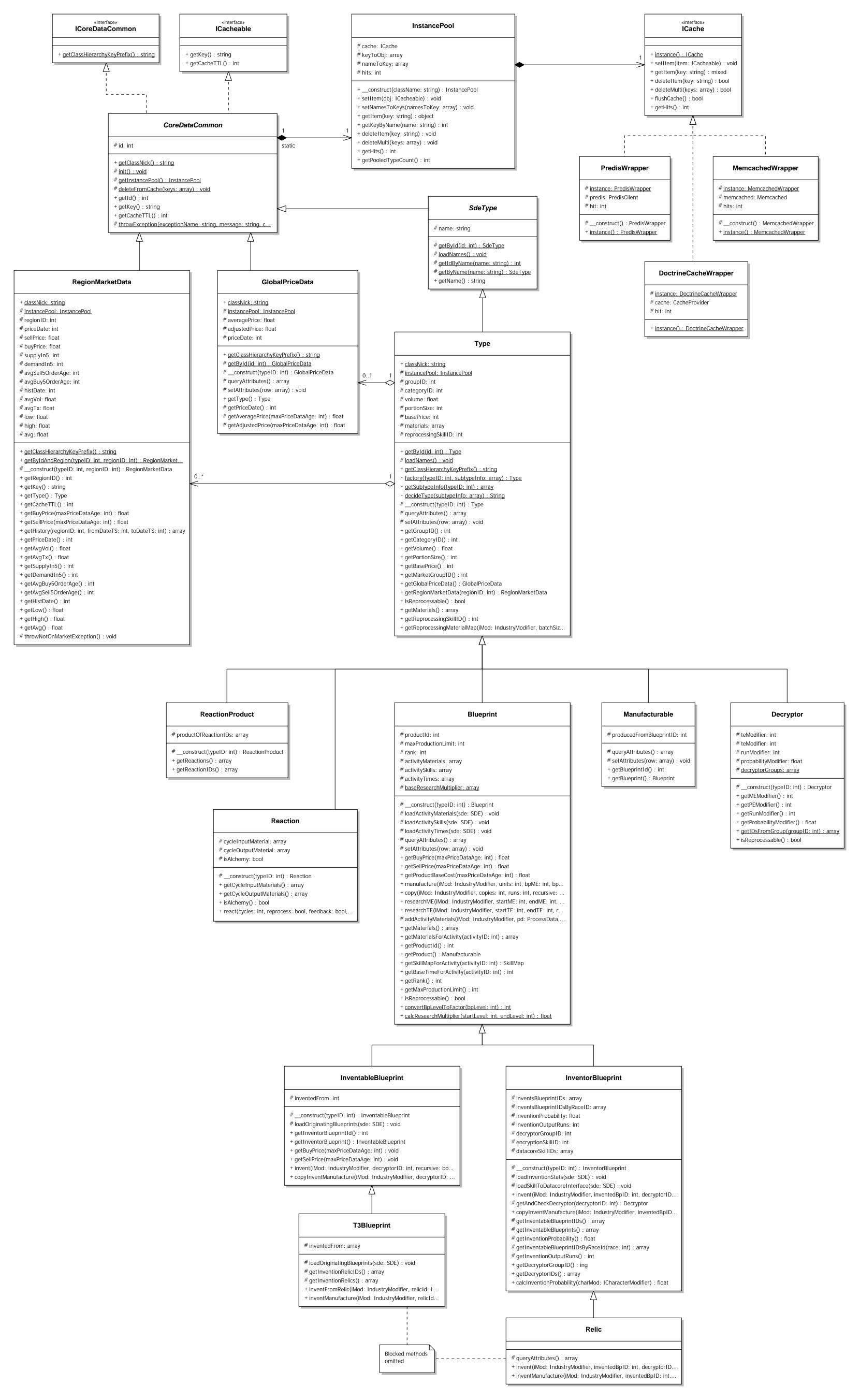
- + parseEftFit(eftFit: string) : MaterialParseResult
- + parseXmlFit(fitDom: DOMDocument) : MaterialParseResult
- + <u>parseScanResult(scanResult: string): MaterialParseResult</u>

