Spring

Start.spring.io 🡪 Gives skeleton of maven project written in java language using framework SpringBoot

IntelliJ IDEA is a JAVA sandbox built for robust code analysis, debugging tools, and seamless integration with various development tools platforms. IntelliJ IDEA is highly regarded for its out-of-the-box functionality and ease of use, making it a popular choice among developers.

Steps to work with Spring:

Goto start.spring.io >> Language=java;Project=maven;Spring boot=lowest version;Metadata=projectyehpahadhai;Package=jar;Java=21 >> Dependencies=Spring web,Spring boot actuator >> Generate >> move skeleton to C:/microservices/makemytrip

Then

Goto Intellij IDEA >> New Project from existing sources >> C:/microservices/makemytrip >> C:\microservices\makemytrip\src\main\java\com\example\makemytrip (creates source code.java) >> Add java class=bus, hotels, flights

Then on addition every feature

* mvn clean compile …………………………………………… creates class file
* mvn clean package …………………………………………… creates jar file
* mvn spring-boot:run …………………………………………… starts app
* <http://localhost:8080/flights> …………………………………………… displays flights
* <http://localhost:8080/hotels> ………………………………………..... displays hotels
* <http://localhost:8080/bus> …………………………………………... displays bus

Then prepare for repo upload from local to remote

* git pull origin master
* git add –all
* git commit –m “coderefactor”
* git push origin master
* 2nd time & rest:
* git add –all
* git commit –m “adding feature”
* git push origin master

Error handling

**Git Push vs -u**

The primary difference between git push and git push -u (or git push --set-upstream) lies in how they handle the tracking of remote branches.

git push is used to send the changes from your local repository to the remote repository. It does not set up any tracking information for the branch. This means that if you want to push changes again, you will need to specify the remote and branch again, or set up tracking manually.

git push -u or git push --set-upstream, on the other hand, **sets up tracking information for the branch**. This means that after you use this command, you can simply use git push in the future to push changes to the same remote branch without needing to specify the remote and branch again.

For example, if you use git push -u origin main, it will push your local main branch to the remote main branch on origin and set up tracking information so that subsequent git push commands will automatically push to the same remote branch.

This tracking setup simplifies future interactions with the remote repository, making it easier to push and pull changes.