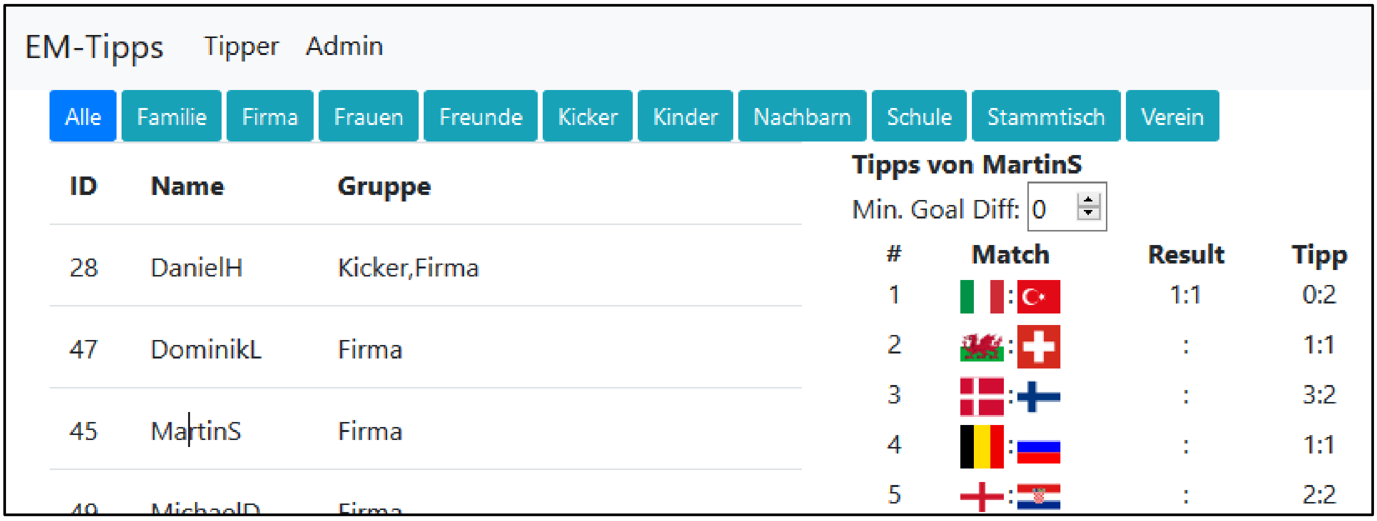
Tipperliste mit Angular

Write an Angular App with which you can view the tipps for the European Football Championship 2020.



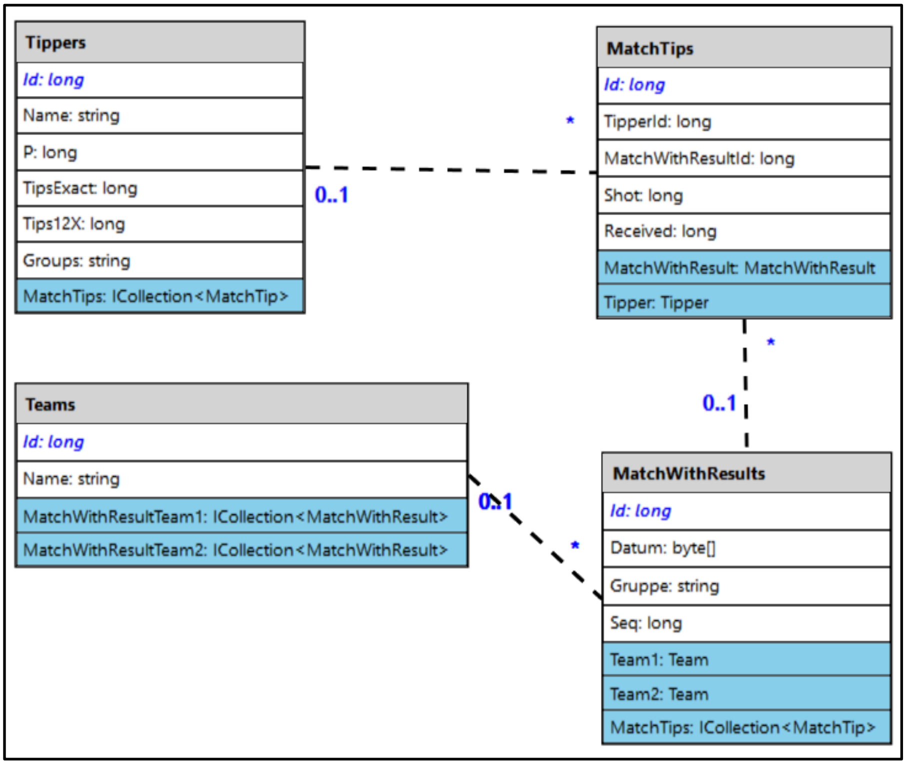
ID	Name	Gruppe
28	DanielH	Kicker,Firma
47	DominikL	Firma
45	MartinS	Firma
49	MichaelD	Firma

Tipps von MartinS

Min. Goal Diff: 0

#	Match	Result	Tipp
1	:	1:1	0:2
2	:	:	1:1
3	:	:	3:2
4	:	:	1:1
5	:	:	2:2

The backend is already available as a WebAPI project. There is nothing more to be programmed.



