

Interview Questions

Take Home Exercise

The StatsAPI service (statsapi.mlb.com) is the service that powers the data behind all things Baseball, including Major League Baseball. It provides a statsapi.mlb.com/schedule endpoint that serves as the source of truth for the daily schedule of baseball games. The games are provided in a consistent order independent of the consumer. The MLB App clients have requirements to “personalize” the Scoreboard by sorting the schedule to best maximize the appeal to the end user. This means sorting the array of games for a given date to bring those games for the user’s favorite team to the top of the array returned in the response payload.

You are being asked to build a backend service that both the iOS and Android clients can use to move this sorting logic out of the native client code into a single source of truth. The service can be built in any language you choose, but you will need to provide instructions with how to run the code. We expect that this is a prototype but it should be able to be executed to the point where the response is available for inspection by the client teams.

The client will pass 2 parameters:

- The client will use the team IDs as identified in the following endpoint: <https://statsapi.mlb.com/api/v1/teams?season=2021&sportId=1> to identify the user’s favorite team (e.g. 133 is the Oakland Athletics, 134 is the Pittsburgh Pirates).
- The date, in the format *yyyy-mm-dd*

Your service will query the StatsAPI */schedule* endpoint such as <https://statsapi.mlb.com/api/v1/schedule?date=2021-09-19&sportId=1&language=en> to get the results for the given day, then transform the output to sort the array of *games* within the *dates* object such that any games where the favorite team’s *teamId* is sorted to the top. The rest of the games may remain in the order as delivered by StatsAPI. The client is expecting a JSON payload. It should be the same data format (“schema”) as StatsAPI as the clients already have models to match.

Please note that a team may be involved in more than one game on a given day (a “doubleheader” in baseball parlance). If that occurs, the games should be sorted as such:

- Both games should be listed as the first two games in the payload response returned to the client
- These games will be denoted by having the *doubleheader* field set to either of these two options (and not “N”):

- “Y” - it is a “single admission” (aka “traditional”) doubleheader where the 2nd game starts 30-45 minutes after the first, without a set start time. In this case, the *startTimeTBD* field in the *status* object will be set to **true**. This is the way to denote that it is the second game
- “S” - it is a “split admission” doubleheader where the 2nd game has a separate and distinct start time. In this case, the *gameDate* field can be used for comparison to determine which game is the earlier game.
- The sorting of these games depends on the state of the game:
 - If both games are in the future, the earlier game should be listed first.
 - If both games are in the past, the games should also be sorted chronologically
 - If either game is live, that game should be listed first
 - When the first game is live, this is still chronological
 - When the second game is live, the earlier game should appear after the later game

Any date in the 2021 and 2022 calendar years may be used to evaluate your service. Key dates to consider:

- 04/01/2021
- 07/13/2021
- 09/11/2021
- 11/02/2021
- 03/31/2022
- 07/21/2022

You can submit your assignment by sharing a link to a publicly available repository or zipping it up and sending it via email or sharing a link to the zip file hosted in the cloud. Please make sure you provide directions for how the evaluators of your work should run it.

Expectations

- We know you are busy and in most cases, have a full time job already. We tried to narrow the scope of the project down to reduce the number of hours it should take to complete. We hope that it takes no more than 4-5 hours to complete
- Please try to return the project within 1-2 weeks. If you need an extension, email me directly.