Config

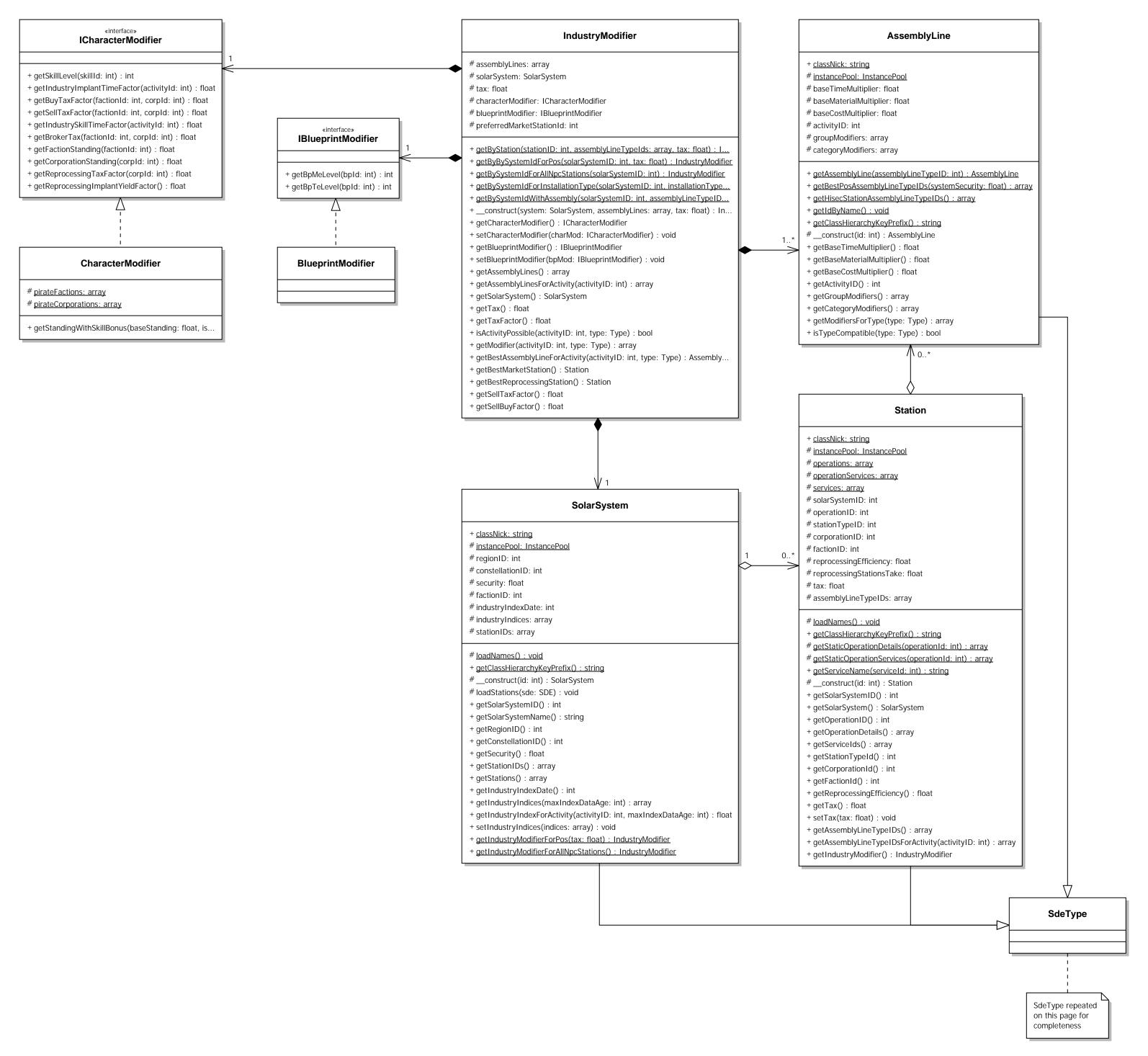
- # sdeDbHost: string
- # sdeDbPort: int
- # sdeDbUser: string
- # sdeDbPw: string
- # sdeDbName: string
- # iveeDbName: string
- # cachePrefix: string
- # cacheHost: string
- # cachePort: int
- # emdrRelayUrl: string
- # crestBaseUrl: string
- # applicationName: string
- # defaultMarketRegionId: int
- # trackedMarketRegionIds: array
- # classes: array
- + getUserAgent(): string
- + getIveeClassName(classNickname: string) : string

trivial getters and setters omitted.

