Top of Form

Task: View this topic

Bottom of Form

**Description**

In UnityConfig.cs

Register our custom types for PlayerService and its respective interfaces

In BundleConfig.cs

Create the '~/bundles/players' bundle consisting of the Scripts/Players/PlayerCallbacks.js script

In PlayerService.cs

Create a constructor with all dependencies, so Unity will inject them

in PlayerController.cs

Create a constructor with all dependencies, so Unity will inject them  
Change the Edit POST Action to return a JSON with the property 'text' equal to 'Success' should the request succeed, 'Something went wrong' otherwise

In the Edit GET Action; create the model to be passed to the view. Set TeamsDropDownItems equal to all distinct team names

In the GetPlayersOnTeam action; Create the \_PlayerListPartialView  
In the Index; Return all distinct Team names to the index view instead of all players  
In the Create GET Action; Create the property TeamDropDownList by getting all distinct team names

In Views/Players/Index

Change the model directive to IEnumerable String

Instead of iterating over a collection of players and displaying the list, iterate over the string  
and for every string, display a button that will make an AJAX request to update the ajax-target div  
with the returned PartialView. This request should go to Players/GetPlayersOnTeam, and the string will be the team routeValue

In Views/Players/Edit

Change the model directive to be of type TeamManager.Models.ViewModel.PlayerEditViewModel  
Add a drop down list for the property TeamName, using the TeamDropDownItems property  
Change the Html.Beginform to an ajax request, invoking the SuccessfulRequest and FailedRequests as callbacks  
Include the script bundle for Players

In Views/Players/Create

Change the model directive to be of type TeamManager.Models.ViewModel.PlayerCreateViewModel  
Add a drop down list for the property TeamName, using the TeamDropDownItems property