SWAGGER

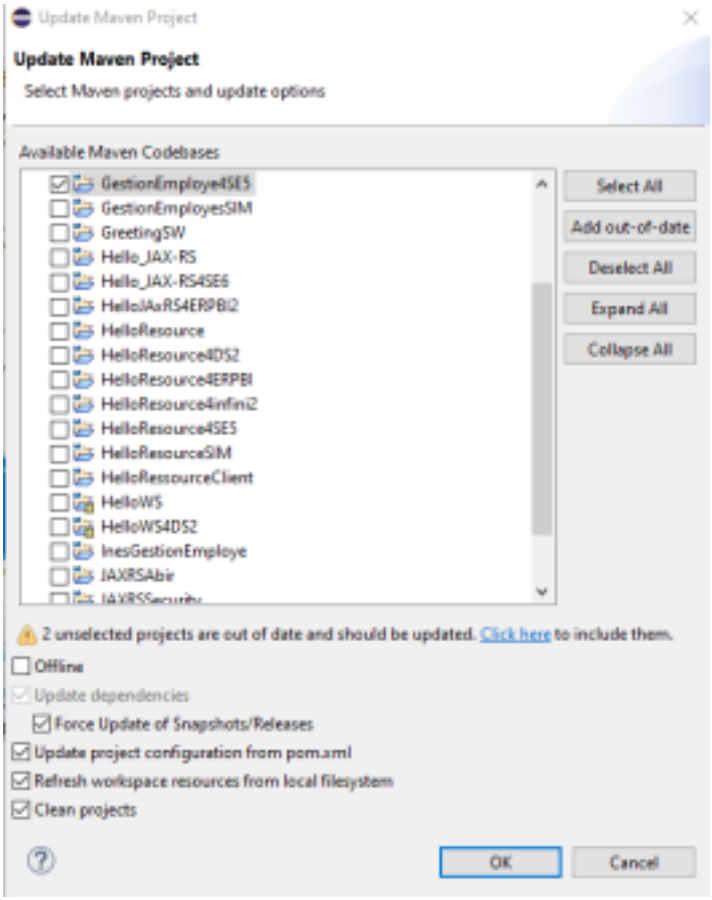
Étapes ajout Swagger a un projet REST:

1. Ajout de dépendances au niveau de “pom.xml”:

```
<dependency>
    <groupId>io.swagger</groupId>
    <artifactId>swagger-jersey2-jaxrs</artifactId>
    <version>1.5.0</version>
</dependency>
```

2. Mettre à jour le projet :

   Click droit sur le projet → Maven → Update project (force update on snapshot)

