

As defined in the problem statement the user interface contains three main important things with respect to the scheduled game

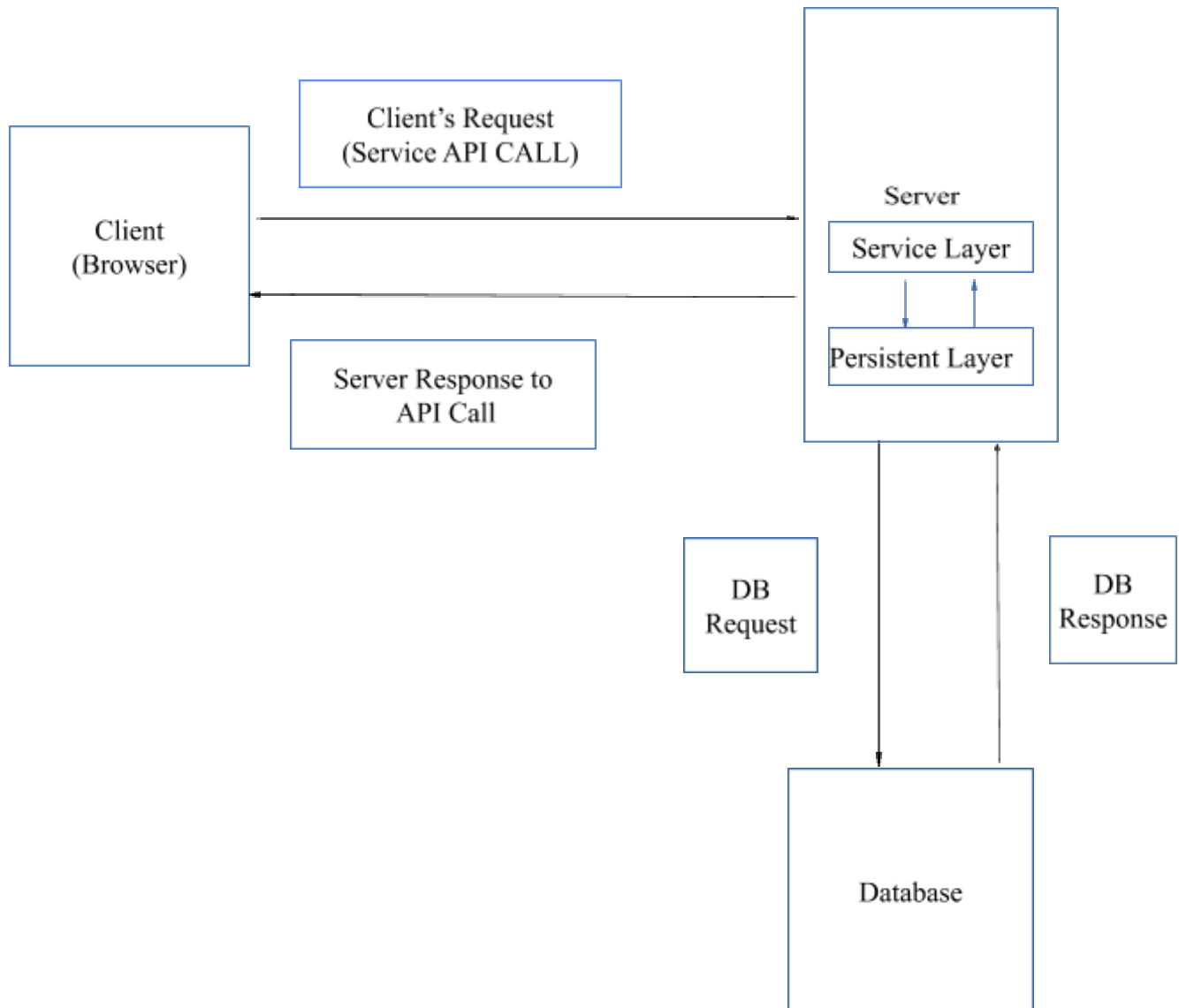
1.The opponent

2.The date

3.The Time

along with that location and other information is also there.

**Overall Client-Server Communication and Architecture.**



The Above UI Components can be displayed and modified by various service layer API calls by using different HTTP request methods for different operations.

**1.GET Method** – The GET HTTP method is used to get the resource from the application.In this case the GET method is used to display all the scheduled fixtures of the team that include Opponent Team, Scheduled Date and Scheduled Time.

Different Scenarios of GET methods.

