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
- + getStats() : array
- + makeUpsertQuery(table: string, insert: array, update: array) :...
- + makeUdateQuery(table: string, update: array, where: array) :...
- + sanitizeString(string: string): string
- + sanitizeAndEnquoteString(string: string): string

Util

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

FitParser

- + parseEftFit(eftFit: string) : MaterialParseResult
- + parseXmlFit(fitDom: DOMDocument) : MaterialParseResult
- + parseScanResult(scanResult: string): MaterialParseResult

Config

- # sdeDbHost: string
- # sdeDbPort: int
- # sdeDbUser: string
- # sdeDbPw: string
- # sdeDbName: string
- # iveeDbName: string
- # useCache: bool
- # cachePrefix: string
- # cacheHost: string
- # cachePort: int
- # predisConnectionString: string
- # emdrRelayUrl: string
- # crestBaseUrl: string
- # userAgent: string
- # classes: array
- + <u>getIveeClassName(classNickname: string): string</u>

trivial getters omitted.

Exception classes omitted.

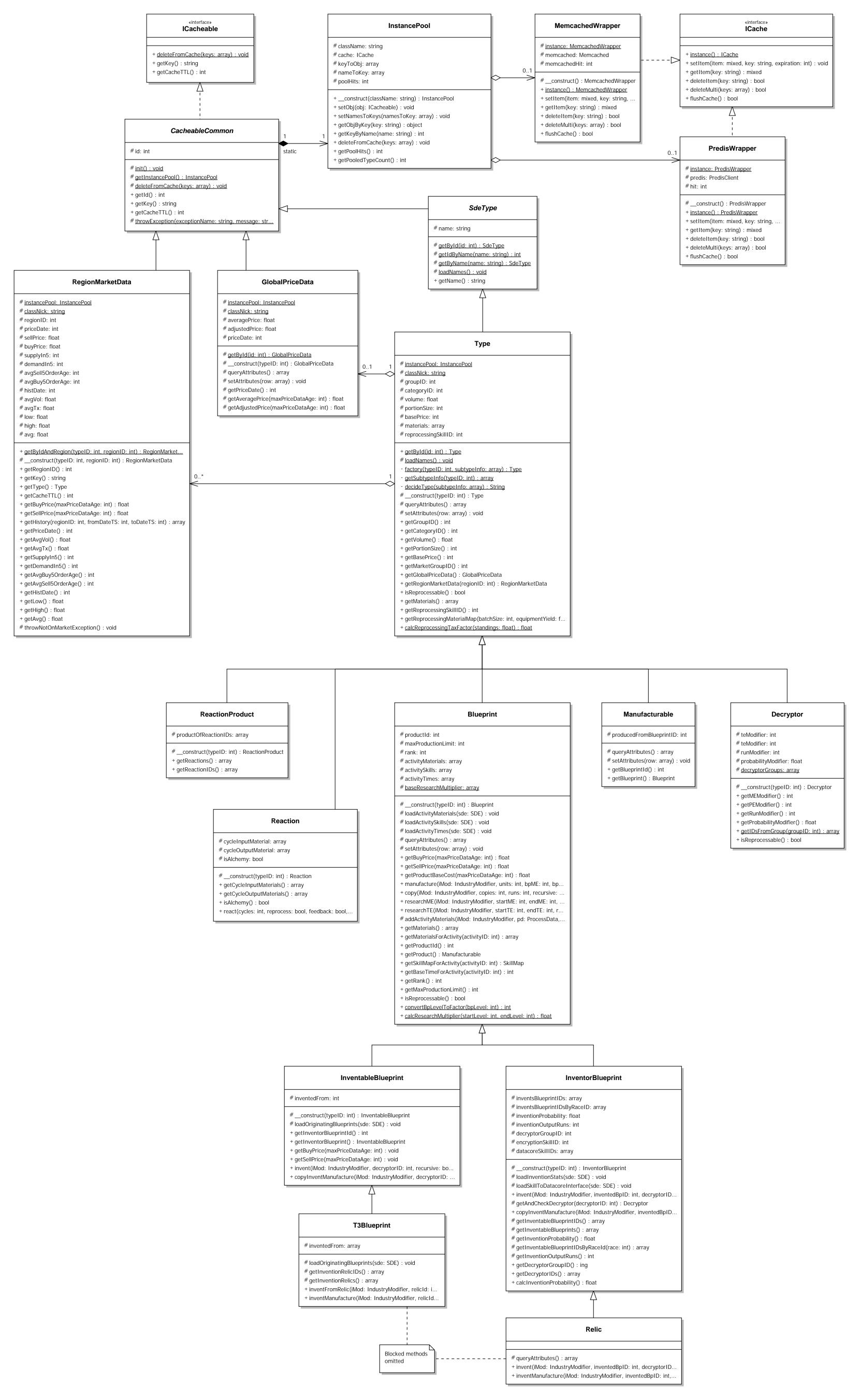
Defaults

- # instance: Defaults
- # defaultRegionID: int
- # trackedMarketRegionIDs: array
- # defaultBuyTaxFactor: float
- # defaultSellTaxFactor: float
- # defaultBpoMe: int
- # defaultBpoMe: int
- # maxPriceDataAge: int
- # bpMeLevels: array
- # bpTeLevels: array
- + instance() : Defaults
- + getSkillLevel(skillID: int) : int
- + getIndustryImplantTimeModifiers(): array
- + setBpMeLevel() : void
- + setBpTeLevel(): void

MyDefaults

trivial getters ommited.

For user implementation. Also serves as example for the intended way of extending iveeCore



activity: int # producesTypeID: int # producesQuantity: int # processTime: int # processCost; float

producesQuantity: int
processTime: int
processCost: float
assemblyLineID: int
solarSystemID: int
teamID: int
skills: SkillMap
materials: MaterialMap

subProcessData: array

- + __construct(producesTypeID: int, producesQuantity: int...
- + addMaterial(typeID: int, amount: int) : void
- + addSkill(skillID: int, level: int) : void
- + addSkillMap(sm: SkillMap) : void
- + addSubProcessData(subProcessData: ProcessData) : void
- + getActivityID() : int + getProducedType() : Type + getNumProducedUnits() : int + getSubProcesses() : array + getProcessCost() : float
- + getSolarSystemID() : int
- + getAssemblyLineTypeID() : int + getTeamID() : int
- + getTotalProcessCost() : float
