

```
1  /* eslint-disable no-use-before-define */
2  import {
3    Association,
4    BelongsToManyAddAssociationMixin,
5    BelongsToManyCountAssociationsMixin,
6    BelongsToManyCreateAssociationMixin,
7    BelongsToManyGetAssociationsMixin,
8    BelongsToManyHasAssociationMixin,
9    DataTypes,
10   HasManyAddAssociationMixin,
11   HasManyCountAssociationsMixin,
12   HasManyCreateAssociationMixin,
13   HasManyGetAssociationsMixin,
14   HasManyHasAssociationMixin,
15   HasOneCreateAssociationMixin,
16   HasOneGetAssociationMixin,
17   HasOneSetAssociationMixin,
18   Model,
19 } from 'sequelize';
20
21 import {FriendlyId, UUID} from '@common/domain/
  genericTypes';
22 import {JobRun} from '@common/domain/job/jobRun';
23 import {RUN_STATUS_NOT_STARTED} from '@common/domain/
  job/runStatus';
24 import {StudyPermissions} from '@common/domain/
  permissions/study';
25 import {StudyAnalysisPermissions} from '@common/
  domain/permissions/studyAnalysis';
26 import {Study, StudyAnalysis, StudyAnalysisAttributes
  , StudyAttributes} from '@common/domain/study';
27 import {ComputationType, StudyRun} from '@common/
  domain/studyRun';
28 import {UserSummary} from '@common/domain/user';
29 import {CounterScreenModel} from '@server/infra/
  postgres/models/counterScreenModel';
30 import {ReadsetModel} from '@server/infra/postgres/
  models/readsetModel';
31 import {ReferenceDbModel} from '@server/infra/
  postgres/models/referenceDbModel';
32 import {StudyRunModel} from '@server/infra/postgres/
```

```
32 models/studyRunModel';
33 import {sequelizeOptions} from '@server/infra/
    postgres/sequelizeInstance';
34 import {
35     comment,
36     createdAt,
37     createdBy,
38     description,
39     friendlyId,
40     id,
41     lastModifiedBy,
42     name,
43     permissions,
44     updatedAt,
45 } from '@server/infra/postgres/sharedAttributes';
46
47 export class StudyAnalysisModel extends Model<
    StudyAnalysis, StudyAnalysisAttributes> implements
    StudyAnalysis {
48     declare static associations: {
49         studyRuns: Association<StudyAnalysisModel,
            StudyRunModel>;
50     };
51     declare id: UUID;
52     declare name: string;
53     declare computationParameters: Record<string,
        unknown>;
54     declare computationType: ComputationType;
55     declare description?: string | null;
56     declare friendlyId: FriendlyId;
57     /**
58      * associations
59      */
60     declare studyId: string;
61     /**
62      * Audit
63      */
64     declare createdBy: UserSummary;
65     declare lastModifiedBy: UserSummary;
66     /**
67      * Automatically added by Sequelize. ISO timestamp.
```

```

68      *
69      * **WARNING**: Not really a string! Actually a
      Date.
70      * This only becomes a string after it's
      stringified when sent from the server to the client.
71      * Which means:
72      * * while on server, the real type is Date
73      * * on client, the real type is String
74      */
75      declare readonly createdAt: string;
76      /**
77      * Automatically added by Sequelize. ISO timestamp
      .
78      *
79      * **WARNING**: Not really a string! Actually a
      Date.
80      * This only becomes a string after it's
      stringified when sent from the server to the client.
81      * Which means:
82      * * while on server, the real type is Date
83      * * on client, the real type is String
84      */
85      declare readonly updatedAt: string;
86      declare permissions: StudyAnalysisPermissions;
87      /**
88      * StudyRuns
89      */
90      declare getStudyRuns: HasManyGetAssociationsMixin<
      StudyRunModel>;
91      declare addStudyRun: HasManyAddAssociationMixin<
      StudyRunModel, string>;
92      declare hasStudyRun: HasManyHasAssociationMixin<
      StudyRunModel, string>;
93      declare countStudyRuns:
      HasManyCountAssociationsMixin;
94      declare createStudyRun:
      HasManyCreateAssociationMixin<StudyRunModel>;
95  }
96
97  StudyAnalysisModel.init(
98  {