Case	Http Response Code	Response(Json Object)
Details of all Scheduled Fixtures of Team	200(Ok)	{“status”:200, “message”:”Success”, gameId: "1234567",sqlId: "1234567",date: "2016-01-01T19h00h00",opp onent: "TestOpponent",opponentId: "123456",isHome: true,gameType: 0,categories: [ ]}  For each opponent team
If there are not any matches for the current team in DB.	204(No Data)	{“status”:204, “message”:”No Data”, “Error”:”No data Found”} Empty Array
Opponent Teams Info(Available)	200(Ok)	{“status”:200, “message”:” Success”, “opponent”: Arsenal, “opponentID”:”123567”, “Coach”:”Arteta”, “Ground”:Emirates stadium }
Opponent Teams Info(Registered in DB but Not Available)	204(No Data)	{“status”:204, “message”:”No Data”, “Error”:”No data Found”} Empty Array
Opponent is not present in DB	404(Resource not Found)	{“status”:404, “message”:”Not Found”, “Error”:”Resource Not Found”}
Due to server problem unable to display scheduled fixtures	500(Internal Server Error)	{ “status”:500, “Message”:”Internal Server Error” “Error”:”Internal Server Problem” }

## 2.POST Method -

If Coach wants to add new fixture. Then it will give OpponentName,Scheduled Date and Scheduled Time over the HTTP Post request to the service layer then the Service layer will check all the info as well as will authenticate it and then it will give to the DAO layer for the database updation. This is a happy path but there can be many situations that will be there.

Case	HTTP Response Code	Object Attached To the URL	Response
When New Match is Added	201(Created)	{ "opponent":Arsenal, "date": "2016-01-01T19h00h00", isHome:true, gameType: 0, }	{ "status":201, "message":."Created", gameId:"1234567", sqlId:"1234567", date: "2016-01-01T19h00h00", opponent:"Arsenal", opponentId:"123456", isHome:true, gameType: 0, categories: [ ] } The new raw will be created in the DB
When Coach is not allowed to add new match because it is unauthorized	403(Forbidden)	{ "opponent":."Liverpool", "date": "2016-01-01T19h00h00", isHome:true, gameType: 0, }	{ "status":404, "Message":."Forbidden", "Error":." Not Authenticated to Resource Access" }
When Coach is unable to add new fixture to his current team due to the server error	500(Internal Server Error)	{ "opponent":."Liverpool", "date": "2016-01-01T19h00h00", isHome:true, gameType: 0, }	{ "status":500, "Message":."Internal Server Error", "Error":." Internal Server Problem" }

**3.Put Method** – When the coach wants to change the scheduled matches timing or date or any other game-related thing then the Put method is used for API calling along with Request object is sent over the URL for making effective changes in the DB.

Case	HTTP Response Code	Request Object Sent Over URL	Response
When Coach updated the Scheduled Date or Time successfully	200(Ok)	{ gameId: "1234567",sqlId: "1234567",date: "2018-01-01T18h00h 00",opponent: "TestOpponent",oppo nentId: "123456",isHome: true,gameType: 0,categories: [ ] }	{status:200, message:"Updated", gameId: "1234567",sqlId: "1234567",date: "2016-01-01T19h00h 00",opponent: "TestOpponent",oppo nentId: "123456",isHome: true,gameType: 0,categories: [ ]}
When Coach is not authenticated to conduct any operations.	401(UnAuthorized)	{ gameId: "1234567",sqlId: "1234567",date: "2018-01-01T18h00h 00",opponent: "TestOpponent",oppo nentId: "123456",isHome: true,gameType: 0,categories: [ ] }	{ "status":401, "Message": UnAuthorized", "Error": " Not Authenticated to do any operations" }
When Coach is authenticated to but due to server policies access is forbidden.	403(Forbidden)	{ gameId: "1234567",sqlId: "1234567",date: "2018-01-01T18h00h 00",opponent: "TestOpponent",oppo nentId: "123456",isHome: true,gameType: 0,categories: [ ] }	{ "status":403, "Message": Forbidden", "Error": " Not have access rights." }

**4.Delete Method** – When Client wants to delete the web resource completely then the Delete method is used.

Case	HTTP Response Code	Object Attached To the URL	Response
For the first time coach wants to delete the scheduled match	200(Ok)	{ gameId: "1234567",sqlId: "1234567",date: "2018-01-01T18h00h00",opponent: "TestOpponent",opponentId: "123456",isHome: true,gameType: 0,categories: [ ] }	{ "status":200, "Message":"Deleted", }
The coach is trying to delete the already deleted match	404(Not Found)	{ gameId: "1234567",sqlId: "1234567",date: "2018-01-01T18h00h00",opponent: "TestOpponent",opponentId: "123456",isHome: true,gameType: 0,categories: [ ] }	{"status":404, "message":"Not Found", "Error":" Resource Not Found"}
The Coach is not allowed to make any changes in the web resources due to permissions and access rights.	403(Forbidden)	{ gameId: "1234567",sqlId: "1234567",date: "2018-01-01T18h00h00",opponent: "TestOpponent",opponentId: "123456",isHome: true,gameType: 0,categories: [ ] }	{ "status":403, "Message":"Forbidden", "Error":"Not have access rights." }

