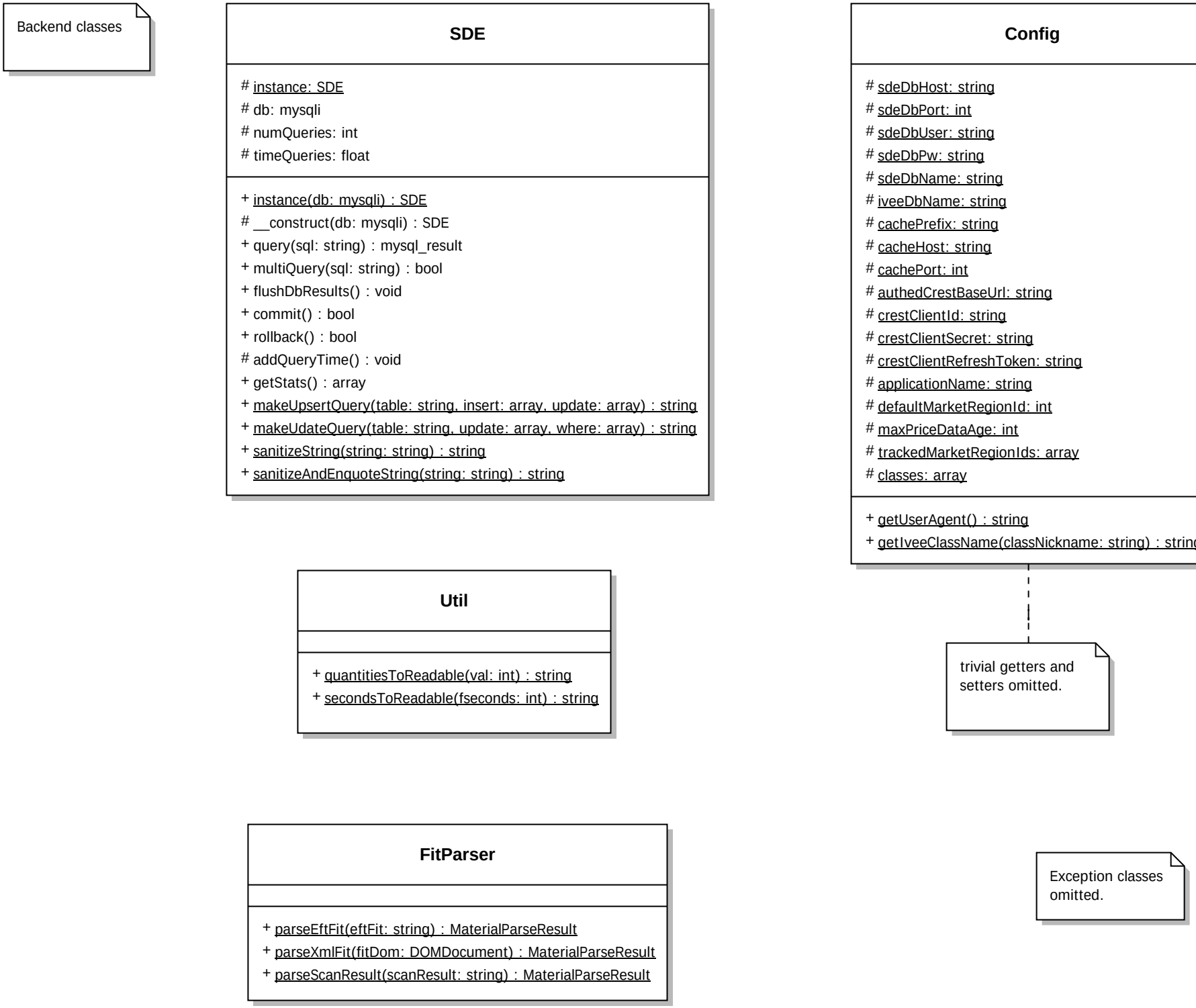

Backend classes
SDE
instance: SDE
db: mysqli
numQueries: int
timeQueries: float
+ instance(db: mysqli) : SDE
# __construct(db: mysqli) : SDE
+ query(sql: string) : mysql_result
+ multiQuery(sql: string) : bool
+ flushDbResults() : void
+ commit() : bool
+ rollback() : bool
# addQueryTime() : void
+ getStats() : array
+ makeUpsertQuery(table: string, insert: array, update: array) : string
+ makeUdateQuery(table: string, update: array, where: array) : string
+ sanitizeString(string: string) : string
+ sanitizeAndEnquoteString(string: string) : string
Config
sdeDbHost: string
sdeDbPort: int
sdeDbUser: string
sdeDbPw: string
sdeDbName: string
iveeDbName: string
cachePrefix: string
cacheHost: string
cachePort: int
authedCrestBaseUrl: string
crestClientId: string
crestClientSecret: string
crestClientRefreshToken: string
applicationName: string
defaultMarketRegionId: int
maxPriceDataAge: int
trackedMarketRegionIds: array
classes: array
+ getUserAgent() : string
+ getIveeClassName(classNickname: string) : string
Util
+ quantitiesToReadable(val: int) : string
+ secondsToReadable(fseconds: int) : string
trivial getters and setters omitted.
FitParser
+ parseEftFit(eftFit: string) : MaterialParseResult
+ parseXmlFit(fitDom: DOMDocument) : MaterialParseResult
+ parseScanResult(scanResult: string) : MaterialParseResult
Exception classes omitted.

