# **~~Build a Record Collection~~**

~~You are creating a function that aids in the maintenance of a musical album collection. The collection is organized as an object that contains multiple albums which are also objects. Each album is represented in the collection with a unique id as the property name. Within each album object, there are various properties describing information about the album. Not all albums have complete information.~~

~~The updateRecords function takes 4 arguments represented by the following function parameters:~~

* ~~records - an object containing several individual albums~~
* ~~id - a number representing a specific album in the records object~~
* ~~prop - a string representing the name of the album’s property to update~~
* ~~value - a string containing the information used to update the album’s property~~

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| updateRecords(records, id, prop, value) | | | | |
| records | id | prop | value | action |
| 2. | true | true | === “” | delete prop |
| 3. | true | != “tracks” | != “” | prop = value |
| 4. | Id[prop] === false | “tracks” | != “” | tracks[] === tracks[value] |
| 5. | Id[prop] === true | “tracks” != “” | “value != “” | Tracks[1,2,…,”value”] |

****User Stories:****

1. ~~Your function must always return the entire records object.~~
2. ~~If value is an empty string, delete the given prop property from the album.~~
3. ~~If prop isn't tracks and value isn't an empty string, assign the value to that album's prop.~~
4. ~~If prop is tracks and value isn't an empty string, but the album doesn't have a tracks property, create an empty array and add value to it.~~
5. ~~If prop is tracks and value isn't an empty string, add value to the end of the album's existing tracks array.~~

****Note:**** A copy of the recordCollection object is used for the tests. Your function should not directly refer to the recordCollection object, only the function parameter.

Failed:3. After updateRecords(recordCollection, 5439, "tracks", "Take a Chance on Me"), tracks should have the string Take a Chance on Me as the last and only element.

Passed:4. After updateRecords(recordCollection, 2548, "artist", ""), artist should not be set

Failed:5. After updateRecords(recordCollection, 1245, "tracks", "Addicted to Love"), tracks should have the string Addicted to Love as the last element.

Failed:6. After updateRecords(recordCollection, 2468, "tracks", "Free"), tracks should have the string 1999 as the first element.

Passed:7. After updateRecords(recordCollection, 2548, "tracks", ""), tracks should not be set

Passed:8. After updateRecords(recordCollection, 1245, "albumTitle", "Riptide"), albumTitle should be the string Riptide

// running tests

3. After updateRecords(recordCollection, 5439, "tracks", "Take a Chance on Me"), tracks should have the string Take a Chance on Me as the last and only element.

5. After updateRecords(recordCollection, 1245, "tracks", "Addicted to Love"), tracks should have the string Addicted to Love as the last element.

6. After updateRecords(recordCollection, 2468, "tracks", "Free"), tracks should have the string 1999 as the first element.

// tests completed