```

```
99     id,
100     friendlyId,
101     createdAt,
102     updatedAt,
103     createdBy,
104     computationParameters: {
105       type: DataTypes.JSON,
106     },
107     computationType: {
108       type: DataTypes.STRING,
109     },
110     lastModifiedBy,
111     name,
112     description,
113     permissions,
114     studyId: {
115       type: DataTypes.UUID,
116       allowNull: false,
117     },
118   },
119   {
120     ...sequelizeOptions,
121     tableName: 'analysis',
122   },
123 );
124 export class StudyModel extends Model<Study,
125   StudyAttributes> implements Study {
126   declare id: UUID;
127   declare studyNumber: string;
128   declare subject: string;
129   declare description?: string | null;
130   declare tags: string[];
131   declare friendlyId: FriendlyId;
132   declare name: string;
133   /**
134    * Audit
135    */
136   declare createdBy: UserSummary;
137   declare lastModifiedBy: UserSummary;
138
```

```
139  /**
140   * Automatically added by Sequelize. ISO timestamp
141   *
142   * **WARNING**: Not really a string! Actually a
    Date.
143   * This only becomes a string after it's
    stringified when sent from the server to the client.
144   * Which means:
145   * * while on server, the real type is Date
146   * * on client, the real type is String
147   */
148  declare readonly createdAt: string;
149  /**
150   * Automatically added by Sequelize. ISO timestamp
151   *
152   * **WARNING**: Not really a string! Actually a
    Date.
153   * This only becomes a string after it's
    stringified when sent from the server to the client.
154   * Which means:
155   * * while on server, the real type is Date
156   * * on client, the real type is String
157   */
158  declare readonly updatedAt: string;
159
160  declare permissions: StudyPermissions;
161
162  /**
163   * StudyRuns
164   */
165  declare getStudyRuns: HasManyGetAssociationsMixin<
    StudyRunModel>;
166  declare addStudyRun: HasManyAddAssociationMixin<
    StudyRunModel, string>;
167  declare hasStudyRun: HasManyHasAssociationMixin<
    StudyRunModel, string>;
168  declare countStudyRuns:
    HasManyCountAssociationsMixin;
169  declare createStudyRun:
```

```

169 HasManyCreateAssociationMixin<StudyRunModel>;
170
171 /**
172  * Readsets
173  */
174 declare getReadsets:
  BelongsToManyGetAssociationsMixin<ReadsetModel>;
175 declare addReadset:
  BelongsToManyAddAssociationMixin<ReadsetModel,
  string>;
176 declare removeReadset:
  BelongsToManyAddAssociationMixin<ReadsetModel,
  string>;
177 declare hasReadset:
  BelongsToManyHasAssociationMixin<ReadsetModel,
  string>;
178 declare countReadsets:
  BelongsToManyCountAssociationsMixin;
179 declare createReadset:
  BelongsToManyCreateAssociationMixin<ReadsetModel>;
180
181 /**
182  * inclusions are optional, populated when
  explicitly include a relation
183  */
184 declare readonly readsets?: ReadsetModel[];
185
186 declare static associations: {
187   studyRuns: Association<StudyModel, StudyRunModel>;
188   readsets: Association<StudyModel, ReadsetModel>;
189   analysis: Association<StudyModel,
  StudyAnalysisModel>;
190 };
191 }
192
193 export class StudyRunModel extends Model<StudyRun,
  StudyRunCreationAttributes> implements StudyRun {
194   declare id: UUID;
195   declare friendlyId: FriendlyId;
196   declare name: string;

```

```
197 declare description: string;
198 declare computationParameters: Record<string,
    unknown>;
199 declare computationType: ComputationType;
200 declare pipeline: JobRun;
201 declare comment: string;
202
203 /**
204  * Audit
205  */
206 declare createdBy: UserSummary;
207 declare lastModifiedBy: UserSummary;
208
209 /**
210  * Automatically added by Sequelize. ISO timestamp
211  *
212  * **WARNING**: Not really a string! Actually a
    Date.
213  * This only becomes a string after it's
    stringified when sent from the server to the client.
214  * Which means:
215  * * while on server, the real type is Date
216  * * on client, the real type is String
217  */
218 declare readonly createdAt: string;
219 /**
220  * Automatically added by Sequelize. ISO timestamp
221  *
222  * **WARNING**: Not really a string! Actually a
    Date.
223  * This only becomes a string after it's
    stringified when sent from the server to the client.
224  * Which means:
225  * * while on server, the real type is Date
226  * * on client, the real type is String
227  */
228 declare readonly updatedAt: string;
229
230 /**
```

```

231     * associations
232     */
233     declare studyId: string;
234     declare readsetId: string;
235     declare referenceDbId: string;
236     declare analysisId: string;
237
238     /**
239      * Reference DBs
240      */
241     declare getReferenceDb: HasOneGetAssociationMixin<
        ReferenceDbModel>;
242     declare setReferenceDb: HasOneSetAssociationMixin<
        ReferenceDbModel, string>;
243     declare createReferenceDb:
        HasOneCreateAssociationMixin<ReferenceDbModel>;
244
245     /**
246      * Readset
247      */
248     declare getReadset: HasOneGetAssociationMixin<
        ReadsetModel>;
249     declare setReadset: HasOneSetAssociationMixin<
        ReadsetModel, string>;
250     declare createReadset:
        HasOneCreateAssociationMixin<ReadsetModel>;
251
252     /**
253      * Counter Screen
254      */
255     declare getCounterScreens:
        HasManyGetAssociationsMixin<CounterScreenModel>;
256     declare createCounterScreen:
        HasManyCreateAssociationMixin<CounterScreenModel>;
257
258     declare static associations: {
259         study: Association<StudyRunModel, StudyModel>;
260         readset: Association<StudyRunModel, ReadsetModel
261         >;
261         refDb: Association<StudyRunModel,
        ReferenceDbModel>;

```



```
262     counterScreens: Association<StudyRunModel,  
      CounterScreenModel>;  
263   };  
264 }  
265  
266 StudyRunModel.init(  
267   {  
268     id,  
269     friendlyId,  
270     name,  
271     description,  
272     comment,  
273     computationParameters: {  
274       type: DataTypes.JSON,  
275     },  
276     computationType: {  
277       type: DataTypes.STRING,  
278     },  
279     createdBy,  
280     lastModifiedBy,  
281     createdAt,  
282     updatedAt,  
283     studyId: {  
284       type: DataTypes.UUID,  
285       allowNull: false,  
286     },  
287     analysisId: {  
288       type: DataTypes.UUID,  
289       allowNull: false,  
290     },  
291     readsetId: {  
292       type: DataTypes.UUID,  
293     },  
294     referenceDbId: {  
295       type: DataTypes.UUID,  
296     },  
297     pipeline: {  
298       type: DataTypes.JSON,  
299       allowNull: false,  
300       defaultValue: {  
301         status: RUN_STATUS_NOT_STARTED,
```

```
302         hpcJob: null,  
303     },  
304 },  
305 },  
306 {  
307     ...sequelizeOptions,  
308     tableName: 'studyRuns',  
309 },  
310 );  
311
```