



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(): float + getTotalMaterialBuyCost(): float + getTotalCost(maxPriceDataAge: int) : float + getMaterialMap() : MaterialMap + getTotalMaterialMap() : MaterialMap + getMaterialVolume() : float + getTotalMaterialVolume() : float + getSkillMap() : SkillMap + getTotalSkillMap() : SkillMap + getTime() : int + getTotalTime(): int + getTotalTimes() : array + getTotalProfit(maxPriceDataAge: int) : float + printData() : void \ _{0..}*

ResearchMEProcessData

ReactionProcessData

+ __construct(inputMaterialMap: MaterialMap, ou...

+ getInputBuyCost(maxPriceDataAge: int) : float

+ getOutputBuyValue(maxPriceDataAge: int) : float

inputMaterialMap: MaterialMap

cycles: float # withRefining: bool

withFeedback: bool

+ getCycles() : float

+ withRefining(): bool

+ withFeedback(): bool

+ getTime() : float

outputMaterialMap: MaterialMap

+ geInputMaterialMap() : MaterialMap

+ getOutputMaterialMap() : MaterialMap

+ getProfit(maxPriceDataAge: int) : float

 $\#\, start MELevel \colon int$

 $\#\, end MELevel \colon int$

 $+ \underline{\hspace{0.3cm}} construct (researched BpID: int, \ research Time: i...$

+ getStartMELevel() : int

+ getEndMELevel() : int

ResearchTEProcessData

startTELevel: int

endTELevel: int

+ __construct(researchedBpID: int, researchTime: i...

+ getStartTELevel() : int + getEndTELevel() : int

CopyProcessData

ManufactureProcessData

+ __construct(producesTypeID: int, producesQua...

+ getTotalCostPerUnit(maxPriceDataAge: int) : float

+ getTotalProfit(maxPriceDataAge: int) : float

 $\#\,outputRuns\colon int$

bpMeLevel: int

bpPeLevel: int

+ getMeLevel() : int + getPeLevel() : int

+ getSlotCost() : float

+ printData() : void

+ __construct(bpCopyID: int, copyQuantity: int, o...

+ getOutputRuns(): int

InventionProcessData

probability: float
resultRuns: int
resultME: 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

get i otal successiviate laive

+ getProcessCost() : float

+ getSuccessProcessCost() : float + getTotalSuccessProcessCost() : float

+ getSuccessMaterialBuyCost(maxPriceDataAge: int) : float

+ getTotalSuccessMaterialBuyCost(maxPriceDataAge: int) : float

+ getTotalSuccessCost(maxPriceDataAge: int) : float

+ printData(): void

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

Team # instancePool: InstancePool # classNick: string # solarSystemID: int # creationTime: int 0..* # expiryTime: int # activityID: int # costModifier: float # specialityID: int # bonusIDs: array # bonusValues: array # workerSpecialities: array # getIdByName(): void # __construct(id: int) : Team + getSolarSystemID(): int

Speciality

- # instancePool: InstancePool
- # classNick: string

+ 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

- # specialityGroupIDs: 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
- + 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

1..*

0..*

- #__construct(id: int) : AssemblyLine
- + getBaseTimeMultiplier() : float
- + getBaseMaterialMultiplier() : float
- + getBaseCostMultiplier() : float
- + getActivityID(): int
- + getGroupModifiers() : array
- + getCategoryModifiers() : array
- + getModifiersForType(type: Type) : array
- + isTypeCompatible(type: Type) : bool

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

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

+ reprocess Materials (equipment Yield: float, reprocessing Tax Fa...

+ getMaterialVolume() : float

+ getMaterialBuyCost(maxPriceDataAge: int): float

+ getMaterialSellValue(maxPriceDataAge: int) : float

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