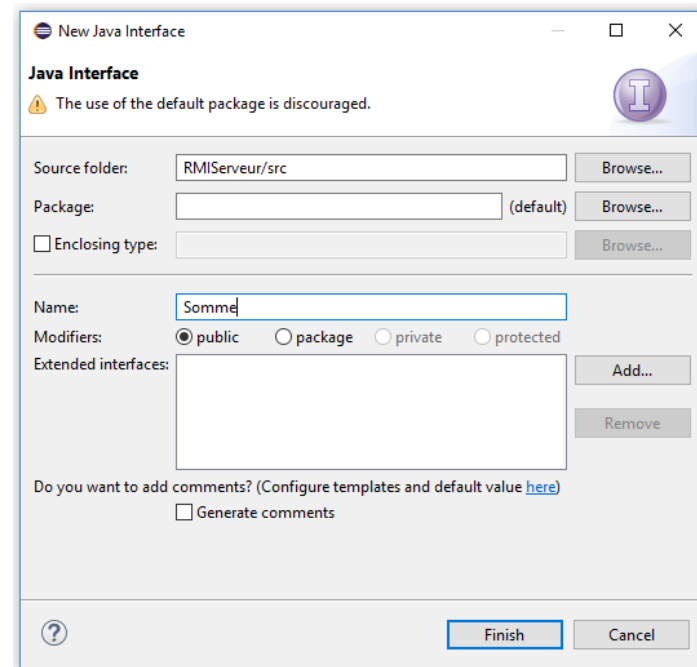


Tutoriel RMI avec Eclipse

Pas à pas

Du côté serveur

- Créer un nouveau projet **RMIServeur**
- Créer une nouvelle interface **Somme**



- Ajoutez le code suivant


```
Somme.java ✕
1  import java.rmi.*;
2
3  public interface Somme extends Remote {
4      public int add(int a,int b) throws RemoteException;
5  }
```

Du côté serveur

- Créez une classe **SommeImpl**
- Avec le code suivant


```
SommeImpl.java
```

```
1+ import java.rmi.*;
3
4
5 public class SommeImpl extends UnicastRemoteObject implements Somme {
6
7     public SommeImpl () throws RemoteException { }
8
9- public int add(int a, int b) throws RemoteException {
10     int result=a+b;
11     return result;
12 }
13 }
```



New Java Class

Java Class

 The use of the default package is discouraged.

Source folder: RMISeurveur/src

Package:

Enclosing type:

Name:

Modifiers: public package private protected
 abstract final static

Superclass:

Interfaces:

Which method stubs would you like to create?

public static void main(String[] args)

Constructors from superclass

Inherited abstract methods

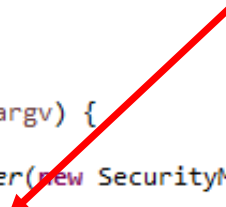
Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

Du côté serveur

- Créer une nouvelle classe **ServeurSomme** avec le code suivant

```
1+  import java.rmi.*;
4
5  public class ServeurSomme {
6
7
8-  public static void main (String[] argv) {
9      try {
10         System.setSecurityManager(new SecurityManager());
11
12         SommeImpl stub = new SommeImpl();
13         Registry registry = LocateRegistry.getRegistry (1099);
14         registry.rebind(" Somme ", stub );
15
16         System.out.println("serveur de sommes prêt");
17     }catch (Exception e) {
18         System.out.println("erreur serveur de sommes " + e);
19     }
20 }
21 }
22
```