Tippers
Id: long
Name: string
P: long
TipsExact: long
Tips12X: long
Groups: string
MatchTips: ICollection<MatchTip>

MatchTips
Id: long
TipperId: long
MatchWithResultId: long
Shot: long
Received: long
MatchWithResult: MatchWithResult
Tipper: Tipper

Teams
Id: long
Name: string
MatchWithResultTeam1: ICollection<MatchWithResult>
MatchWithResultTeam2: ICollection<MatchWithResult>

MatchWithResults
Id: long
Datum: byte[]
Gruppe: string
Seq: long
Team1: Team
Team2: Team
MatchTips: ICollection<MatchTip>

Modules

The following modules shall be generated:

- Core
- Shared
- Bootstrap as additional Shared Module
- Tipper as Eager Loaded Module
- Admin als Lazy Loaded Module

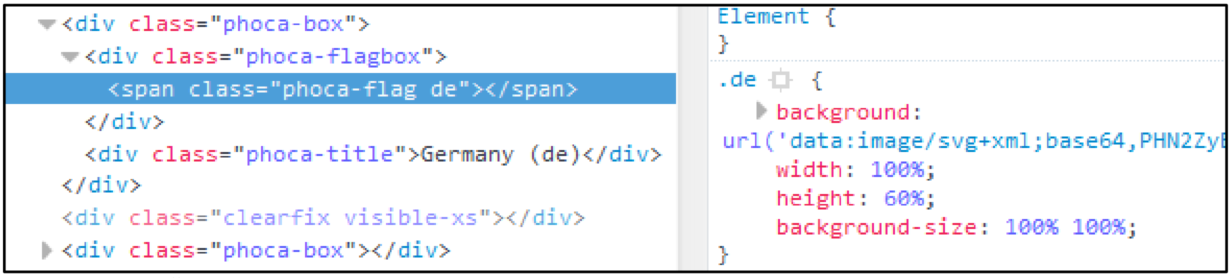
1) Module Core

Contains the service tipps.service.ts which implements the communication with the TipsController of the backend.

2) Module Shared

Contains three pipes and two components

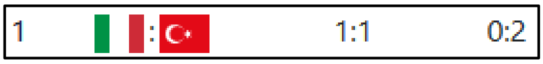
- **Pipe** `goalDiff:minDiff`: This pipe filters only those matches where the goal difference is greater than the specified value.
- **Pipe** `filterByGroup:groupName`: filters thos tippers that belong to the given group.
- **Pipe** `result`: displays the result of a SintleTippDto or a MatchDto in the form ***shot:received***. Bear in mind, that both values are nullable and you should avoid display strings as ***null:null***.
- **Component** `flag`: displays a flag symbol instead of the country name. You can download all these symbols as bsae64-CSS from http://www.phoca.cz/cssflags. Note: names are different than those in the database, e.g. „Ger“ in database but „de“ in CSS.



```
▼<div class="phoca-box">
  ▼<div class="phoca-flagbox">
      <span class="phoca-flag de"></span>
    </div>
    <div class="phoca-title">Germany (de)</div>
  </div>
  <div class="clearfix visible-xs"></div>
 ▶<div class="phoca-box"></div>
```

```
Element {
}
.de {
  ▶ background:
url('data:image/svg+xml;base64,PHN2ZyE
    width: 100%;
    height: 60%;
    background-size: 100% 100%;
}
```

  The only Input() therefore is the name of the country.
- **Component** `match-result`: displays a single match and the corresonding tipp as one line:



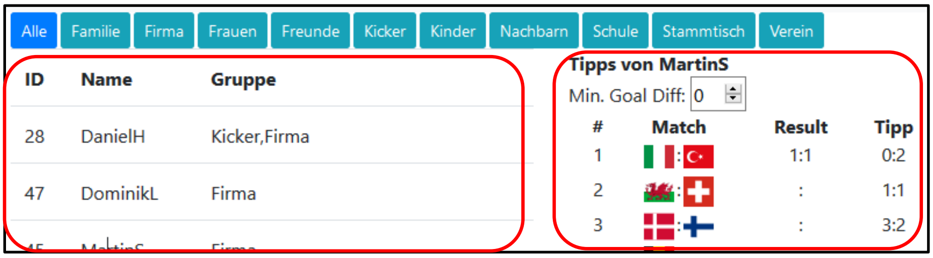
```
1   :   1:1   0:2
```

  So you need two Input() variables of type MatchDto and SingleTippDto, respectively.

