

from: <https://newline.theironyard.com/cohorts/15/courses/9/projects/74>

## Week 4 - iTunes Music Search

In this project you'll use everything you've learned so far to create a simple application.

### iTunes Music Search

---

As your final project for this portion of the course, let's take stock of all you have learned and build a real application that is useful and you can share with friends. For this app, we will be using iTunes API to pull data from.

You'll use this data to pull songs based on a search that your user performs. Here is an idea of what the end result should look like, though you can have fun with the design.

Here are the steps you'll need to take in order to complete this project.

1. Build a simple form that has an `<input>` where a user can fill in their favorite band, like "Backstreet Boys", and it will return a handful of songs by them or with their name in it.
2. When the user types in a band name and presses the **submit** button, you should then make the search request. You can trap this with an event listener.
3. Once you have the search term, you should `fetch` the specific endpoint and use the results to display a listing of songs related to the search term.

4. Then to add some features, you should set it up so when a user clicks on one of the songs, it should then play in an `<audio>` tag that you've also added to the page (see the mockup).

## Hints & Tips

---

There will be some new concepts you'll need to work through on this project, so feel free to ask for assistance along the way.

- API Documents
  - You've likely never worked with a robust API before, so you'll need to take time to read through the documentation before getting started. Everything you'll need to know is in there.
- Form Submission
  - You should use the `submit` event listener method on your form. This will trap when the button was pressed, thus allowing you to write a handler function.
  - You'll also need to get the current value after the submission has happened.
- Fetching Data
  - You'll need to take the value from above and use that to build out your URL to send to iTunes.
- Playing a Song
  - You'll need to research the `<audio>` tag for this part - [docs here](#)
  - **Hint:** *You'll need to dynamically change the `src` value*

The biggest gotcha will likely be getting the song to play. Since your JavaScript has already run, if you added an `click` event listener to your page, but you add the content after the fact it won't register. You need to get creative here, but putting your `click`

around the entire results section, and then making sure to get the correct item clicked on. Here is a great [article on Event Delegation](#)

## Mockup & Starter Files

---

Note, the below wireframe is just an idea of the structure. You can follow it perfectly or do something completely different. As long as the requirements above are there, then you win.

- [Music App Wireframe](#)

[Forked starter files.](#)

## Bonus: Deployment

---

Ok, so you've built a really cool web application. Congrats! Now you'd like a way to share it with your friends. There are *many* deployment strategies you can take and your instructor might have others they want to share, but let's talk through a really simple one that you can do without adding any other tools.

Many of you might not know this, but GitHub has a deployment tool built in that's free to use. It's called GitHub Pages, and it's pretty easy to use. Here is how:

1. Ensure your `index.html` file is in the root of your project
2. Create a branch called `gh-pages`
  - `git branch gh-pages`
3. Push the branch
  - `git push origin gh-pages`
4. That's it!

You can now view your live site at

`http://<username>.github.io/<repository>` (just replace username and repository).

One thing to note though, is if you later change your master branch, you need to merge the update into your `gh-pages` branch or it won't show up. It also can take up to an hour to update on their servers. Because of this, I would suggest not doing this step until you are done with your project and ready to deploy it.