Type classes

«interface»
ICoreDataCommon

+ getClassHierarchyKeyPrefix() : string

«interface»
ICacheable

+ getKey() : string
+ getCacheExpiry() : int

InstancePool

cache: ICache
keyToObj: array
nameToKey: array
hits: int

+ __construct(className: string) : InstancePool
+ setItem(obj: ICacheable) : void
+ setNamesToKeys(namesToKey: array) : void
+ getItem(key: string) : object
+ getKeyByName(name: string) : int
+ deleteItem(key: string) : void
+ deleteMulti(keys: array) : void
+ getHits() : int
+ getPooledTypeCount() : int

«interface»
ICache

+ instance() : ICache
+ setItem(item: ICacheable) : void
+ getItem(key: string) : mixed
+ deleteItem(key: string) : bool
+ deleteMulti(keys: array) : bool
+ flushCache() : bool
+ getHits() : int

1
1
static

CoreDataCommon

id: int
expiry: int

+ getClassNick() : string
# init() : void
# getInstancePool() : InstancePool
# deleteFromCache(keys: array) : void
+ getId() : int
+ getKey() : string
+ getCacheExpiry() : int
# throwException(exceptionName: string, message: string, c...

RedisWrapper

instance: RedisWrapper
redis: Redis
hit: int

__construct() : PredisWrapper
+ instance() : PredisWrapper

MemcachedWrapper

instance: MemcachedWrapper
memcached: Memcached
hits: int
hasMultiDelete: bool

__construct() : MemcachedWrapper
+ instance() : MemcachedWrapper

SdeType

name: string

getById(id: int) : SdeType
loadNames() : void
getIdByName(name: string) : int
getByName(name: string) : SdeType
+ getName() : string
# setExpiry() : void

DoctrineCacheWrapper

instance: DoctrineCacheWrapper
cache: CacheProvider
hit: int

+ instance() : DoctrineCacheWrapper
+ setCache(cache: CacheProvider) : void

MarketHistory

+ classNick: string
# instancePool: InstancePool
# crestMarketProcessor: CrestMarketProcessor
# regionId: int
# lastHistUpdate: int
# oldestDate: int
# newestDate: int
# vol: array
# tx: array
# low: array
# high: array
# avg: array
# sell: array
# buy: array
# supplyIn5: array
# demandIn5: array
# avgSell5OrderAge: array
# avgBuy5OrderAge: array

+ getClassHierarchyKeyPrefix() : string
+ getByIdAndRegion(typeId: int, regionId: int, maxPriceDataAg...
# __construct(typeId: int, regionId: int) : MarketHistory
+ getRegionId() : int
+ getKey() : string
+ getType() : Type
+ getLastHistUpdateTs() : int
+ getOldestDate() : int
+ getNewestDate() : int
+ isTooOld(maxPriceDataAge: int) : bool
+ getValuesForDate(dateTs: int) : array
+ getVol() : array
+ getTx() : array
+ getLow() : array
+ getHigh() : array
+ getAvg() : array
+ getSell() : array
+ getBuy() : array
+ getSupplyIn5() : array
+ getDemandIn5() : array
+ getAvgSell5OrderAge() : array
+ getAvgBuy5OrderAge() : array
# throwNotOnMarketException() : void

MarketPrices

+ classNick: string
# instancePool: InstancePool
# crestMarketProcessor: CrestMarketProcessor
# regionId: int
# lastPriceUpdate: int
# lastHistUpdate: int
# sellPrice: float
# buyPrice: float
# supplyIn5: int
# demandIn5: int
# avgSell5OrderAge: int
# avgBuy5OrderAge: int
# avgVol: float
# avgTx: float

+ getClassHierarchyKeyPrefix() : string
+ getByIdAndRegion(typeId: int, regionId: int, maxPriceDataAg...
# __construct(typeId: int, regionId: int, maxPriceDataAge: int, ...
# getDataFromDb() : array
+ getRegionId() : int
+ getKey() : string
+ getLastPriceUpdateTs() : int
+ isTooOld(maxPriceDataAge: int) : bool
+ getLastHistUpdateTs() : int
+ getType() : Type
+ getBuyPrice(maxPriceDataAge: int) : float
+ getSellPrice(maxPriceDataAge: int) : float
+ getSupplyIn5() : int
+ getDemandIn5() : int
+ getAvgBuy5OrderAge() : int
+ getAvgSell5OrderAge() : int
+ getHistDate() : int
+ getAvgVol() : float
+ getAvgTx() : float
# throwNotOnMarketException() : void

GlobalPriceData

+ classNick: string
# instancePool: InstancePool
# averagePrice: float
# adjustedPrice: float
# priceDate: int

+ getClassHierarchyKeyPrefix() : string
# getById(id: int) : GlobalPriceData
# __construct(typeId: int) : GlobalPriceData
# queryAttributes() : array
# setAttributes(row: array) : void
+ getType() : Type
# getPriceDate() : int
# getAveragePrice(maxPriceDataAge: int) : float
# getAdjustedPrice(maxPriceDataAge: int) : float
+ isTooOld(maxPriceDataAge: int) : bool

0..1
1
0..*
1
0..*
1

Type

+ classNick: string
# instancePool: InstancePool
# groupId: int
# categoryId: int
# volume: float
# portionSize: int
# basePrice: int
# materials: array
# reprocessingSkillId: int

+ getById(id: int) : Type
# loadNames() : void
+ getClassHierarchyKeyPrefix() : string
- factory(typeId: int, subtypeInfo: array) : Type
- getSubtypeInfo(typeId: int) : array
- decideType(subtypeInfo: array) : String
__construct(typeId: int) : Type
queryAttributes() : array
setAttributes(row: array) : void
+ getGroupId() : int
+ getCategoryId() : int
+ getVolume() : float
+ getPortionSize() : int
+ getBasePrice() : int
+ getMarketGroupId() : int
+ onMarket() : bool
+ getGlobalPriceData() : GlobalPriceData
+ getMarketPrices(regionId: int) : MarketPrices
+ getMarketHistory(regionId: int) : MarketHistory
+ isReprocessable() : bool
+ getMaterials() : array
+ getReprocessingSkillId() : int
+ getReprocessingMaterialMap(iMod: IndustryModifier, batchSiz...

