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

Config

sdeDbHost: string
sdeDbPort: int
sdeDbUser: string
sdeDbPw: string
sdeDbName: string
iveeDbName: string
useCache: bool
cacheHost: string
cachePort: int
cachePrefix: string
emdrRelayUrl: string
crestBaseUrl: string
userAgent: string
classes: array

+ getIveeClassName(classNickname: string) : string

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

trivial getters
omitted.
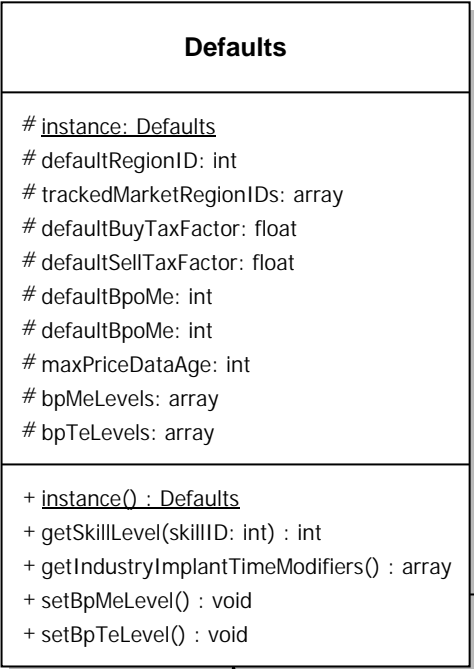
trivial getters
ommited.

Exception classes
omitted.

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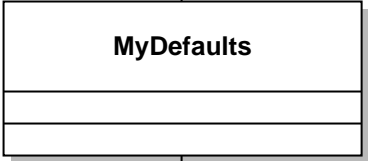
For user implementation.
Also serves as example
for the intended way of
extending iveeCore

Util

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

FitParser

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

«interface» ICacheable

+ deleteFromCache(keys: array) : void
+ getKey() : string
+ getCacheTTL() : int

InstancePool

className: string
cache: ICache
keyToObj: array
nameToKey: array
poolHits: int

+ __construct(className: string) : InstancePool
+ setObj(obj: ICacheable) : void
+ setNamesToKeys(namesToKey: array) : void
+ getObjByKey(key: string) : object
+ getKeyByName(name: string) : int
+ deleteFromCache(keys: array) : void
+ getPoolHits() : int
+ getPooledTypeCount() : int

MemcachedWrapper

instance: MemcachedWrapper
memcached: Memcached
memcachedHit: int

__construct() : MemcachedWrapper
instance() : MemcachedWrapper
+ setItem(item: mixed, key: string, ...
+ getItem(key: string) : mixed
+ deleteItem(key: string) : bool
+ deleteMulti(keys: array) : bool
+ flushCache() : bool

«interface» ICache

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

CacheableCommon

id: int

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

SdeType

name: string

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

RegionMarketData

instancePool: InstancePool
classNick: string
regionID: int
priceDate: int
sellPrice: float
buyPrice: float
supplyIn5: int
demandIn5: int
avgSell5OrderAge: int
avgBuy5OrderAge: int
histDate: int
avgVol: float
avgTx: float
low: float
high: float
avg: float

getByIdAndRegion(typeID: int, regionID: int) : RegionMarket...
__construct(typeID: int, regionID: int) : RegionMarketData
+ getRegionID() : int
+ getKey() : string
+ getType() : Type
+ getCacheTTL() : int
+ getBuyPrice(maxPriceDataAge: int) : float
+ getSellPrice(maxPriceDataAge: int) : float
+ getHistory(regionID: int, fromDateTS: int, toDateTS: int) : array
+ getPriceDate() : int
+ getAvgVol() : float
+ getAvgTx() : float
+ getSupplyIn5() : int
+ getDemandIn5() : int
+ getAvgBuy5OrderAge() : int
+ getAvgSell5OrderAge() : int
+ getHistDate() : int
+ getLow() : float
+ getHigh() : float
+ getAvg() : float
# throwNotOnMarketException() : void