3) Module Bootstrap

Create a shared module for bootstrap components and add a component for button and navbar.

4) Module Tipper



ID	Name	Gruppe
28	DanielH	Kicker,Firma
47	DominikL	Firma
45	MartinS	Firma

Tipps von MartinS

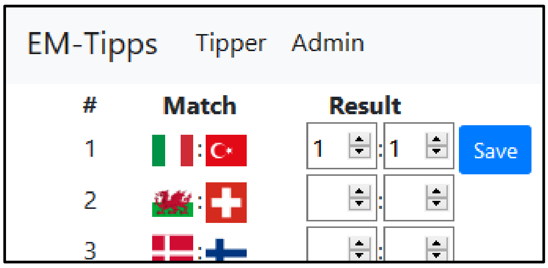
Min. Goal Diff: 0

#	Match	Result	Tipp
1	:	1:1	0:2
2	:	:	1:1
3	:	:	3:2

This module shall be loaded at startup (eager module). It shall consist of three components:

- **Component** `tipper-overview`: Contains the other two components of this module and one button for each different group the tippers belong to. Clicking one of these buttons will display only those tippers in the left component that belong to this group.
  - Input: none
  - Output: none
- **Component** `tipper-names`: Contains the list of tippers with respect to the above mentioned button click, using pipe filterByGroup.
  - Input: TipperDto[], selectedGroup
  - Output: tipperIdSelected
- **Component** `tipped-matches`: Displays all matches as shown with the tipped result of the selected tipper. Use the comonent match-result for displaying the matches and the pipe goalDiff to filter the list.
  - Input: SingleTippDto[], tipperName
  - Output: none

5) Module Admin



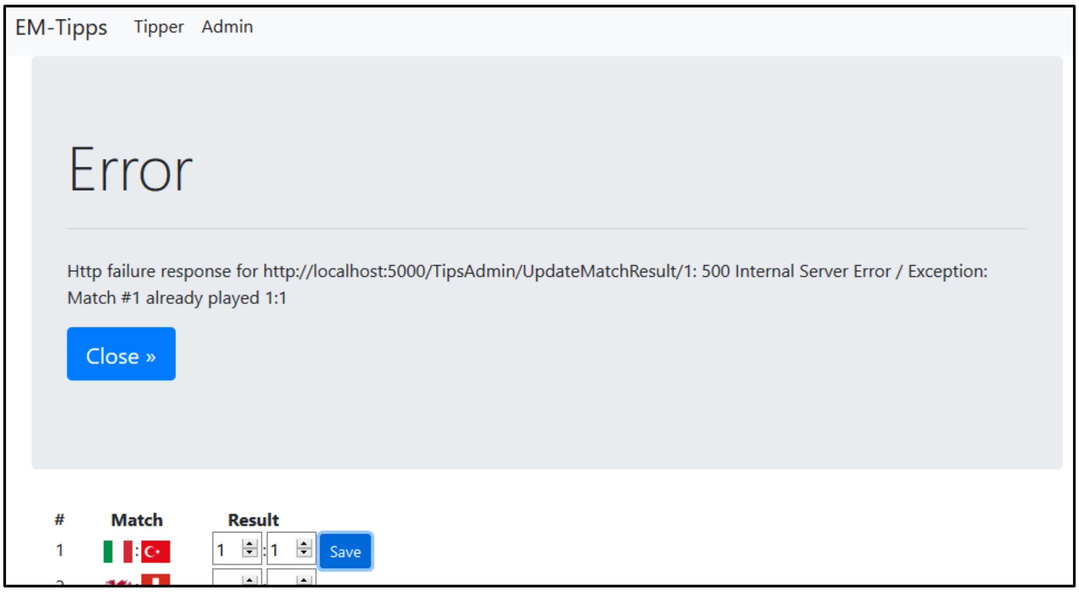
#	Match	Result
1	:	1 : 1 Save
2	:	:
3	:	:

When clicking „Admin“ of the navbar the only component of this module shall be displayed. It displays all matches for the admin so that a new result can be entered and sent to the backend.

You can adapt the match-result component or create a new one.

Create a new service admin.service.ts that talks to the TipsAdminController of the backend.

6) Extension



Error

Http failure response for http://localhost:5000/TipsAdmin/UpdateMatchResult/1: 500 Internal Server Error / Exception: Match #1 already played 1:1

#	Match	Result
1	:	1 : 1 Save

Write an error interceptor that notifies the user in case of a communication error. For this you need the following:

- Error interceptor in the core module
- Notifier/Distributor service in the core module
- Bootstrap jumbotron in app.comonent which therefore acts as a listener of the service.