ReactionProduct

productOfReactionIds: array

__construct(typeId: int) : ReactionProduct
+ getReactions() : array
+ getReactionIds() : array

Reaction

cycleInputMaterial: array
cycleOutputMaterial: array
isAlchemy: bool

__construct(typeId: int) : Reaction
+ getCycleInputMaterials() : array
+ getCycleOutputMaterials() : array
+ isAlchemy() : bool
+ react(cycles: int, reprocess: bool, feedback: bool,...

Blueprint

productId: int
maxProductionLimit: int
rank: int
activityMaterials: array
activitySkills: array
activityTimes: array
productBaseCost: float
baseResearchMultiplier: array

__construct(typeId: int) : Blueprint
loadActivityMaterials(sde: SDE) : void
loadActivitySkills(sde: SDE) : void
loadActivityTimes(sde: SDE) : void
queryAttributes() : array
setAttributes(row: array) : void
+ getBuyPrice(maxPriceDataAge: int) : float
+ getSellPrice(maxPriceDataAge: int) : float
+ getProductBaseCost(maxPriceDataAge: int) : float
+ manufacture(iMod: IndustryModifier, units: int, bpME: int, bp...
+ copy(iMod: IndustryModifier, copies: int, runs: int, recursive: ...
+ researchME(iMod: IndustryModifier, startME: int, endME: int, ...
+ researchTE(iMod: IndustryModifier, startTE: int, endTE: int, r...
# addActivityMaterials(iMod: IndustryModifier, pd: ProcessData,...
+ getMaterials() : array
+ getMaterialsForActivity(activityId: int) : array
+ getProductId() : int
+ getProduct() : Manufacturable
+ getSkillMapForActivity(activityId: int) : SkillMap
+ getBaseTimeForActivity(activityId: int) : int
+ getRank() : int
+ getMaxProductionLimit() : int
+ isReprocessable() : bool
+ convertBpLevelToFactor(bpLevel: int) : int
+ calcResearchMultiplier(startLevel: int, endLevel: int) : float

Manufacturable

producedFromBlueprintId: int

queryAttributes() : array
setAttributes(row: array) : void
+ getBlueprintId() : int
+ getBlueprint() : Blueprint

Decryptor

teModifier: int
teModifier: int
runModifier: int
probabilityModifier: float
decryptorGroups: array

__construct(typeId: int) : Decryptor
+ getMEModifier() : int
+ getPEModifier() : int
+ getRunModifier() : int
+ getProbabilityModifier() : float
+ getIdsFromGroup(groupId: int) : array
+ isReprocessable() : bool

InventableBlueprint

inventedFrom: int

__construct(typeId: int) : InventableBlueprint
loadOriginatingBlueprints(sde: SDE) : void
+ getInventorBlueprintId() : int
+ getInventorBlueprint() : InventableBlueprint
+ getBuyPrice(maxPriceDataAge: int) : void
+ getSellPrice(maxPriceDataAge: int) : void
+ invent(iMod: IndustryModifier, decryptorId: int, recursive: bo...
+ copyInventManufacture(iMod: IndustryModifier, decryptorId: ...

InventorBlueprint

inventsBlueprintIds: array
inventsBlueprintIdsByRaceId: array
inventionProbability: float
inventionOutputRuns: int
decryptorGroupId: int
encryptionSkillId: int
datacoreSkillIds: array

__construct(typeId: int) : InventorBlueprint
loadInventionStats(sde: SDE) : void
loadSkillToDatacoreInterface(sde: SDE) : void
+ invent(iMod: IndustryModifier, inventedBpId: int, decryptorId...
# getAndCheckDecryptor(decryptorId: int) : Decryptor
+ copyInventManufacture(iMod: IndustryModifier, inventedBpId...
# getInventableBlueprintIds() : array
# getInventableBlueprints() : array
# getInventionProbability() : float
# getInventableBlueprintIdsByRaceId(race: int) : array
# getInventionOutputRuns() : int
+ getDecryptorGroupId() : ing
+ getDecryptorIds() : array
+ calcInventionProbability(charMod: ICharacterModifier) : float

T3Blueprint

inventedFrom: array

loadOriginatingBlueprints(sde: SDE) : void
getInventionRelicIds() : array
getInventionRelics() : array
+ inventFromRelic(iMod: IndustryModifier, relicId: i...
+ inventManufacture(iMod: IndustryModifier, relicId...

Blocked methods omitted

Relic

queryAttributes() : array
+ invent(iMod: IndustryModifier, inventedBpId: int, decryptorId...
+ inventManufacture(iMod: IndustryModifier, inventedBpId: int,...

Process classes

ReactionProcessData

inputMaterialMap: MaterialMap
outputMaterialMap: MaterialMap
cycles: float
withRefining: bool
withFeedback: bool

+ __construct(inputMaterialMap: MaterialMap, outputMaterialMap:...
+ geInputMaterialMap() : MaterialMap
+ getOutputMaterialMap() : MaterialMap
+ getCycles() : float
+ getTime() : float
+ withRefining() : bool
+ withFeedback() : bool
+ getInputBuyCost(iMod: IndustryModifier) : float
+ getOutputSellValue(iMod: IndustryModifier) : float
+ getProfit(buyContext: IndustryModifier, sellContext: IndustryMo...