- + getMaterialBuyCost(maxPriceDataAge: int, regionId: int...
- + getTotalMaterialBuyCost(maxPriceDataAge: int, regionI...
- + getTotalCost(maxPriceDataAge: int, regionId: int): float
- + getMaterialMap(): MaterialMap
- + getTotalMaterialMap(): MaterialMap
- + getMaterialVolume() : float
- + getTotalMaterialVolume() : float
- + getSkillMap() : SkillMap + getTotalSkillMap() : SkillMap
- + getTime() : int + getTotalTime() : int + getTotalTimes() : array
- + getTotalProfit(maxPriceDataAge: int, regionId: int) : float
- + printData(): void

\ _{0..*}

ReactionProcessData

- # inputMaterialMap: MaterialMap
 # outputMaterialMap: MaterialMap
- # cycles: float
 # withRefining: bool
 # withFeedback: bool
- + __construct(inputMaterialMap: MaterialMap, ou...
- + geInputMaterialMap() : MaterialMap
- + getOutputMaterialMap(): MaterialMap
- + getCycles() : float
 + getTime() : float
 + withRefining() : bool
 + withFeedback() : bool
- + getInputBuyCost(maxPriceDataAge: int) : float
- + getOutputBuyValue(maxPriceDataAge: int) : float
- + getProfit(maxPriceDataAge: int) : float

ManufactureProcessData

- # bpMeLevel: int
- # bpPeLevel: int
- $+ __construct(producesTypeID:\ int,\ producesQua...$
- + getMeLevel() : int
- + getPeLevel() : int
- + getSlotCost() : float
- + getTotalCostPerUnit(maxPriceDataAge: int, regi...
- + getTotalProfit(maxPriceDataAge: int, regionId: i...
- + printData() : void

CopyProcessData

- $\# \ output Runs: \ int$
- + __construct(bpCopyID: int, copyQuantity: int, o...
- + getOutputRuns(): int

InventionProcessData

- # probability: float # resultRuns: int # resultME: int
- # resultTE: int
- + __construct(producesTypeID: int, inventionTime: int, proce...
- + getResultRuns() : int + getResultME() : int
- + getResultPE(): int
- + getProbability() : float + getSuccessTime() : float
- getsuccess time(): noat
- + getTotalSuccessTime() : float
- + getTotalSuccessTimes(): array
- + getSuccessMaterialMap() : MaterialMap
- + getTotalSuccessMaterialMap() : MaterialMap + getSuccessMaterialVolume() : float
- + getTotalSuccessMaterialVolume(): float
- + getProcessCost() : float
- + getSuccessProcessCost() : float
- + getTotalSuccessProcessCost() : float
- + getSuccessMaterialBuyCost(maxPriceDataAge: int, regionId:...
- + getTotalSuccessMaterialBuyCost(maxPriceDataAge: int, regi...
- + getTotalSuccessCost(maxPriceDataAge: int, regionId: int) : ...
- + printData() : void

