API

https://cracklier-alia-uninserted.ngrok-free.dev

SignalR

https://cracklier-alia-uninserted.ngrok-free.dev/broadcastHub

Figma UI

https://www.figma.com/make/LblQuhdKOezKs8qXHGhFdd/Playout-Application-Dashboard?p=f&t=r5ZDfj4LTorXey2S-0

Task

1. Create Log in panel
2. Create interface from Figma, with dynamic left menu
3. Add functions to Left menu and Playouts table (playouts table changes status, left menu change menus, all of that using signalR)

You can start working on task 1 and task 2, for 3 I must set SignalR how to work with token. Main problem is that I have desktop dashboard window which works without token, since there is no login for that, so just to find best solution to mix those things

LOGIN

For login, use this call

api/Auth/login

Username: admin

Password: admin

Use LoginRequest to send username and password

```
public class LoginRequest
{
    public required string Username { get; set; }
    public required string Password { get; set; }
}
```

If everything is ok you will receive json login response

```
public class LoginResponse
{
    public string? Token { get; set; }
    public string? RefreshToken { get; set; }
    public DateTime ExpiresAt { get; set; }
    public UserInfoDto? User { get; set; }
}
```

And important thing here, beside token is UserInfoDto. These are information about user, what playout he can work on, what channel to work on. And we define that in left menu.

```
public class UserInfoDto
 {
     public required string Id { get; set; }
     public string? Username { get; set; }
     public string? Name { get; set; }
     public int? Role { get; set; }
     public byte[]? Picture { get; set; }
     public List<PlayoutListDto>? AllPlayouts { get; set; }
     public List<PlayoutListDto>? RemotePlayouts { get; set; }
     public List<ChannelListDto>? SchedulerChannels { get; set; }
 }
```

REFRESH

You can refresh token by calling api/Auth/refresh, and you send RefreshTokenRequest

```
public class RefreshTokenRequest
 {
     public string? RefreshToken { get; set; }
 }
```

You will get same LoginResponse

LOGOUT

And finally, you can log out by calling api/Auth/logout and you again send RefreshTokenRequest

ROLES

We take user’s role from public int? Role { get; set; } There are 4 roles

```
public enum UserRole
 {
     Supervisor = 0,
     Administrator = 1,
     Operator = 2,
     Viewer = 3

 }
```

Supervisor – can do everything on system, he can access everything, so everything in left menu is visible for him

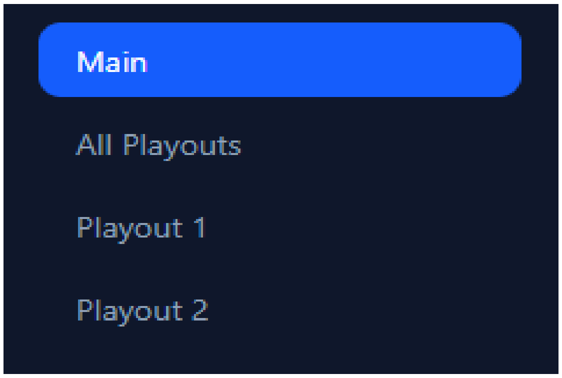
Administrator – Dashboard and Administration are visible, everything else hidden

Operator – everything is visible except Administration

Viewer – only Dashboard is visible

User admin has supervisor role.