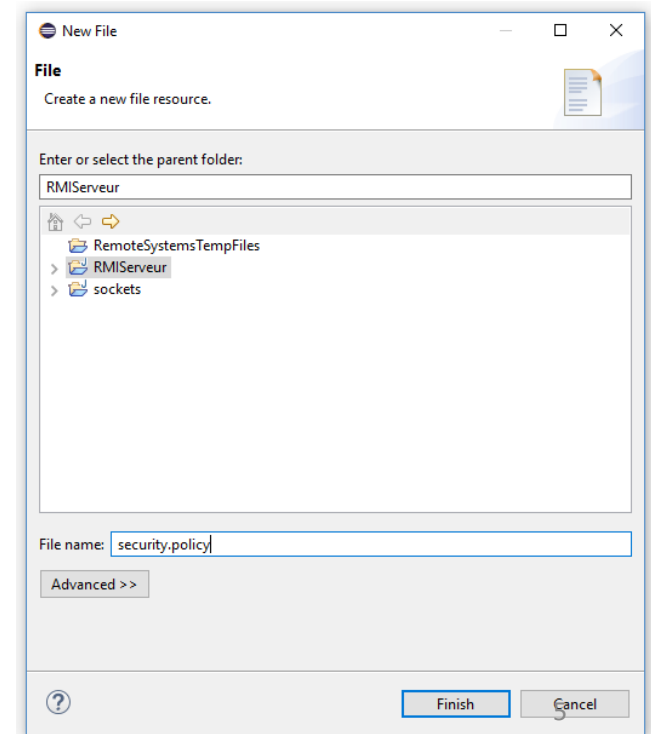


Du côté serveur

- Nous allons créer une politique de sécurité.
- Créer un nouveau *fichier* dans le projet **RMIServeur** et appelez le *security.policy*
- Ajoutez une politique de sécurité permissive

```
security.policy ✕  
1 grant {  
2     permission java.security.AllPermission;  
3 };
```

- Compiler le tout (il y aura des erreurs)

