

Sprint challenge for Sprint 11 of Lambda's iOS course

Edit

[Manage topics](#)

2 commits

1 branch

0 releases

2 contributors

Branch: master

New pull request

Create new file


Upload files

Find file

Clone or download

This branch is even with LambdaSchool:master.

[Pull request](#) [Compare](#)

 **SpencerCurtis** Create README.md

Latest commit 692a8cb on Oct 5, 2018

<a href="#">.gitignore</a>	Initial commit	8 months ago
<a href="#">README.md</a>	Create README.md	5 months ago

README.md

# Lyric Finder

## Instructions

Please read this entire README to make sure you understand what is expected of you before you begin.

This sprint challenge is designed to ensure that you are competent with the concepts taught throughout Sprint 11, Objective-C Part 1.

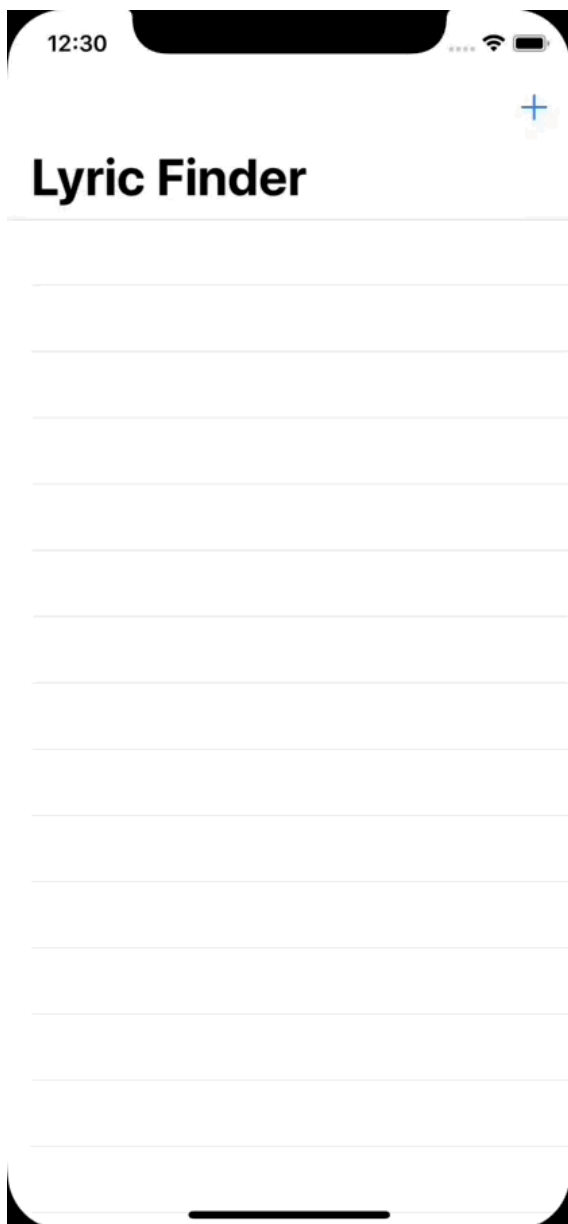
Begin by forking this repository. Clone your forked repository to your machine. There is no starter project. Create a project for this challenge in this repository and commit as appropriate while you work. Push your final project to GitHub, then create a pull request back to this original repository.

**You will have 3 hours to complete this sprint challenge**

If you have any questions about the project requirements or expectations, ask your PM or instructor. Good luck!

## Screen Recording

Please view the screen recordings so you will know what your finished project should look like:



(The gif is fairly large in size. It may take a few seconds for it to appear)

## Before you begin

The API you will be using is [here](#). If you do not have one already, you will first need to sign up for a Mashape account. From there, you will be able to use/test the "Artist.search" part of the API to see some example JSON.

We have not gone over HTTP Headers before in class. In order to use this API, you must use your API key (or, "Mashape key" in this API) in a header, not a URL parameter like you are used to. Use the following screenshot and code snippet to set your API key as a header in your request:

## MATCHER

## Matcher.lyrics

Get the lyrics for a song starting by title and artist

Your API key is here

## URL PARAMETERS

q.artist STRING	<input type="text" value="coldplay"/>
The artist name	
q.track STRING	<input type="text" value="paradise"/>
The song title	

## REQUEST HEADERS

X-Mashape-Key	<input type="text" value="Default Application"/>
The Mashape application you want to use for this session.	

## ENDPOINT DEFINITION

**GET** <https://musixmatchcom-musixmatch.p.mashape.com/wsr/1.1/matcher.lyrics.get>

## REQUEST EXAMPLE

```
curl --get --insecure 'https://musixmatchcom-musixmatch.p.mashape.com/wsr/1.1/matcher.lyrics.get' \
-H 'X-Mashape-Key: [REDACTED]' \
-H 'Accept: application/json'
```

## RESPONSE BODY

**200 / JSON** Lyrics  
Lyrics model

```
NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:requestURL];
```

```
[request setValue:@"yourAPIKeyHere" forHTTPHeaderField:@"X-Mashape-Key"];
```

## Requirements

The goal of this sprint challenge is to create an app that fetches a song's lyrics.

The requirements for this project are as follows:

2. A model object that represents a song. It should have the following properties:
  - o A title.
  - o An artist.
  - o The lyrics.
  - o A rating. **This must be a primitive.**
3. A category called `NSJSONSerialization`. In this category add:
  - o An initializer to initialize your model object from a dictionary.
  - o A function that returns your model object in dictionary form.
4. Persist your model object using `NSFileManager` to save your model objects to a file on the application's document directory. **Note:** As you do not have access to `Codable` use `NSJSONSerialization` to help turn your model objects into `NSData`.
5. A table view controller that displays a list of saved songs with their lyrics.
6. A detail view controller that allows the user to search for new song lyrics, and also view saved song lyrics.

## Go Further

- Persist your model objects using Core Data instead of saving them to a file.