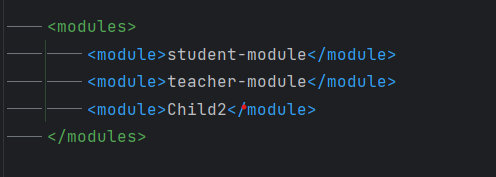
Add Library On Existing project

Our purpose is to include a library module to an existing project and use the feature of that library to our existing project. To add the library module or any other existing projects module we have to follow below steps but based on the project requirements there might be a slight variation on adding the configuration.

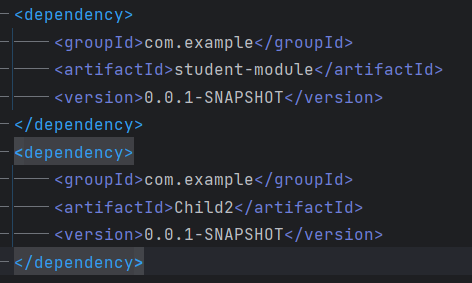
## Steps

### Monolithic Project

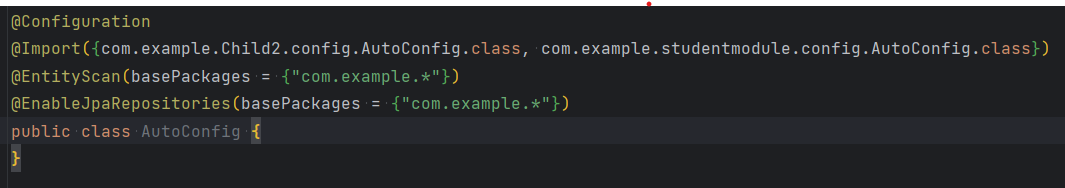
1. Create a parent project.
2. Add the Existing project
3. On parent modules pom.xml / settings.gradle add the module
4. Add the library module or module and add module on parent pom.xml file



1. On Existing projects module add the dependency of library module
2. Include groupId, artifactId and version of the modules.



1. If the library module contains a JPA repository or Entity and we need to make an instance of components then we need to add this information to the configuration.
   1. For JPARepository we need to add @EnableJpaRepositories(basePackages = {"com.example.\*"})
   2. For Entity we need to add @EntityScan(basePackages = {"com.example.\*"})
   3. If we want to include additional class of the modules then we can also import the classes: @Import({com.example.Child2.config.AutoConfig.class, com.example.studentmodule.config.AutoConfig.class})



1. Then run the project

### Modular Based Project

1. Include the module on the parent pom.xml file.
2. On Existing projects module add the dependency of library module
3. If the library module contains a JPA repository or Entity and we need to make an instance of components then we need to add this information to the configuration.
   1. For JPARepository we need to add @EnableJpaRepositories(basePackages = {"com.example.\*"})
   2. For Entity we need to add @EntityScan(basePackages = {"com.example.\*"})
   3. If we want to include additional class of the modules then we can also import the classes: @Import({com.example.Child2.config.AutoConfig.class, com.example.studentmodule.config.AutoConfig.class})
4. Run the project.

## Dependency

* While adding the library or making the multimodule project, we need to take our common dependencies to the parent module to make it more flexible.
* Modules should be designed as independent as possible.
* Before running the project we need to recheck the versions if it’s compatible or not.