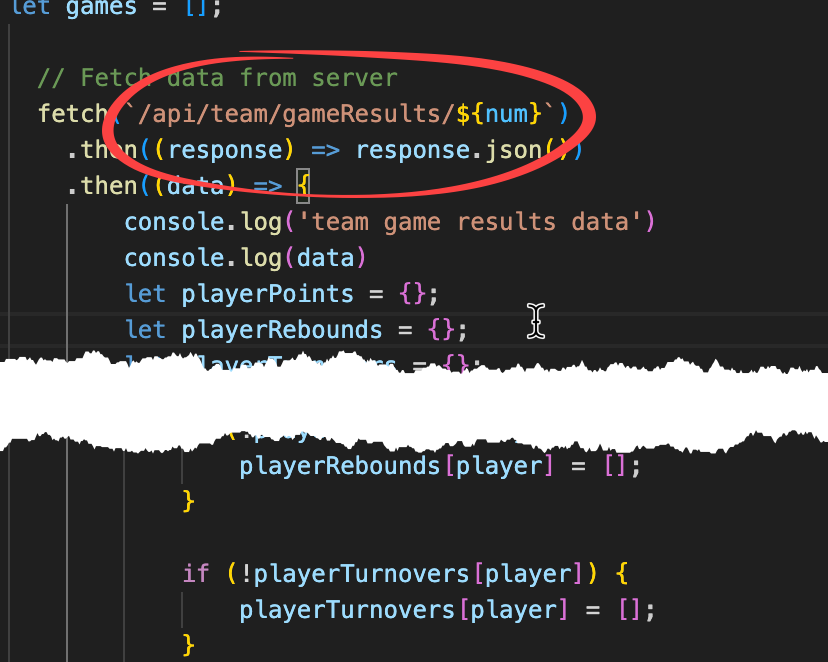
Browser URL: /pages/team?team=1

This brings up the web page but not the data used to populate the charts. The javascript below will do that.



File: /public/js/stackedBar.js

The javascript file does another web request (“fetch”) to get the data to populate the chart(s). This is called an AJAX request. We need to find out how we get data from the database using: /api/team/gameResults



All /api requests follow this pattern:

Javascript -> Route -> Controller -> Service -> DB (get the data)

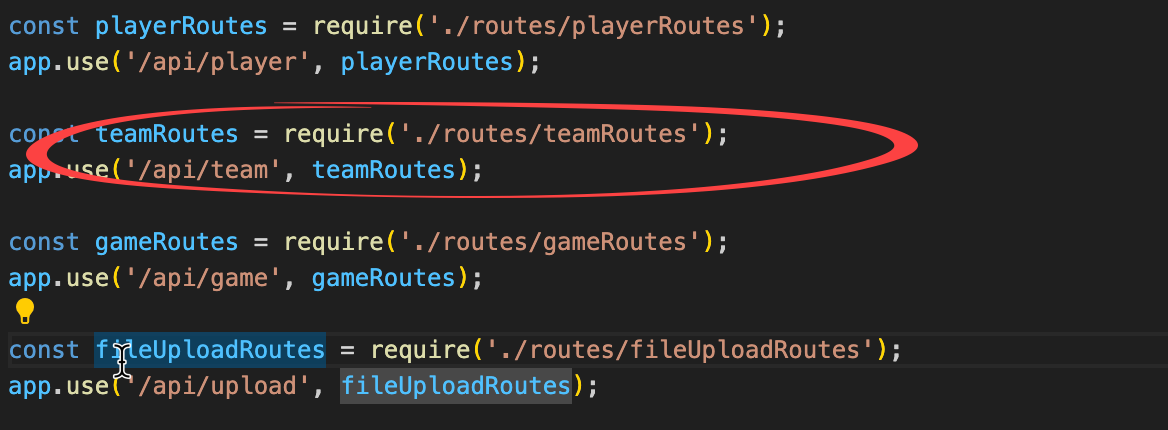
Then the data goes back …

DB -> Service -> Controller -> Javascript (transform the data to match what webpage/chart needs)

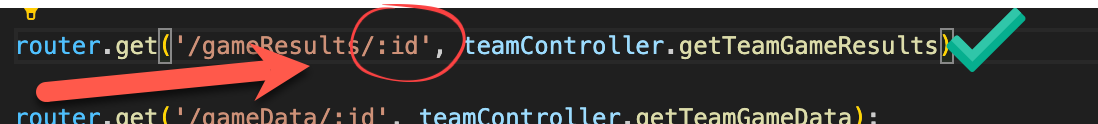
Below are the steps …

The main webserver file bcbServer.js sets up how to process routes:

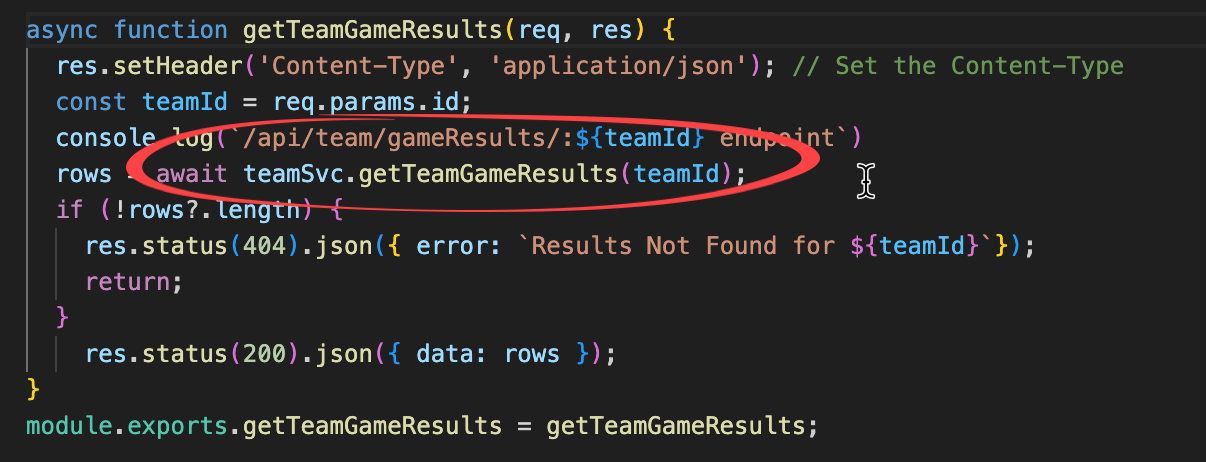
/api/**team** mean it’s the “routes/teamRoutes.js” file.



Look in /routes/teamRoutes.js file for /api/team/gameResults



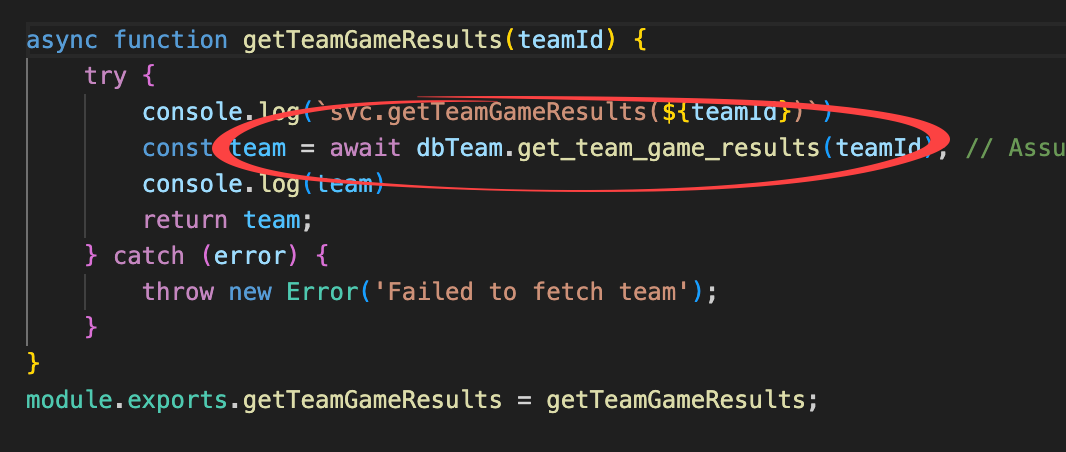
Look in teamController.js for the getTeamGameResults() function:



Look in /svc/teamService.js for getTeamGameResults() function:

(This is where “business logic” would go if there was any)

File: svc/teamService.js



Look in file /db/dbTeam.js for the get\_team\_game\_results() function

(Notice the team Id is passed in from the web page URL all the way back to the db layer: teamId)

Way back here, in the db layer, there is no business logic, no sense of the web, etc.

It is simply getting data from the database and returning it, to be processed elsewhere, e.g., turned into JSON, displayed on a web page, filtered, mapped, transformed, etc.



All these functions just return the data back until it gets to the javascript on the webpage.

The javascript does what’s needed to make it appear properly on the webpage, on a div tag or on a chart or datatable or whatever.