ResearchMEProcessData

- $\#\, start MELevel \colon int$
- $\#\,endMELevel\colon int$
- $+ \underline{\hspace{0.3cm}} construct (researched BpID: int, \ research Time: i...$
- + getStartMELevel(): int
- + getEndMELevel(): int

Research TEP rocess Data

- # startTELevel: int
- # endTELevel: int
- + __construct(researchedBpID: int, researchTime: i...
- + getStartTELevel(): int
- + getEndTELevel() : int

All these classes except IndustryModifier inherit from SdeType, omitted for readibility

Team

- # instancePool: InstancePool
- # classNick: string
- # solarSystemID: int
- # creationTime: int
- # expiryTime: int
- # activityID: int
- # costModifier: float
- # specialityID: int
- # bonusIDs: array
- # bonusValues: array
- # workerSpecialities: array
- # getIdByName(): void
- # __construct(id: int) : Team
- + getSolarSystemID(): int
- + getCreationTime(): int
- + getExpiryTime(): int
- + getActivityID() : int
- + getCostModifier() : float
- + getSpecialityID(): int
- + getBonusIDs() : array
- + getBonusValues() : array
- + getWorkerSpecialities() : array
- + isTypeCompatible(type: Type) : bool
- + isGroupIDCompatible(groupID: int) : bool
- + getWorkerIDsForGroupID(groupID: int) : array
- + getModifiersForGroupID(groupID: int) : array

Speciality

- # instancePool: InstancePool
- # classNick: string
- ${\it \# speciality Group IDs: array}$
- # getIdByName(): void
- #__construct(id: int) : Speciality
- + getSpecialityGroupIDs() : array
- + appliesToGroupID(groupID: int): bool

IndustryModifier

- # assemblyLines: array
- # solarSystem: SolarSystem
- # tax: float
- # teams: array
- # skillTimeModifiers: array # implantTimeModifiers: array
- + getByNpcStation(stationID: int) : IndustryModifier
- + getByBySystemIdForPos(solarSystemID: int, tax: float) : IndustryModifier
- + getBySystemIdForAllNpcStations(solarSystemID: int) : IndustryModifier
- + <u>getBySystemIdWithAssembly(solarSystemID: int, assemblyLineTypeID...</u> + <u>__construct(system: SolarSystem, assemblyLines: array, teams: array, ...</u>
- + getAssemblyLines() : array
- + getAssemblyLinesForActivity(activityID: int) : array
- + getSolarSystem() : SolarSystem
- + getTax() : float

0..*

- + getImplantTimeModifiers() : array
- + getImplantTimeModifierForActivity(activityID: int) : float
- + setImplantTimeModifierForActivity(modifier: float, activityID: int) : void
- + setImplantTimeModifiers(modifiers: array) : void
- + getSkillTimeModifiers() : array
- + getSkillTimeModifierForActivity(activityID: int): float
- + setSkillTimeModifiers(modifiers: array) : void
- + setSkillTimeModifierForActivity(modifier: float, activityID: int) : void
- + getTeams() : array
- + getTeamsForActivity(activityID: int) : array
- + setTeams(teams: array) : void
- + setTeamsForActivity(teams: array, activityID: int): void
- + isActivityPossible(activityID: int, type: Type) : bool
- + getModifier(activityID: int, type: Type) : array
- + getBestAssemblyLineForActivity(activityID: int, type: Type): Assembly...
- $+ getBestTeamForActivity(activityID: int, \ type: \ Type): Team$

AssemblyLine

- # instancePool: InstancePool
- # classNick: string
- # baseTimeMultiplier: float
- ${\it \#}\ base Material Multiplier:\ float$
- # baseCostMultiplier: float
- # activityID: int

1..*

- # groupModifiers: array
 # categoryModifiers: array
- + getAssemblyLine(assemblyLineTypeID: int) : AssemblyLine
- + getBestPosAssemblyLineTypeIDs(systemSecurity: float) : array
- + getHisecStationAssemblyLineTypeIDs(): array
- + getIdByName() : void
- #__construct(id: int) : AssemblyLine
- + getBaseTimeMultiplier() : float
- + getBaseMaterialMultiplier() : float
- + getBaseCostMultiplier() : float
- + getActivityID() : int
- + getGroupModifiers() : array
- + get Category Modifiers (): array
- + getModifiersForType(type: Type) : array
- $+\ is Type Compatible (type:\ Type):\ bool$