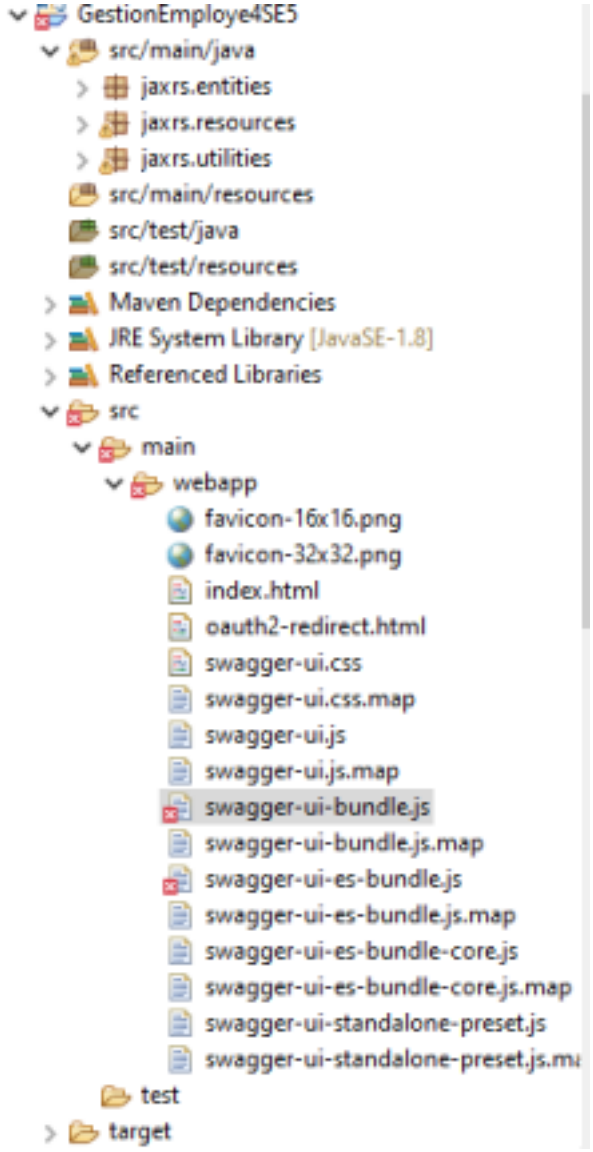


3. Télécharger Swagguer-ui

Lien de téléchargement :[Swagger-ui]

4. Copier Swagger-ui vers le projet Rest :

Copier le contenu de **swagger-ui-3.52.0.zip\swagger-ui-3.52.0\dist** vers PROJETREST → src → main→ webapp



5. Changer l'url au niveau de intex.html pour pointer sur le fichier **swagger.json**:

   http://localhost:8085/**URLDEBASE**/swagger.json

```
window.onload = function() {
  // Begin Swagger UI call region
  const ui = SwaggerUIBundle({
    url: "http://localhost:8085/GestionEmploye4SE5/gestionEmploye/swagger.json",
    dom_id: '#swagger-ui',
    deepLinking: true,
    presets: [
      SwaggerUIBundle.presets.apis,
      SwaggerUIStandalonePreset
    ],
    plugins: [
      SwaggerUIBundle.plugins.DownloadUrl
    ],
    layout: "StandaloneLayout"
  });
  // End Swagger UI call region

  window.ui = ui;
};
```

6. Ajouter les fournisseurs de swagger-core au processus de configuration:

```
@ApplicationPath(value = "gestionEmploye")
public class RestActivator extends Application {

    |


    @Override
       public Set<Class<?>> getClasses() {
           Set<Class<?>> resources = new HashSet();

           resources.add(EmployeSW.class);
           //resources.add(SecondResource.class);


           resources.add(io.swagger.jaxrs.listing.ApiListingResource.class);
           resources.add(io.swagger.jaxrs.listing.SwaggerSerializers.class);

           return resources;
       }

}
```

Le code à ajouter :

```
@Override
public Set<Class<?>> getClasses() {
    Set<Class<?>> resources = new HashSet();
```

```
        resources.add(EmployeSW.class);
        //resources.add(SecondResource.class);


        resources.add(io.swagger.jaxrs.listing.ApiListingResource.class);
        resources.add(io.swagger.jaxrs.listing.SwaggerSerializers.class);

        return resources;
    }
```

PS: Remplacer **EmployeSW.class** par la classe qui contient le web service

7. Configurer Swagger au niveau du constructeur de la classe d'application (Activation rest)

```
@ApplicationPath(value = "gestionEmploye")
public class RestActivator extends Application {

    public RestActivator() {
        BeanConfig beanConfig = new BeanConfig();
        beanConfig.setVersion("1.0.2");
        beanConfig.setSchemes(new String[]{"http"});
        beanConfig.setHost("localhost:8085");
        beanConfig.setBasePath("GestionEmploye4SE5/gestionEmploye");
        beanConfig.setResourcePackage("io.swagger.resources");
        beanConfig.setResourcePackage("jaxrs.resources");
        beanConfig.setScan(true);
    }


    @Override
        public Set<Class<?>> getClasses() {
            Set<Class<?>> resources = new HashSet();

            resources.add(EmployeSW.class);
            //resources.add(SecondResource.class);


            resources.add(io.swagger.jaxrs.listing.ApiListingResource.class);
            resources.add(io.swagger.jaxrs.listing.SwaggerSerializers.class);

            return resources;
        }

}
```

