

Mode: All

Left file: /Users/zachtindall/Downloads/WeightPoint-nodenext.js

Right file: /Users/zachtindall/Downloads/WeightPoint-preserve.js

"use strict";	+ -	
<pre>var __decorate = (this && this.__decorate) ? function (decorators, target, key, desc) { ? var c = arguments.length, r = c < 3 ? targ ? et : desc === null ? desc = Object.getOwnPro ? pertyDescriptor(target, key) : desc, d; ? if (typeof Reflect === "object" && typeof ? Reflect.decorate === "function") r = Reflec ? t.decorate(decorators, target, key, desc); ? else for (var i = decorators.length - 1; i ? >= 0; i--) if (d = decorators[i]) r = (c < 3 ? ? d(r) : c > 3 ? d(target, key, r) : d(targe ? t, key)) r; ? return c > 3 && r && Object.defineProperty ? (target, key, r), r; ? }; var __metadata = (this && this.__metadata) ? function (k, v) { ? if (typeof Reflect === "object" && typeof ? Reflect.metadata === "function") return Re ? flect.metadata(k, v); ? }; var WeightPoint_1;</pre>	=	<pre>var __decorate = (this && this.__decorate) ? function (decorators, target, key, desc) { ? var c = arguments.length, r = c < 3 ? targ ? et : desc === null ? desc = Object.getOwnPro ? pertyDescriptor(target, key) : desc, d; ? if (typeof Reflect === "object" && typeof ? Reflect.decorate === "function") r = Reflect ? .decorate(decorators, target, key, desc); ? else for (var i = decorators.length - 1; i ? >= 0; i--) if (d = decorators[i]) r = (c < 3 ? ? d(r) : c > 3 ? d(target, key, r) : d(targe ? t, key)) r; ? return c > 3 && r && Object.defineProperty ? (target, key, r), r; ? }; var __metadata = (this && this.__metadata) ? function (k, v) { ? if (typeof Reflect === "object" && typeof ? Reflect.metadata === "function") return Refl ? ect.metadata(k, v); ? }; var WeightPoint_1;</pre>
<pre>Object.defineProperty(exports, "__esModule", ? { value: true }); exports.WeightPoint = void 0; const typegoose_1 = require("@typegoose/typego ? ose"); const logging_1 = require("@wastewizer/loggin ? g"); const weight_algorithm_1 = require("@wastewiz ? er/weight-algorithm"); const safe_units_1 = require("safe-units"); const BaseModel_1 = require("../Base/BaseMode ? l"); const ContainerSite_1 = require("../Container ? Site/ContainerSite"); const ForcePoint_1 = require("../ForcePoint/F ? orcePoint"); const WeightAlgo_1 = require("../WeightAlgo"); let WeightPoint = WeightPoint_1 = class Weight ? Point extends BaseModel_1.BaseModel {</pre>	<>	<pre>import { prop, index, modelOptions, } from '@t ? ypegoose/typegoose'; import { logger } from '@wastewizer/logging'; import { CorgiAlgorithm, WeightAdjustmentFacto ? r, } from '@wastewizer/weight-algorithm'; import { kilograms } from 'safe-units'; import { BaseModel } from '../Base/BaseModel'; import { ContainerSite } from '../ContainerSit ? e/ContainerSite'; import { ForcePoint } from '../ForcePoint/Forc ? ePoint'; import { WeightAlgoBase, WeightAlgoDiscrimina ? tors } from '../WeightAlgo'; let WeightPoint = WeightPoint_1 = class Weight ? Point extends BaseModel {</pre>
<pre> constructor(args) { super(); this.algorithm = args.algorithm; this.sourceContainerSite = args.sourc ? eContainerSite; this.sourceForcePoint = args.sourceFo ? rcePoint; this.timestamp = args.timestamp;</pre>	=	<pre> constructor(args) { super(); this.algorithm = args.algorithm; this.sourceContainerSite = args.sourc ? eContainerSite; this.sourceForcePoint = args.sourceFo ? rcePoint; this.timestamp = args.timestamp;</pre>

Left file: /Users/zachtindall/Downloads/WeightPoint-nodenext.js
Right file: /Users/zachtindall/Downloads/WeightPoint-preserve.js
(continued)

```

    this.weightKg = args.weightKg;
  }
  // This is the time the point was recorded, not when this record was created in the
  ? DB (createdAt)
  ? timestamp;
  ? weightKg;
  ? sourceContainerSite;
  ? sourceForcePoint;
  ? algorithm;
  ? static CreateCalculate({ sourceForcePoint
  ? , sourceContainerSite, }) {
    if (sourceForcePoint.forceNewtons ===
  ? undefined) {
      throw new Error('Cannot create we
  ? ight point without a force');
    }
    const binBarAlgorithm = sourceContain
  ? erSite.binBarAlgorithms.find((bba => bba.b
  ? inBar._id.equals(sourceForcePoint.sourceBin
  ? Bar._id));
    if (!binBarAlgorithm) {

