

```
1 namespace AdventureLog.Migrations
2 {
3     using AdventureLog.Models;
4     using Microsoft.AspNet.Identity;
5     using Microsoft.AspNet.Identity.EntityFramework;
6     using System;
7     using System.Data.Entity;
8     using System.Data.Entity.Migrations;
9     using System.Linq;
10
11     internal sealed class Configuration : DbMigrationsConfiguration<AdventureLog.Models.ApplicationDbContext> ↗
12     {
13         public Configuration()
14         {
15             AutomaticMigrationsEnabled = true;
16             AutomaticMigrationDataLossAllowed = true;
17         }
18
19         protected override void Seed(AdventureLog.Models.ApplicationDbContext context ↗)
20         {
21             // This method will be called after migrating to the latest ↗
22             // version.
23             SeedRoles(context);
24             SeedAdminUser(context);
25             SeedPlayerRole(context);
26         }
27
28         /// <summary>
29         /// Adds Asp.Net user Roles
30         /// </summary>
31         /// <param name="context"></param>
32         private void SeedRoles(ApplicationDbContext context)
33         {
34             if (!context.Roles.Any(r => r.Name == "Admin"))
35             {
36                 var store = new RoleStore<IdentityRole>(context);
37                 var manager = new RoleManager<IdentityRole>(store);
38                 var role = new IdentityRole
39                 {
40                     Name = "Admin"
41                 };
42                 manager.Create(role);
43             }
44         }
45     }
46
47     /// <summary>
48     /// test@gmail.com
49     /// Test123456!
50     /// </summary>
51     /// <param name="context"></param>
52     private void SeedAdminUser(ApplicationDbContext context)
53     {
```

```
54         if (!context.Users.Any(u => u.UserName == "AdventureLogAdmin"))
55         {
56             var store = new UserStore<ApplicationUser>(context);
57             var manager = new UserManager<ApplicationUser>(store);
58             var user = new ApplicationUser
59             {
60                 UserName = "AdventureLogAdmin",
61                 Email = "adventurequestlog@gmail.com",
62                 LastLogin = DateTime.Now,
63                 JoinDate = DateTime.Now
64             };
65
66             manager.Create(user, "Test123456!");
67             manager.AddToRole(user.Id, "Admin");
68         }
69     }
70
71     private void SeedPlayerRole(ApplicationDbContext context)
72     {
73         if (!context.PlayerRoles.Any(r => r.PlayerRole_PK == (int)
74             PlayerRole.PlayerRoleKey.Player))
75         {
76             var role = new PlayerRole()
77             {
78                 PlayerRole_PK = (int)PlayerRole.PlayerRoleKey.Player,
79                 RoleName = "Player",
80                 RoleDescription = "Player in a game, can comment and
81                 view",
82                 CreatedDate = DateTime.Now,
83                 LastModifiedDate = DateTime.Now,
84                 LastModifiedUser = "SysAdmin"
85             };
86             context.PlayerRoles.Add(role);
87         }
88         if (!context.PlayerRoles.Any(r => r.PlayerRole_PK == (int)
89             PlayerRole.PlayerRoleKey.Gamemaster))
90         {
91             var role = new PlayerRole()
92             {
93                 PlayerRole_PK = (int)PlayerRole.PlayerRoleKey.Gamemaster,
94                 RoleName = "Gamemaster",
95                 RoleDescription = "Gamemaster of a game, can control all
96                 features of the adventure",
97                 CreatedDate = DateTime.Now,
98                 LastModifiedDate = DateTime.Now,
99                 LastModifiedUser = "SysAdmin"
100             };
101             context.PlayerRoles.Add(role);
102         }
103         context.SaveChanges();
104     }
105 }
```

106 }

107