SolarSystem

- # instancePool: InstancePool
- # classNick: string
- # regionID: int
- # constellationID: int
- # security: float
- # factionID: int
- # industryIndexDate: int
- # industryIndices: array
- # stationIDs: array
- # teamIDs: array

loadNames() : void

- #__construct(id: int) : SolarSystem
- + getSolarSystemID(): int
- + getSolarSystemName() : string
- + getRegionID(): int
- $+ \ getConstellation ID() : int$
- + getSecurity() : float
- + getStationIDs() : array
- + getStations() : array + getTeamIDs() : array
- + getTeams() : array
- + gerreams(): array
- + getIndustryIndexDate() : int + getIndustryIndices(maxIndexDataAge: int) : array
- + getIndustryIndexForActivity(activityID: int, maxIndexDataAge: int) : float
- + setIndustryIndices(indices: array) : void
- + getIndustryModifierForPos(tax: float) : IndustryModifier
- + getIndustryModifierForAllNpcStations(): IndustryModifier

Station

- # instancePool: InstancePool
- # classNick: string
- # solarSystemID: int
- # operationID: int
- # stationTypeID: int
- # corporationID: int
- # reprocessingEfficiency: float
- # tax: float
- # assemblyLineTypeIDs: array
- # loadNames() : void
- #__construct(id: int) : Station
- + getSolarSystemID(): int
- + getSolarSystem() : SolarSystem
 + getOperationID() : int
- + getCorporationID() : int
- + getReprocessingEfficiency(): float
- + getTax() : float
- + getAssemblyLineTypeIDs(): array
- + getAssemblyLineTypeIDsForActivity(activityID: int) : array
- + getIndustryModifier(): IndustryModifier

MaterialMap

materials: array

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

+ addMaterials(materials: array): void

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

+ <u>symmetricDifference(m1: MaterialMap, m2: MaterialMap) : void</u>

+ addMaterialMap(materials: MaterialMap) : void

+ getMaterials(): array

+ getMultipliedMaterialMap (factor: float): MaterialMap

+ reprocessMaterials(equipmentYield: float, reprocessingTaxFa...

+ getMaterialVolume() : float

+ getMaterialBuyCost(maxPriceDataAge: int, regionId: int) : float

+ getMaterialSellValue(maxPriceDataAge: int, regionId: int) : fl...



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 EmdrPriceUpdate EmdrHistoryUpdate # typeID: int # typeID: int # instance: EmdrConsumer # regionID: int # trackedTypeIDs: array # regionID: int # trackedMarketRegionIDs: array # generatedAt: int # generatedAt: int # regions: array # averages: array # rows: array # sde: SDE # sell: float + __construct(typeID: int, regionID: int, gen... # cache: ICache # avgSell5OrderAge: int + insertIntoDB(): void # emdrPriceUpdateClass: string # buy: float # emdrHistoryUpdateClass: string # avgBuy5OrderAge: int # demandIn5: int + instance() : EmdrConsumer # supplyIn5: int # __construct() : EmdrConsumer + run(): void + __construct(typeID: int, regionID: int, gen... # handleMarketData(marketData: stdClass) : void # cmp(a: array, b: array) : int # filterData(typeID: int, regionID: int, generatedAt: int) : bool + insertIntoDB(): void # getTimestamps() : array # getPriceStats(odata: array, averages: arra... # getTimestampsDB(): array # updateCaches(): void # handleOrderData(): void # handleHistoryData(): void + getTypeNameByID() : string + getRegionNameByID() : string CrestDataUpdater CrestFetcher # path: string # baseUrl: string # representationName: string # userAgent: string # data: stdClass # updatedIDs: array + __construct() : Fetcher + getCrestData(path: string, representationName: string) : stdClass + __construct(data: stdClass) : CrestDataUpdater # curlGetJson(url: string) : stdClass + insertIntoDB(): void # parseContentTypeToRepresentation(contentType: string) : string + processDataItemToSQL(item: stdClass) : string # invalidateCaches(): void + doUpdate() : void **TeamsUpdater IndustryFacilitiesUpdater IndustrySystemsUpdater MarketPricesUpdater SpecialitiesUpdater**