```

```

    this.weightKg = args.weightKg;
  }
  // This is the time the point was recorded
  ? d, not when this record was created in the
  ? DB (createdAt)
  ? timestamp;
  ? weightKg;
  ? sourceContainerSite;
  ? sourceForcePoint;
  ? algorithm;
  ? static CreateCalculate({ sourceForcePoint
  ? , sourceContainerSite, }) {
    if (sourceForcePoint.forceNewtons ===
  ? undefined) {
      throw new Error('Cannot create we
  ? ight point without a force');
    }
    const binBarAlgorithm = sourceContain
  ? erSite.binBarAlgorithms.find((bba => bba.bi
  ? nBar._id.equals(sourceForcePoint.sourceBinB
  ? ar._id));
    if (!binBarAlgorithm) {

```

logging_1.logger.error({ <> logger.error({

```

    sourceForcePoint,
    sourceContainerSite,
  }, 'Cannot find BinBar algorithm
  ? for container site');
  throw new Error('Cannot find BinB
  ? ar algorithm for container site');
}

```

```

    sourceForcePoint,
    sourceContainerSite,
  }, 'Cannot find BinBar algorithm
  ? for container site');
  throw new Error('Cannot find BinB
  ? ar algorithm for container site');
}

```

const algorithm = new weight_algorithm m_1.CorgiAlgorithm(); // TODO: add cycle in stall date for CAF <> const algorithm = new CorgiAlgorithm(); // TODO: add cycle install date for CAF

```

const adjustmentFactors = [];
if (binBarAlgorithm.algorithm.isCorgi
  ? Algo()) {

```

```

const adjustmentFactors = [];
if (binBarAlgorithm.algorithm.isCorgi
  ? Algo()) {

```

```

    adjustmentFactors.push(new weight
  ? _algorithm_1.WeightAdjustmentFactor(binBarA
  ? lgorithm.algorithm.weightAdjustmentFactor))
  ? ;

```

```

    adjustmentFactors.push(new Weight
  ? AdjustmentFactor(binBarAlgorithm.algorithm.
  ? weightAdjustmentFactor));

```

```

  }
  const weight = algorithm.calculateAdj
  ? ustedWeight(sourceForcePoint.forceNewtons,
  ? sourceForcePoint.timestamp, adjustmentFacto
  ? rs);
  return new WeightPoint_1({
    algorithm: binBarAlgorithm.algori
  ? thm,
    sourceContainerSite: sourceContai
  ? nerSite,
    sourceForcePoint: sourceForcePoin

```

```

  }
  const weight = algorithm.calculateAdj
  ? ustedWeight(sourceForcePoint.forceNewtons,
  ? sourceForcePoint.timestamp, adjustmentFacto
  ? rs);
  return new WeightPoint_1({
    algorithm: binBarAlgorithm.algori
  ? thm,
    sourceContainerSite: sourceContai
  ? nerSite,
    sourceForcePoint: sourceForcePoin

```

Left file: /Users/zachtindall/Downloads/WeightPoint-nodenext.js
Right file: /Users/zachtindall/Downloads/WeightPoint-preserve.js
(continued)