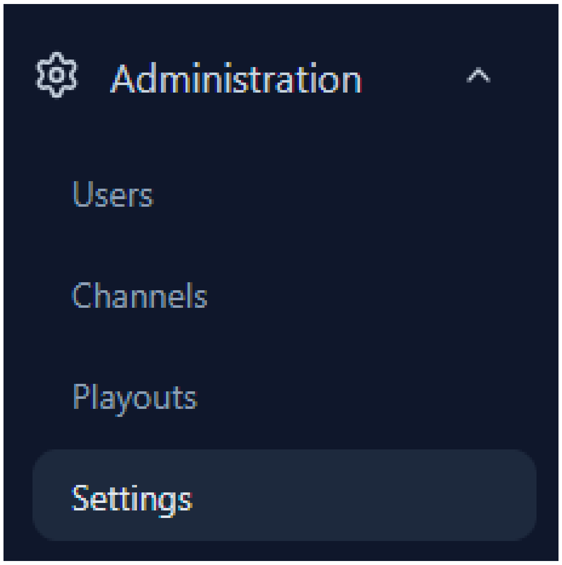
Role overrides RemotePlayouts and SchedulerChannels. So even if there are remote playouts or scheduler channels in lists, if user role is viewer or administrator, Playout Control menu and Scheduler menu will be hidden.



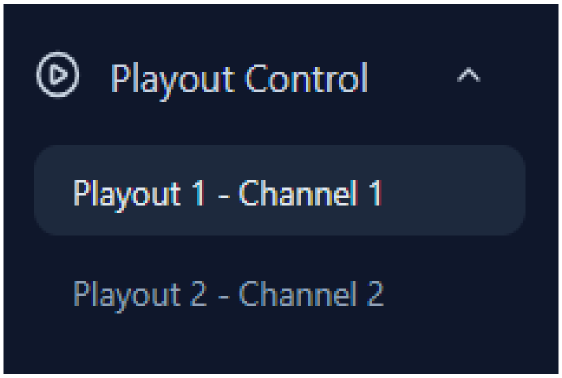
Once you click on Main, you get this screen I sent you in Figma, that’s our homepage. Below, you add All Playouts fixed submenu item, and all other playouts. You can get playouts from this list
List<PlayoutListDto>? AllPlayouts { get; set; }
So everyone can see dashboard of all playouts. If playout is Spare (spare == true), add icon as in Figma Playout 3

```
public class PlayoutListDto

 {
     public required string Id { get; set; }
     public string? Name { get; set; }
     public string? Channel { get; set; }
     public bool? Spare { get; set; }
 }
```



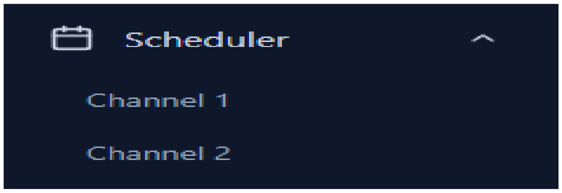
For administration, everything is fixed.



This menu is dynamic, and we get data from this List

```
List<PlayoutListDto>? RemotePlayouts { get; set; }
```

You get id, playout name and channel, so combine PlayoutName – Channel for name in the menu. If Playout has no channel, then just type playout name (without -). If playout is Spare (spare == true), put same icon as in Figma. If List is empty, then hide Playout Control menu

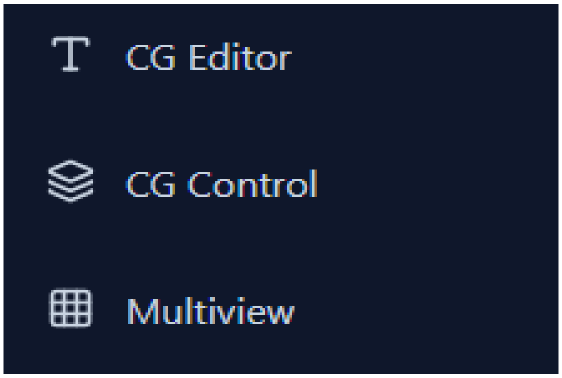


This menu is also dynamic, and we get data from this List

```
List<ChannelListDto>? SchedulerChannels { get; set; }
```

You will get id and name of channels which user can access in Scheduler. If this list is empty, hide Scheduler menu.

```
    public class ChannelListDto
    {
        public required string Id { get; set; }
        public string? Name { get; set; }
    }
```



Just add these things as they are, they are phase 2, we will probably comment them in code or hide for now, but put them to see how it looks