




# Work on project. Stage 2/5: Rocking OAuth

Project: [Music Advisor](#)

## Rocking OAuth

 Hard  18 minutes  1099 users solved this stage. Latest completion was **about 15 hours ago**.

### §1. Description

You have written a simulation of the final application, and now we will make it fully functional, step by step.

### §2. Working with OAuth

OAuth means a protocol for authorization. When you are visiting a website that allows you to register, you often have the option of signing in with Google, Twitter, Facebook, or another service. All these services use OAuth to provide user information to third-party applications.

The main entity of OAuth is `access_token`, the secret code that should be sent with an HTTP request to API, so that the service is sure that you have enough rights to get information from API. So, before an application can receive `access_token`, the user should confirm access to that application.

In our case, using the Spotify API, it will look like this:

```
https://accounts.spotify.com/authorize?client_id=YOURCLIENT&redirect_uri=https://www.example.com&response_type=code
```

1. The user follows this link and confirms access.
2. Then the user is redirected to `redirect_uri` with the response in the URL  
( `http://www.example.com?code=7angkw2DAsdfkQ` ). (Your own server waiting for `code`.)
3. You use this `code` to get `access_token` and make requests to an API.

You can find more information in the [authorization guide](#) (look at *Authorization Code Flow* section).

### §3. Objectives

So, in this stage, you must go to the [Spotify Web site for developers](#) and create your application.

To create an application, you should select `Dashboard` tab on the site, log in to Spotify, and click the button `Create an App`.

Add one more command to your program: an `auth` command that will print the auth link and allow us to use another command (**no real authorization required**, just print a workable link and make some Boolean field true). Don't forget to put your client id in this link.

Make commands unavailable if user access for your program is not confirmed (if they did not call the `auth` command).

Remember that you should add your `redirect_uri` in the settings of your application.

### §4. Example

Below is an output example of the described program. Try to output all cases like in the example.

The greater-than symbol followed by a space ( `>`  ) represents the user input. Note that it's not part of the input.

```
> new
Please, provide access for application.
> featured
Please, provide access for application.
> auth
https://accounts.spotify.com/authorize?client_id=a19ee7dbfda443b2a8150c9101bfd645&redirect_uri=http://localhost:8080&response_type=code
```

```
---SUCCESS---
> new
---NEW RELEASES---
Mountains [Sia, Diplo, Labrinth]
Runaway [Lil Peep]
The Greatest Show [Panic! At The Disco]
All Out Life [Slipknot]
> exit
---GOODBYE!---
```

[Report a typo](#)[⚡ See hint](#)[↩ Write a program](#)[Code Editor](#)[IDE](#)

### CONNECTION STATUS

× IDE is not installed or is not running (not opened)

Please open a JetBrains IDE [or install and configure it from this page ↗](#)

[Solve in IDE](#)

build.gradle

```
apply plugin: 'java'
apply plugin: 'application'

group 'advisor'
version '1.0-SNAPSHOT'

repositories {
    mavenCentral()
}

dependencies {
    implementation 'com.google.code.gson:gson:+'
}

jar {
    manifest {
        attributes 'Main-Class' : 'advisor.Main'
    }
    from {
        configurations.compile.collect { it.isDirectory() ? it : zipTree(it) }
    }
}
```

src/advisor/Main.java

```
package advisor;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        MusicAdvisor musicAdvisor = new MusicAdvisor();
        musicAdvisor.start();
    }
}
```