ProcessData

activity: int
producesTypeId: int
producesQuantity: int
processTime: int
processCost: float
assemblyLineId: int
solarSystemId: int
skills: SkillMap
materials: MaterialMap
subProcessData: array

+ __construct(producesTypeId: int, producesQuantity: int, pr...
+ 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
+ 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(buyContext: IndustryModifier, sellContext: In...
+ printData(iMod: IndustryModifier) : void

0..*

ManufactureProcessData

bpMeLevel: int
bpPeLevel: int

+ __construct(producesTypeId: int, producesQua...
+ getMeLevel() : int
+ getPeLevel() : int
+ getTotalCostPerUnit(iMod: IndustryModifier) : fl...
+ getTotalProfit(buyContext: IndustryModifier, sell...
+ printData(buyContext: IndustryModifier, sellCon...

InventionProcessData

probability: float
resultRuns: int
resultME: int
resultTE: int

+ __construct(producesTypeId: int, inventionTime: int, proces...
+ getResultRuns() : int
+ getResultME() : int
+ getResultPE() : int
+ getProbability() : float
+ getSuccessTime() : float
+ getTotalSuccessTime() : float
+ getTotalSuccessTimes() : array
+ getSuccessMaterialMap() : MaterialMap
+ getTotalSuccessMaterialMap() : MaterialMap
+ getSuccessMaterialVolume() : float
+ getTotalSuccessMaterialVolume() : float
+ getProcessCost() : float
+ getSuccessProcessCost() : float
+ getTotalSuccessProcessCost() : float
+ getSuccessMaterialBuyCost(iMod: IndustryModifier) : float
+ getTotalSuccessMaterialBuyCost(iMod: IndustryModifier) : fl...
+ getTotalSuccessCost(iMod: IndustryModifier) : float
+ printData(buyContext: IndustryModifier, sellContext: Indust...

ResearchMEProcessData

startMELevel: int
endMELevel: int

+ __construct(researchedBpId: int, researchTime: i...
+ getStartMELevel() : int
+ getEndMELevel() : int

ResearchTEProcessData

startTELevel: int
endTELevel: int

+ __construct(researchedBpId: int, researchTime: i...
+ getStartTELevel() : int
+ getEndTELevel() : int

CopyProcessData

outputRuns: int

+ __construct(bpCopyId: int, copyQuantity: int, o...
+ getOutputRuns() : int

«interface»
ICharacterModifier

+ getSkillLevel(skillId: int) : int
+ getIndustryImplantTimeFactor(activityId: int) : float
+ getBuyTaxFactor(factionId: int, corpId: int) : float
+ getSellTaxFactor(factionId: int, corpId: int) : float
+ getIndustrySkillTimeFactor(activityId: int) : float
+ getBrokerTax(factionId: int, corpId: int) : float
+ getFactionStanding(factionId: int) : float
+ getCorporationStanding(corpId: int) : float
+ getReprocessingTaxFactor(corpId: int) : float
+ getReprocessingImplantYieldFactor() : float

«interface»
IBlueprintModifier

+ getBpMeLevel(bpId: int) : int
+ getBpTeLevel(bpId: int) : int

CharacterModifier

pirateFactions: array
pirateCorporations: array

+ getStandingWithSkillBonus(baseStanding: float, is...

BlueprintModifier

IndustryModifier

assemblyLines: array
solarSystem: SolarSystem
tax: float
characterModifier: ICharacterModifier
blueprintModifier: IBlueprintModifier
preferredMarketStationId: int
marketStations: array
maxPriceDataAge: int

+ getByStation(stationId: int, assemblyLineTypeIds: array, tax: float) : I...
+ getBySystemIdForPos(solarSystemId: int, tax: float) : IndustryModifier
+ getBySystemIdForAllNpcStations(solarSystemId: int) : IndustryModifier
+ getBySystemIdForInstallationType(solarSystemId: int, installationTypeI...
+ getBySystemIdWithAssembly(solarSystemId: int, assemblyLineTypeIds...
+ __construct(system: SolarSystem, assemblyLines: array, tax: float) : In...
+ getCharacterModifier() : ICharacterModifier
+ setCharacterModifier(charMod: ICharacterModifier) : void
+ getBlueprintModifier() : IBlueprintModifier
+ setBlueprintModifier(bpMod: IBlueprintModifier) : void
+ setPreferredMarketStation(stationId: int) : void
# getMaxPriceDataAge() : int
# setMaxPriceDateAge(maxPriceDataAge: int) : void
+ getAssemblyLines() : array
+ getAssemblyLinesForActivity(activityId: int) : array
+ getSolarSystem() : SolarSystem
+ getTax() : float
+ getTaxFactor() : float
+ isActivityPossible(activityId: int, type: Type) : bool
+ getModifier(activityId: int, type: Type) : array
+ getBestAssemblyLineForActivity(activityId: int, type: Type) : AssemblyL...
+ getBestMarketStation() : Station
+ getBestReprocessingStation() : Station
+ getSellTaxFactor() : float
+ getSellBuyFactor() : float

AssemblyLine

+ classNick: string
# instancePool: InstancePool
# baseTimeMultiplier: float
# baseMaterialMultiplier: float
# baseCostMultiplier: float
# activityId: int
# groupModifiers: array
# categoryModifiers: array

