

```
1 using System;
2
3 namespace MusicStore.Logic
4 {
5     public class Factory
6     {
7         public enum PersistenceType
8         {
9             Csv,
10            Ser,
11        }
12        public static PersistenceType Persistence { get; set; } =
13            Factory.PersistenceType.Csv;
14        private static Context.ContextObject CreateContext()
15        {
16            Context.ContextObject result = null;
17
18            if (Persistence == PersistenceType.Csv)
19            {
20                result = new Context.CsvMusicStoreContext();
21            }
22            else if (Persistence == PersistenceType.Ser)
23            {
24                result = new Context.SerMusicStoreContext();
25            }
26            return result;
27        }
28        public static IController<T> CreateController<T>() where T :
29            Contracts.IIdentifiable
30        {
31            IController<T> result = null;
32
33            if (typeof(T) == typeof(Contracts.IGenre))
34            {
35                result = (IController<T>)CreateGenreController();
36            }
37            else if (typeof(T) == typeof(Contracts.IArtist))
38            {
39                result = (IController<T>)CreateArtistController();
40            }
41            else if (typeof(T) == typeof(Contracts.IAlbum))
42            {
43                result = (IController<T>)CreateAlbumController();
44            }
45            else if (typeof(T) == typeof(Contracts.ITrack))
46            {
47                result = (IController<T>)CreateTrackController();
48            }
49            return result;
50        }
51        public static IController<T> CreateController<T>(object
52            sharedController) where T : Contracts.IIdentifiable
53        {
54            IController<T> result = null;
```

```
54         if (typeof(T) == typeof(Contracts.IGenre))
55         {
56             result = (IController<T>)CreateGenreController
57                 (sharedController);
58         }
59         else if (typeof(T) == typeof(Contracts.IArtist))
60         {
61             result = (IController<T>)CreateArtistController
62                 (sharedController);
63         }
64         else if (typeof(T) == typeof(Contracts.IAlbum))
65         {
66             result = (IController<T>)CreateAlbumController
67                 (sharedController);
68         }
69         else if (typeof(T) == typeof(Contracts.ITrack))
70         {
71             result = (IController<T>)CreateTrackController
72                 (sharedController);
73         }
74         return result;
75     }
76
77     public static IController<Contracts.IGenre> CreateGenreController()
78     {
79         return new Controllers.GenreController(CreateContext());
80     }
81     public static IController<Contracts.IGenre> CreateGenreController
82         (object sharedController)
83     {
84         if (sharedController == null)
85             throw new ArgumentNullException(nameof(sharedController));
86
87         Controllers.ControllerObject controller =
88             (Controllers.ControllerObject)sharedController;
89
90         return new Controllers.GenreController(controller);
91     }
92
93     public static IController<Contracts.IArtist> CreateArtistController()
94     {
95         return new Controllers.ArtistController(CreateContext());
96     }
97     public static IController<Contracts.IArtist> CreateArtistController
98         (object sharedController)
99     {
100         if (sharedController == null)
101             throw new ArgumentNullException(nameof(sharedController));
102
103         Controllers.ControllerObject controller =
104             (Controllers.ControllerObject)sharedController;
105
106         return new Controllers.ArtistController(controller);
107     }
108
109     public static IController<Contracts.IAlbum> CreateAlbumController()
```

```
102     {
103         return new Controllers.AlbumController(CreateContext());
104     }
105     public static IController<Contracts.IAlbum> CreateAlbumController  ↗
106         (object sharedController)
107     {
108         if (sharedController == null)
109             throw new ArgumentNullException(nameof(sharedController));
110
111         Controllers.ControllerObject controller =  ↗
112             (Controllers.ControllerObject)sharedController;
113
114         return new Controllers.AlbumController(controller);
115     }
116     public static IController<Contracts.ITrack> CreateTrackController()
117     {
118         return new Controllers.TrackController(CreateContext());
119     }
120     public static IController<Contracts.ITrack> CreateTrackController  ↗
121         (object sharedController)
122     {
123         if (sharedController == null)
124             throw new ArgumentNullException(nameof(sharedController));
125
126         Controllers.ControllerObject controller =  ↗
127             (Controllers.ControllerObject)sharedController;
128
129         return new Controllers.TrackController(controller);
130     }
131 }
```