Le code a ajouter :

```
public RestActivator() {
```

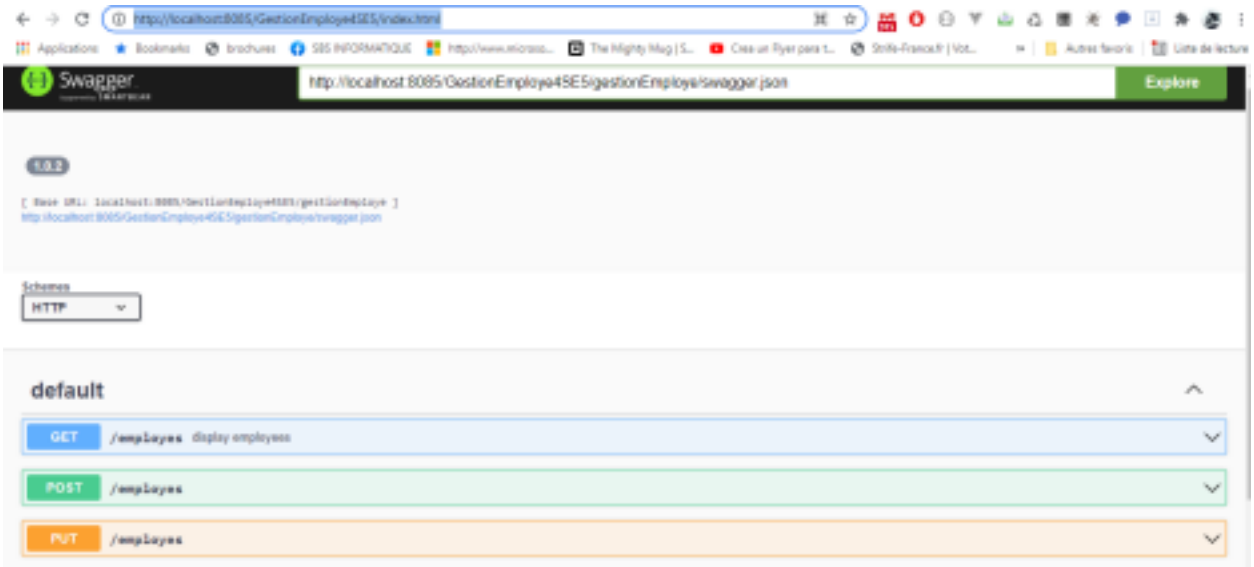
```
        BeanConfig beanConfig = new BeanConfig();
        beanConfig.setVersion("1.0.2");
        beanConfig.setSchemes(new String[]{"http"});
        beanConfig.setHost("localhost:8085");
        beanConfig.setBasePath("GestionEmploye4SE5/gestionEmploye");
        beanConfig.setResourcePackage("io.swagger.resources");
        beanConfig.setResourcePackage("jaxrs.resources");
        beanConfig.setScan(true);
    }
```

Ps : Changer **GestionEmploye4SE5/gestionEmploye** par l'url de base de l'application en question

Changer **jaxrs.resources** par le package qui contient les ressources du projet

8. Tester Swagger :

Pour tester Swagger il suffit de lancer le projet puis lancer le fichier **index.html (**http://localhost:**PORT/NomDuProjet**/index.html**)**



9. Dernière étape :

Ajouter les Annotation adéquates (voir cours)

```java
@Path(value = "employes")
@Api
public class EmployeSW {


    public static  List<Employe> listemploye=new ArrayList<Employe>();



    @POST
    @Consumes(MediaType.APPLICATION_XML)
    @Produces(MediaType.APPLICATION_JSON)
    public Response addEmployee(Employe e) {
        listemploye.add(e);
        return Response.status(Status.CREATED).entity(e).build();
    }

    @ApiOperation(value = "display employees", produces = MediaType.APPLICATION_XML )
    @GET
    @Produces(MediaType.APPLICATION_XML)
    public List<Employe> displayEmployeesList() {
            return listemploye;

        }
```