+ getAssemblyLine(assemblyLineTypeId: int) : AssemblyLine
+ getBestPosAssemblyLineTypeIds(systemSecurity: float) : array
+ getHisecStationAssemblyLineTypeIds() : array
+ getIdByName() : void
+ getClassHierarchyKeyPrefix() : string
# __construct(id: int) : AssemblyLine
+ getBaseTimeMultiplier() : float
+ getBaseMaterialMultiplier() : float
+ getBaseCostMultiplier() : float
+ getActivityId() : int
+ getGroupModifiers() : array
+ getCategoryModifiers() : array
+ getModifiersForType(type: Type) : array
+ isTypeCompatible(type: Type) : bool

Station

+ classNick: string
# instancePool: InstancePool
# operations: array
# operationServices: array
# services: array
# solarSystemId: int
# operationId: int
# stationTypeId: int
# corporationId: int
# factionId: int
# conquerable: bool
# reprocessingEfficiency: float
# reprocessingStationsTake: float
# tax: float
# assemblyLineTypeIds: array

loadNames() : void
+ getClassHierarchyKeyPrefix() : string
# getStaticOperationDetails(operationId: int) : array
# getStaticOperationServices(operationId: int) : array
+ getServiceName(serviceId: int) : string
# __construct(id: int) : Station
+ getSolarSystemId() : int
+ getSolarSystem() : SolarSystem
+ getOperationId() : int
+ getOperationDetails() : array
+ getServiceIds() : array
+ getStationTypeId() : int
+ getCorporationId() : int
+ getFactionId() : int
+ isConquerable() : bool
+ getReprocessingEfficiency() : float
+ getTax() : float
+ setTax(tax: float) : void
+ getAssemblyLineTypeIds() : array
+ getAssemblyLineTypeIdsForActivity(activityId: int) : array
+ getIndustryModifier() : IndustryModifier

SolarSystem

+ classNick: string
# instancePool: InstancePool
# regionId: int
# constellationId: int
# security: float
# factionId: int
# industryIndexDate: int
# industryIndices: array
# stationIds: array
# stations: array

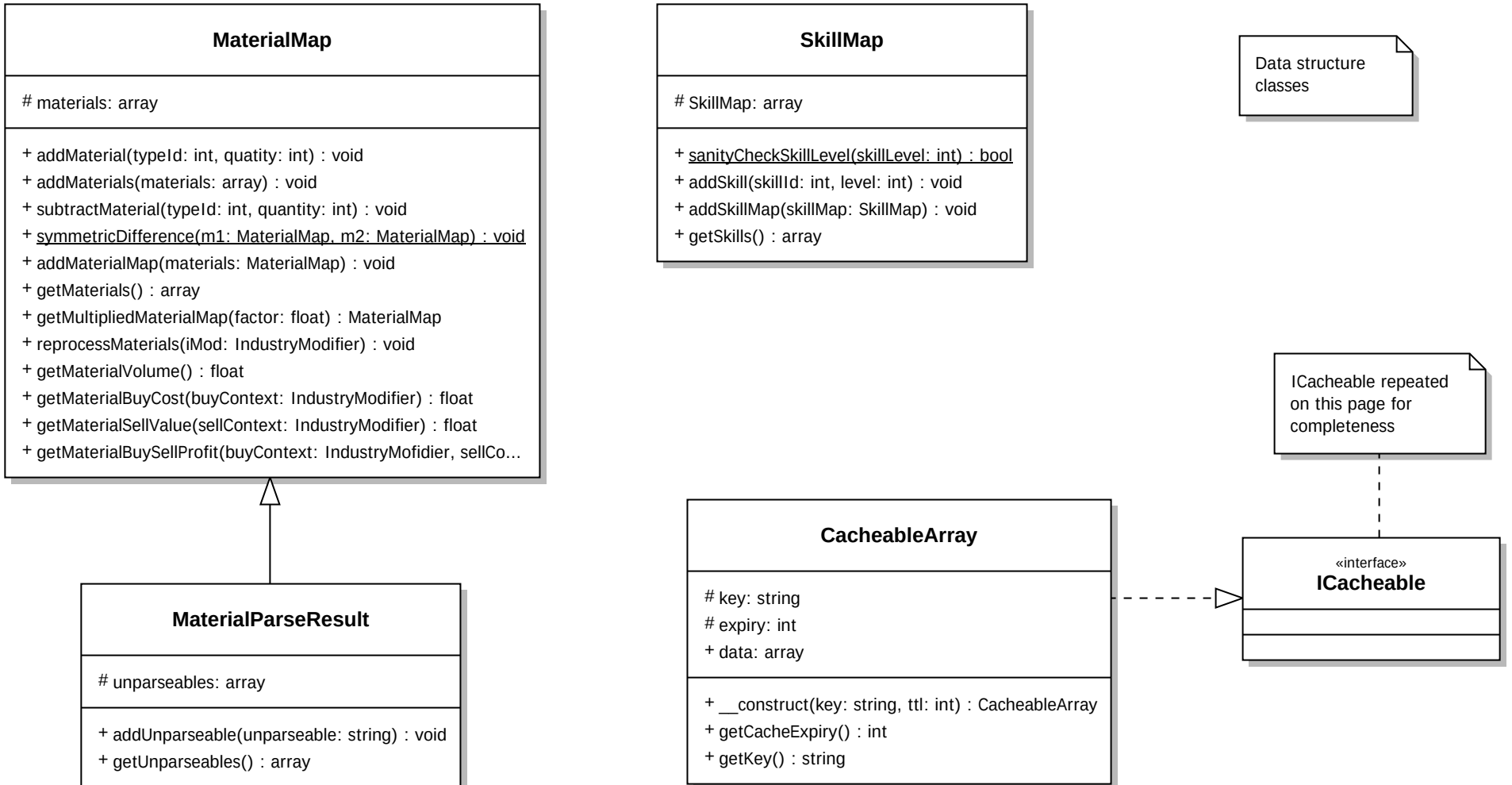
loadNames() : void
+ getClassHierarchyKeyPrefix() : string
# __construct(id: int) : SolarSystem
# loadStations(sde: SDE) : void
+ getSolarSystemId() : int
+ getSolarSystemName() : string
+ getRegionId() : int
+ getConstellationId() : int
+ getSecurity() : float
+ getStationIds() : array
+ getStations() : array
+ getIndustryIndexDate() : int
+ getIndustryIndices(maxIndexDataAge: int) : array
+ getIndustryIndexForActivity(activityId: int, maxIndexDataAge: int) : float
+ setIndustryIndices(indices: array) : void
+ getIndustryModifierForPos(tax: float) : IndustryModifier
+ getIndustryModifierForAllNpcStations() : IndustryModifier