<pre>? t, timestamp: sourceForcePoint.times ? tamp,</pre>		<pre>? t, timestamp: sourceForcePoint.times ? tamp,</pre>
<pre> weightKg: weight.over(safe_units_ ? 1.kilograms).value,</pre>	<>	<pre> weightKg: weight.over(kilograms). ? value,</pre>
<pre> }); } };</pre>	=	<pre> }); } };</pre>
<pre>exports.WeightPoint = WeightPoint;</pre>	+ -	
<pre>__decorate([(0, typegoose_1.prop)({ required: true, ty ? pe: () => Date })),</pre>	<>	<pre> prop({ required: true, type: () => Date }) ? ,</pre>
<pre> __metadata("design:type", Date)], WeightPoint.prototype, "timestamp", void 0 ?); __decorate([</pre>	=	<pre> __metadata("design:type", Date)], WeightPoint.prototype, "timestamp", void 0) ? ; __decorate([</pre>
<pre> (0, typegoose_1.prop)({ required: false, t ? ype: () => Number })),</pre>	<>	<pre> prop({ required: false, type: () => Number ? })),</pre>
<pre> __metadata("design:type", Number)], WeightPoint.prototype, "weightKg", void 0) ? ; __decorate([</pre>	=	<pre> __metadata("design:type", Number)], WeightPoint.prototype, "weightKg", void 0); __decorate([</pre>
<pre> (0, typegoose_1.prop)({ required: true, re ? f: () => ContainerSite_1.ContainerSite })),</pre>	<>	<pre> prop({ required: true, ref: () => Containe ? rSite })),</pre>
<pre> __metadata("design:type", Object)], WeightPoint.prototype, "sourceContainerSit ? e", void 0); __decorate([</pre>	=	<pre> __metadata("design:type", Object)], WeightPoint.prototype, "sourceContainerSit ? e", void 0); __decorate([</pre>
<pre> (0, typegoose_1.prop)({ required: true, re ? f: () => ForcePoint_1.ForcePoint })),</pre>	<>	<pre> prop({ required: true, ref: () => ForcePo ? int })),</pre>
<pre> __metadata("design:type", Object)], WeightPoint.prototype, "sourceForcePoint", ? void 0); __decorate([</pre>	=	<pre> __metadata("design:type", Object)], WeightPoint.prototype, "sourceForcePoint", ? void 0); __decorate([</pre>
<pre> (0, typegoose_1.prop)({</pre>	<>	<pre> prop({</pre>
<pre> required: true,</pre>	=	<pre> required: true,</pre>
<pre> type: () => WeightAlgo_1.WeightAlgoBa ? se,</pre>	<>	<pre> type: () => WeightAlgoBase,</pre>
<pre> discriminators: () => WeightAlgo_1.We ? ightAlgoDiscriminators,</pre>		<pre> discriminators: () => WeightAlgoDiscr ? iminators,</pre>
<pre> })),</pre>	=	<pre> })),</pre>
<pre> __metadata("design:type", WeightAlgo_1.We ? ightAlgoBase)</pre>	<>	<pre> __metadata("design:type", WeightAlgoBase)</pre>
<pre>], WeightPoint.prototype, "algorithm", void 0 ?);</pre>	=	<pre>], WeightPoint.prototype, "algorithm", void 0) ? ;</pre>
<pre>exports.WeightPoint = WeightPoint = WeightPoi ? nt_1 = __decorate([</pre>	<>	<pre>WeightPoint = WeightPoint_1 = __decorate([</pre>
<pre> (0, typegoose_1.modelOptions)({</pre>		<pre> modelOptions({</pre>
<pre> schemaOptions: { collection: 'weightp ? oints_v2', timestamps: true },</pre>	=	<pre> schemaOptions: { collection: 'weightpo ? ints_v2', timestamps: true },</pre>
<pre> })),</pre>		<pre> })),</pre>

Left file: /Users/zachtindall/Downloads/WeightPoint-nodenext.js

Right file: /Users/zachtindall/Downloads/WeightPoint-preserve.js

(continued)

<pre>(0, typegoose_1.index)({ timestamp: 1 }}, (0, typegoose_1.index)({ sourceContainerS ? ite: 1, timestamp: 1 }}, (0, typegoose_1.index)({ sourceContainerS ? ite: 1, sourceForcePoint: 1 }}, (0, typegoose_1.index)({ sourceContainerS ? ite: 1, timestamp: -1, weightKg: 1 }}, (0, typegoose_1.index)({ sourceContainerS ? ite: 1, weightKg: 1, timestamp: 1 }, {</pre>	<>	<pre>index({ timestamp: 1 }}, index({ sourceContainerSite: 1, timestamp ? : 1 }}, index({ sourceContainerSite: 1, sourceFor ? cePoint: 1 }}, index({ sourceContainerSite: 1, timestamp ? : -1, weightKg: 1 }}, index({ sourceContainerSite: 1, weightKg: ? 1, timestamp: 1 }, {</pre>
<pre>name: 'sourceContainerSite_1_weightKg ? _1_timestamp_1_weightKg_p', partialFilterExpression: { weightKg: { \$exists: true }, }, }), __metadata("design:paramtypes", [Object])], WeightPoint);</pre>	=	<pre>name: 'sourceContainerSite_1_weightKg ? _1_timestamp_1_weightKg_p', partialFilterExpression: { weightKg: { \$exists: true }, }, }), __metadata("design:paramtypes", [Object])], WeightPoint);</pre>
	-+	<pre>export { WeightPoint };</pre>
<pre>/// sourceMappingURL=WeightPoint.js.map</pre>	=	<pre>/// sourceMappingURL=WeightPoint.js.map</pre>