GlobalPriceData

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

getById(id: int) : GlobalPriceData
__construct(typeID: int) : GlobalPriceData
queryAttributes() : array
+ setAttributes(row: array) : void
+ getPriceDate() : int
+ getAveragePrice(maxPriceDataAge: int) : float
+ getAdjustedPrice(maxPriceDataAge: int) : float

Type

instancePool: InstancePool
classNick: string
groupID: int
categoryID: int
volume: float
portionSize: int
basePrice: int
materials: array
reprocessingSkillID: int

+ getById(id: int) : Type
+ loadNames() : void
- 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
+ getGlobalPriceData() : GlobalPriceData
+ getRegionMarketData(regionID: int) : RegionMarketData
+ isReprocessable() : bool
+ getMaterials() : array
+ getReprocessingSkillID() : int
+ getReprocessingMaterialMap(batchSize: int, equipmentYield: f...
+ calcReprocessingTaxFactor(standings: float) : float

ReactionProduct

productOfReactionIDs: array

__construct(typeID: int) : ReactionProduct
+ getReactions() : array
+ getReactionIDs() : array

Blueprint

productId: int
maxProductionLimit: int
rank: int
activityMaterials: array
activitySkills: array
activityTimes: array
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

Reaction

cycleInputMaterial: array
cycleOutputMaterial: array
isAlchemy: bool

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

ProcessData

activity: int
producesTypeID: int
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

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

0..*

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

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
+ 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

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

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

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
- + getBySystemIdWithAssembly(solarSystemID: int, assemblyLineTypeID...
- + __construct(system: SolarSystem, assemblyLines: array, teams: array, ...
- + getAssemblyLines() : array
- + getAssemblyLinesForActivity(activityID: int) : array
- + getSolarSystem() : SolarSystem
- + getTax() : float
- + 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
- # baseMaterialMultiplier: float
- # baseCostMultiplier: float
- # activityID: int
- # 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
- + getCategoryModifiers() : array
- + getModifiersForType(type: Type) : array
- + isTypeCompatible(type: Type) : bool

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

0..* 1

Speciality

- # instancePool: InstancePool
- # classNick: string
- # specialityGroupIDs: array

- # getIdByName() : void
- # __construct(id: int) : Speciality
- + getSpecialityGroupIDs() : array
- + appliesToGroupID(groupID: int) : bool

5

SolarSystem

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

- # loadNames() : void
- # __construct(id: int) : SolarSystem
- + getSolarSystemID() : int
- + getSolarSystemName() : string
- + getRegionID() : int
- + getConstellationID() : int
- + getSecurity() : float
- + getStationIDs() : array
- + getStations() : array
- + getTeamIDs() : array
- + getTeams() : 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

0..* 1..* 1 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(equipmentYield: float, reprocessingTaxFa…
+ getMaterialVolume() : float
+ getMaterialBuyCost(maxPriceDataAge: int, regionId: int) : float
+ getMaterialSellValue(maxPriceDataAge: int, regionId: int) : fl…

SkillMap

SkillMap: array

+ sanityCheckSkillLevel(skillLevel: int) : bool
+ addSkill(skillID: int, level: int) : void
+ addSkillMap(skillMap: SkillMap) : void
+ getSkills() : array

MaterialParseResult

unparseables: array

+ addUnparseable(unparseable: string) : void
+ getUnparseables() : 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

EmdrPriceUpdate

typeID: int
regionID: int
generatedAt: int
averages: array
sell: float
avgSell5OrderAge: int
buy: float
avgBuy5OrderAge: int
demandIn5: 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

+ __construct(data: stdClass) : CrestDataUpdater
+ insertIntoDB() : void
+ processDataItemToSQL(item: stdClass) : string
# invalidateCaches() : void
+ doUpdate() : void

CrestFetcher

baseUrl: string
userAgent: string

+ __construct() : Fetcher
+ getCrestData(path: string, representationName: string) : stdClass
# curlGetJson(url: string) : stdClass
# parseContentTypeToRepresentation(contentType: string) : string

TeamsUpdater

IndustryFacilitiesUpdater

IndustrySystemsUpdater

MarketPricesUpdater

SpecialitiesUpdater