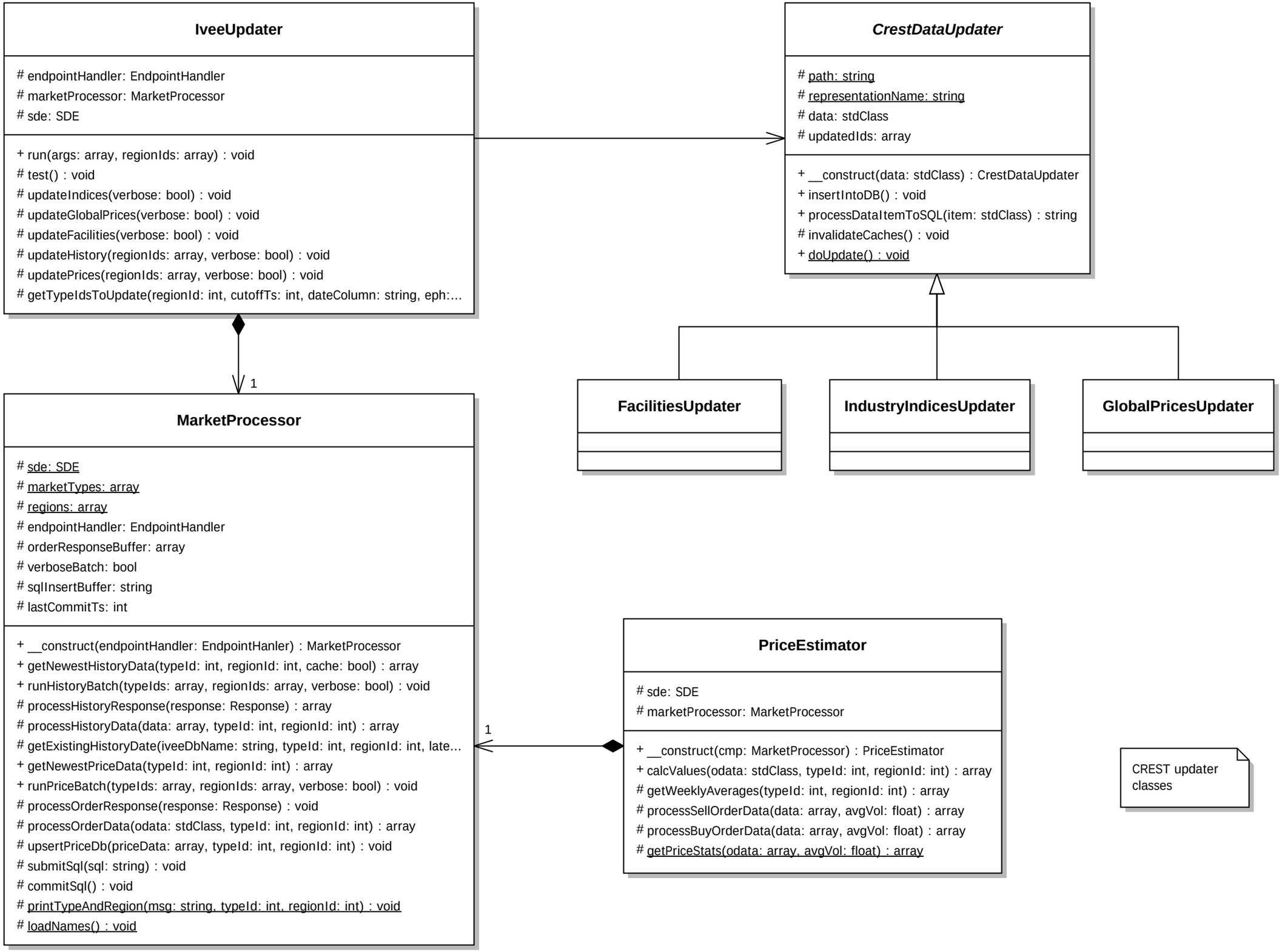
SdeType

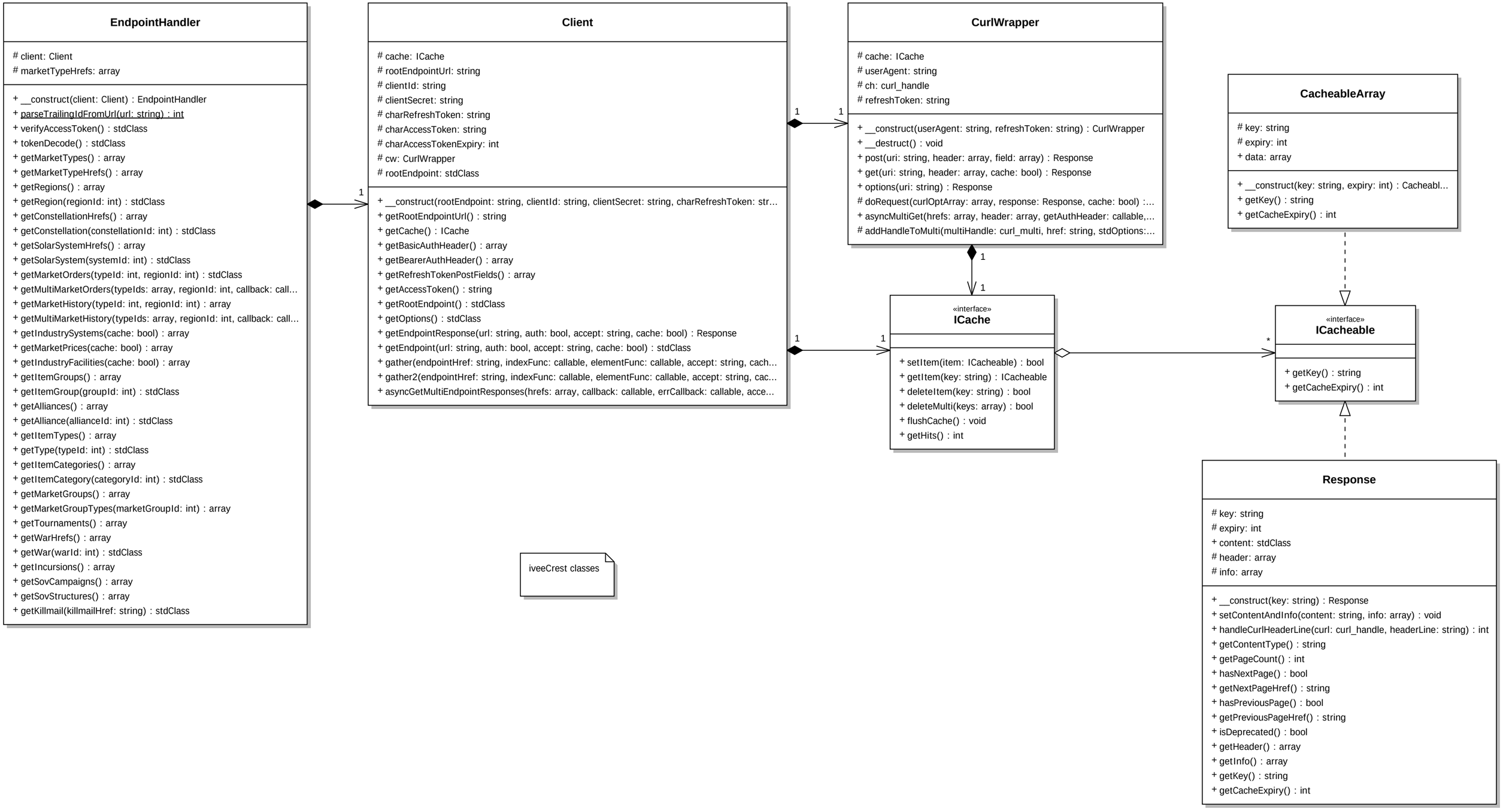
SdeType repeated on this page for completeness

Industry context classes

1
1
1
1
1..*
0..*
1
0..*


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(buyContext: IndustryModifier) : float
+ getMaterialSellValue(sellContext: IndustryModifier) : float
+ getMaterialBuySellProfit(buyContext: IndustryMofidier, sellCo...
SkillMap
SkillMap: array
+ sanityCheckSkillLevel(skillLevel: int) : bool
+ addSkill(skillId: int, level: int) : void
+ addSkillMap(skillMap: SkillMap) : void
+ getSkills() : array
Data structure classes
ICacheable repeated on this page for completeness
MaterialParseResult
unparseables: array
+ addUnparseable(unparseable: string) : void
+ getUnparseables() : array
CacheableArray
key: string
expiry: int
+ data: array
+ __construct(key: string, ttl: int) : CacheableArray
+ getCacheExpiry() : int
+ getKey() : string
«interface»
ICacheable


IveeUpdater
endpointHandler: EndpointHandler
marketProcessor: MarketProcessor
sde: SDE
+ run(args: array, regionIds: array) : void
# test() : void
# updateIndices(verbose: bool) : void
# updateGlobalPrices(verbose: bool) : void
# updateFacilities(verbose: bool) : void
# updateHistory(regionIds: array, verbose: bool) : void
# updatePrices(regionIds: array, verbose: bool) : void
# getTypeIdsToUpdate(regionId: int, cutoffTs: int, dateColumn: string, eph:...
CrestDataUpdater
path: string
representationName: string
data: stdClass
updatedIds: array
+ __construct(data: stdClass) : CrestDataUpdater
+ insertIntoDB() : void
+ processDataItemToSQL(item: stdClass) : string