Exception classes omitted.



ReactionProcessData **ProcessData** # inputMaterialMap: MaterialMap # activity: int # outputMaterialMap: MaterialMap # producesTypeID: int # cycles: float # producesQuantity: int # withRefining: bool # processTime: int # withFeedback: bool # processCost: float # assemblyLineID: int + __construct(inputMaterialMap: MaterialMap, outputMaterialMap:... # solarSystemID: int + geInputMaterialMap(): MaterialMap # skills: SkillMap + getOutputMaterialMap() : MaterialMap # materials: MaterialMap + getCycles() : float # subProcessData: array + getTime() : float + withRefining(): bool + __construct(producesTypeID: int, producesQuantity: int... + withFeedback() : bool + addMaterial(typeID: int, amount: int) : void + getInputBuyCost(iMod: IndustryModifier) : float + addSkill(skillID: int, level: int) : void + getOutputSellValue(iMod: IndustryModifier) : float + addSkillMap(sm: SkillMap) : void + getProfit(iMod: IndustryModifier) : float + addSubProcessData(subProcessData: ProcessData) : void + getActivityID(): int + getProducedType() : Type + getNumProducedUnits(): int + getSubProcesses() : array + getProcessCost() : float + getSolarSystemID(): int + getAssemblyLineTypeID(): int + getTotalProcessCost() : float + getMaterialBuyCost(iMod: IndustryModifier) : float + getTotalMaterialBuyCost(iMod: IndustryModifier) : float + getTotalCost(iMod: IndustryModifier) : float + getMaterialMap() : MaterialMap + getTotalMaterialMap(): MaterialMap + getMaterialVolume() : float + getTotalMaterialVolume() : float + getSkillMap() : SkillMap + getTotalSkillMap() : SkillMap + getTime() : int + getTotalTime(): int + getTotalTimes() : array + getTotalProfit(iMod: IndustryModifier) : float + printData(iMod: IndustryModifier) : void \ _{0..}* ResearchMEProcessData ManufactureProcessData InventionProcessData # bpMeLevel: int # probability: float # startMELevel: int # endMELevel: int # bpPeLevel: int # resultRuns: int # resultME: int + __construct(producesTypeID: int, producesQua... + __construct(researchedBpID: int, researchTime: i... # resultTE: int + getMeLevel() : int + getStartMELevel(): int + __construct(producesTypeID: int, inventionTime: int, proce... + getEndMELevel() : int + getPeLevel(): int + getTotalCostPerUnit(iMod: IndustryModifier) : fl... + getResultRuns(): int + getTotalProfit(iMod: IndustryModifier) : float + getResultME() : int + getResultPE() : int + getProbability() : float + getSuccessTime() : float ResearchTEProcessData + getTotalSuccessTime(): float + getTotalSuccessTimes() : array # startTELevel: int + getSuccessMaterialMap(): MaterialMap # endTELevel: int + getTotalSuccessMaterialMap(): MaterialMap CopyProcessData + getSuccessMaterialVolume(): float + __construct(researchedBpID: int, researchTime: i... + getTotalSuccessMaterialVolume(): float + getStartTELevel(): int # outputRuns: int + getProcessCost() : float + getEndTELevel(): int + getSuccessProcessCost() : float + __construct(bpCopyID: int, copyQuantity: int, o... + getTotalSuccessProcessCost(): float + getOutputRuns(): int + getSuccessMaterialBuyCost(iMod: IndustryModifier) : float + getTotalSuccessMaterialBuyCost(iMod: IndustryModifier) : fl... + getTotalSuccessCost(iMod: IndustryModifier) : float