# invalidateCaches() : void
+ doUpdate() : void
1
FacilitiesUpdater
IndustryIndicesUpdater
GlobalPricesUpdater
MarketProcessor
sde: SDE
marketTypes: array
regions: array
endpointHandler: EndpointHandler
orderResponseBuffer: array
verboseBatch: bool
sqlInsertBuffer: string
lastCommitTs: int
+ __construct(endpointHandler: EndpointHanler) : MarketProcessor
+ getNewestHistoryData(typeId: int, regionId: int, cache: bool) : array
+ runHistoryBatch(typeIds: array, regionIds: array, verbose: bool) : void
# processHistoryResponse(response: Response) : array
# processHistoryData(data: array, typeId: int, regionId: int) : array
# getExistingHistoryDate(iveeDbName: string, typeId: int, regionId: int, late...
+ getNewestPriceData(typeId: int, regionId: int) : array
+ runPriceBatch(typeIds: array, regionIds: array, verbose: bool) : void
# processOrderResponse(response: Response) : void
# processOrderData(odata: stdClass, typeId: int, regionId: int) : array
# upsertPriceDb(priceData: array, typeId: int, regionId: int) : void
# submitSql(sql: string) : void
# commitSql() : void
# printTypeAndRegion(msg: string, typeId: int, regionId: int) : void
# loadNames() : void
1
PriceEstimator
sde: SDE
marketProcessor: MarketProcessor
+ __construct(cmp: MarketProcessor) : PriceEstimator
+ calcValues(odata: stdClass, typeId: int, regionId: int) : array
# getWeeklyAverages(typeId: int, regionId: int) : array
# processSellOrderData(data: array, avgVol: float) : array
# processBuyOrderData(data: array, avgVol: float) : array
# getPriceStats(odata: array, avgVol: float) : array
CREST updater classes


EndpointHandler
client: Client
marketTypeHrefs: array
+ __construct(client: Client) : EndpointHandler
+ parseTrailingIdFromUrl(url: string) : int
+ verifyAccessToken() : stdClass
+ tokenDecode() : stdClass
+ getMarketTypes() : array
+ getMarketTypeHrefs() : array
+ getRegions() : array
+ getRegion(regionId: int) : stdClass
+ getConstellationHrefs() : array
+ getConstellation(constellationId: int) : stdClass
+ getSolarSystemHrefs() : array
+ getSolarSystem(systemId: int) : stdClass
+ getMarketOrders(typeId: int, regionId: int) : stdClass
+ getMultiMarketOrders(typeIds: array, regionId: int, callback: call...
+ getMarketHistory(typeId: int, regionId: int) : array
+ getMultiMarketHistory(typeIds: array, regionId: int, callback: call...
+ getIndustrySystems(cache: bool) : array
+ getMarketPrices(cache: bool) : array
+ getIndustryFacilities(cache: bool) : array
+ getItemGroups() : array
+ getItemGroup(groupId: int) : stdClass
+ getAlliances() : array
+ getAlliance(allianceId: int) : stdClass
+ getItemTypes() : array
+ getType(typeId: int) : stdClass
+ getItemCategories() : array
+ getItemCategory(categoryId: int) : stdClass
+ getMarketGroups() : array
+ getMarketGroupTypes(marketGroupId: int) : array
+ getTournaments() : array
+ getWarHrefs() : array
+ getWar(warId: int) : stdClass
+ getIncursions() : array
+ getSovCampaigns() : array
+ getSovStructures() : array
+ getKillmail(killmailHref: string) : stdClass
1
Client
cache: ICache
rootEndpointUrl: string
clientId: string
clientSecret: string
charRefreshToken: string
charAccessToken: string
charAccessTokenExpiry: int
cw: CurlWrapper
rootEndpoint: stdClass
+ __construct(rootEndpoint: string, clientId: string, clientSecret: string, charRefreshToken: str...
+ getRootEndpointUrl() : string
+ getCache() : ICache
+ getBasicAuthHeader() : array
+ getBearerAuthHeader() : array
+ getRefreshTokenPostFields() : array
+ getAccessToken() : string
+ getRootEndpoint() : stdClass
+ getOptions() : stdClass
+ getEndpointResponse(url: string, auth: bool, accept: string, cache: bool) : Response
+ getEndpoint(url: string, auth: bool, accept: string, cache: bool) : stdClass
+ gather(endpointHref: string, indexFunc: callable, elementFunc: callable, accept: string, cach...
+ gather2(endpointHref: string, indexFunc: callable, elementFunc: callable, accept: string, cac...
+ asyncGetMultiEndpointResponses(hrefs: array, callback: callable, errCallback: callable, acce...
1
1
CurlWrapper
cache: ICache
userAgent: string
ch: curl_handle
refreshToken: string
+ __construct(userAgent: string, refreshToken: string) : CurlWrapper
+ __destruct() : void
+ post(uri: string, header: array, field: array) : Response
+ get(uri: string, header: array, cache: bool) : Response
+ options(uri: string) : Response
# doRequest(curlOptArray: array, response: Response, cache: bool) :...
+ asyncMultiGet(hrefs: array, header: array, getAuthHeader: callable,...
# addHandleToMulti(multiHandle: curl_multi, href: string, stdOptions:...
1
1
1
1
«interface»
ICache
+ setItem(item: ICacheable) : bool
+ getItem(key: string) : ICacheable
+ deleteItem(key: string) : bool
+ deleteMulti(keys: array) : bool
+ flushCache() : void
+ getHits() : int
*
CacheableArray
key: string
expiry: int
+ data: array
+ __construct(key: string, expiry: int) : Cacheabl...
+ getKey() : string
+ getCacheExpiry() : int
«interface»
ICacheable
+ getKey() : string
+ getCacheExpiry() : int
Response
key: string
expiry: int
+ content: stdClass
# header: array
# info: array
+ __construct(key: string) : Response
+ setContentAndInfo(content: string, info: array) : void
+ handleCurlHeaderLine(curl: curl_handle, headerLine: string) : int
+ getContentType() : string
+ getPageCount() : int
+ hasNextPage() : bool
+ getNextPageHref() : string
+ hasPreviousPage() : bool
+ getPreviousPageHref() : string
+ isDeprecated() : bool
+ getHeader() : array
+ getInfo() : array
+ getKey() : string
+ getCacheExpiry() : int
iveeCrest classes