MaterialMap

materials: array

+ addMaterial(typeID: int, quatity: int) : void

+ addMaterials(materials: array): void

+ subtractMaterial(typeID: int, quantity: int): void

+ symmetricDifference(m1: MaterialMap, m2: MaterialMap) : void

+ addMaterialMap(materials: MaterialMap) : void

+ getMaterials() : array

+ getMultipliedMaterialMap(factor: float) : MaterialMap

+ reprocessMaterials(iMod: IndustryModifier): void

+ getMaterialVolume() : float

+ getMaterialBuyCost(iMod: IndustryModifier, maxPriceDataAge...

+ getMaterialSellValue(iMod: IndustryModifier, maxPriceDataAg...

MaterialParseResult

unparseables: array

+ addUnparseable(unparseable: string) : void

+ getUnparseables() : array

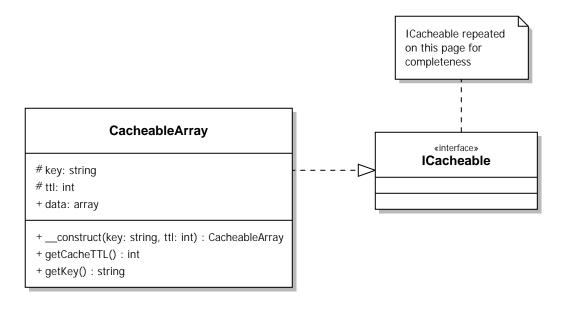
SkillMap

SkillMap: array

+ sanityCheckSkillLevel(skillLevel: int) : bool

+ addSkill(skillID: int, level: int) : void + addSkillMap(skillMap: SkillMap) : void

+ getSkills() : array



EmdrConsumer

- # instance: EmdrConsumer
- # trackedTypeIDs: array
- # trackedMarketRegionIDs: array
- # regions: array # sde: SDE
- # cache: ICache
- # emdrPriceUpdateClass: string
 # emdrHistoryUpdateClass: string
- + instance() : EmdrConsumer
- # __construct() : EmdrConsumer
- + run() : void
- # handleMarketData(marketData: stdClass) : void
- # filterData(typeID: int, regionID: int, generatedAt: int): bool
- # getTimestamps() : array
- # getTimestampsDB() : array
- # updateCaches(): void
- # handleOrderData(): void
- # handleHistoryData(): void
- + getTypeNameByID() : string
- + getRegionNameByID() : string

CrestFetcher

- $\# \ baseUrl: \ string$
- # userAgent: string
- + __construct() : Fetcher
- + getCrestData(path: string, representationName: string): stdClass
- $\#\, curl Get Json (url:\, string)\,:\, std Class$
- # parseContentTypeToRepresentation(contentType: string) : string

EmdrPriceUpdate

- # typeID: int
- # regionID: int
- # generatedAt: int
 # averages: array
- # sell: float
- # avgSell5OrderAge: int
- # buy: float
- # avgBuy5OrderAge: int
- $\#\,demandIn5\colon int$
- # supplyIn5: int
- + __construct(typeID: int, regionID: int, gen...
- # cmp(a: array, b: array) : int
- + insertIntoDB(): void
- # getPriceStats(odata: array, averages: arra...

EmdrHistoryUpdate

- # typeID: int
- # regionID: int
- # generatedAt: int
- # rows: array
- + __construct(typeID: int, regionID: int, gen...
- + insertIntoDB(): void

CrestDataUpdater

- # path: string
- # representationName: string
- # data: stdClass
- # updatedIDs: array
- $+ \underline{\hspace{1.5cm}} construct (data: stdClass): CrestDataUpdater \\$
- + insertIntoDB(): void
- + process DataItem To SQL (item: stdClass): string
- # invalidateCaches(): void
- + doUpdate(): void

IndustryFacilitiesUpdater IndustrySystemsUpdater MarketPricesUpdater