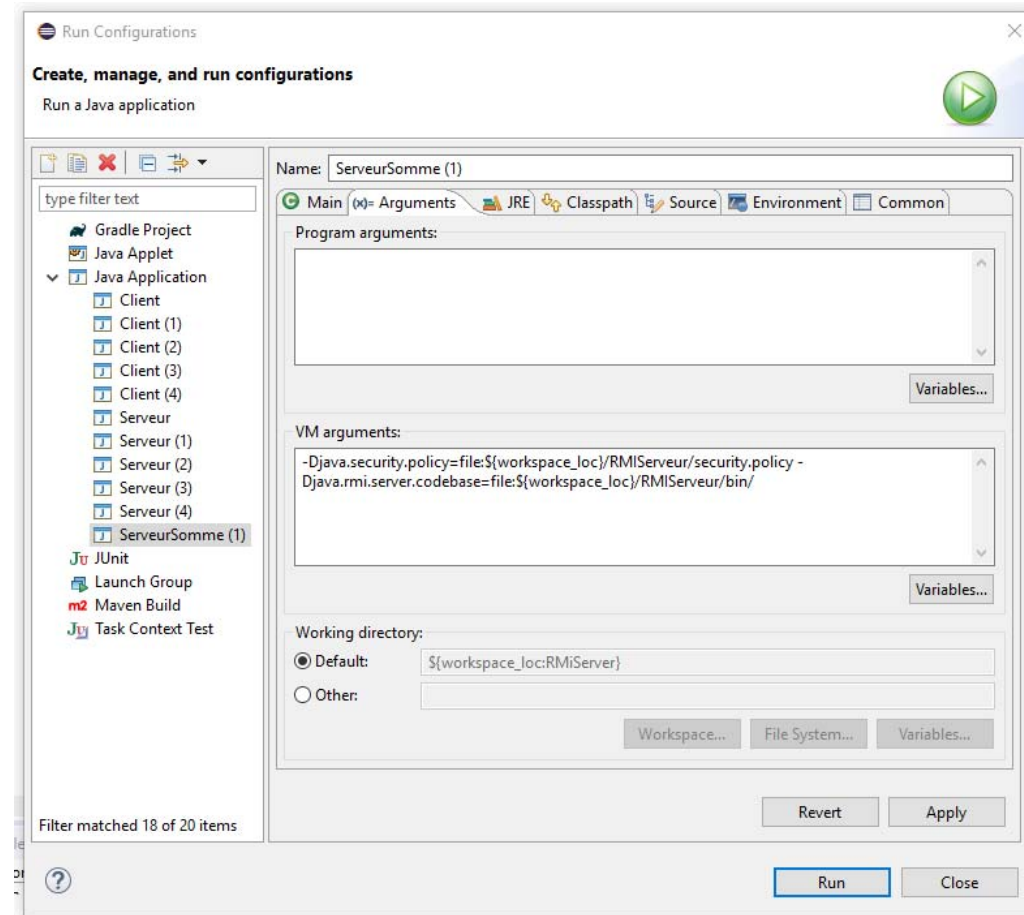


Lancer le serveur de noms

- Ouvrez une console,
- Naviguez jusqu'au dossier /bin de votre projet **RMIServeur**
- Lancez le serveur de noms

Configuration de la JVM

- Sélectionner le projet
- Cliquer **Run -> Run configurations**
- Choisir la classe **ServeurSomme** et cliquer sur **Arguments**
- Ajouter les arguments suivants pour la JVM

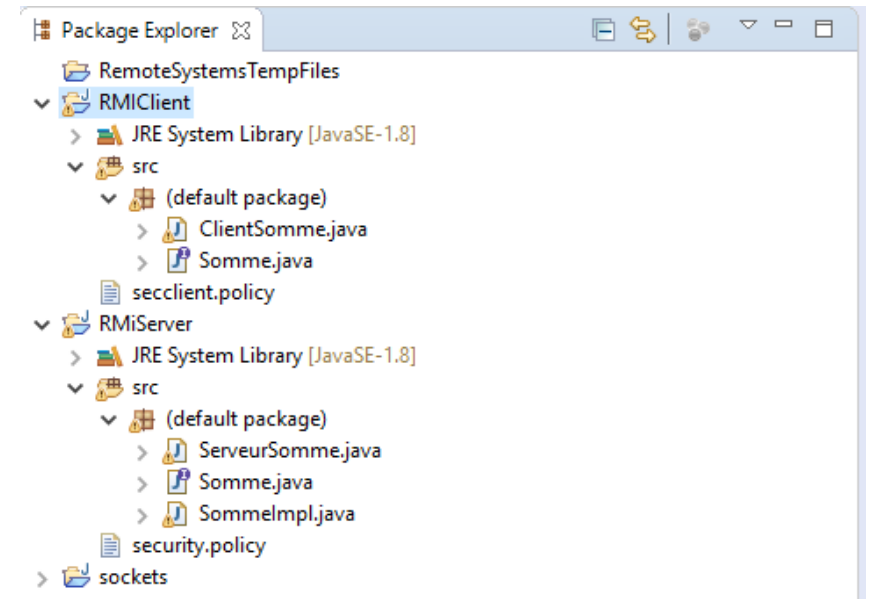


```
-Djava.security.policy=file:${workspace_loc}/RMIServeur/security.policy -  
Djava.rmi.server.codebase=file:${workspace_loc}/RMIServeur/bin/
```

- Cliquer **Apply -> Run**

Du côté client

- Il va falloir créer un nouveau projet **RMIClient**
- Le client doit connaître l'interface du service proposé par le serveur .. En général on va la télécharger de quelque part. Mais ici, on va copier le code **Somme.java** du serveur
- Et aussi, notre client va mettre en place une politique de sécurité permissive. Nous allons créer un fichier avec la 8politique *seccclient.policy*



Du côté client

- Et créer la classe **ClientSomme.java** comme ceci

```
ClientSomme.java Somme.java secclient.policy
1 import java.rmi.*;
2 import java.rmi.registry.LocateRegistry ;
3 import java.rmi.registry.Registry ;
4
5 public class ClientSomme {
6     public static void main (String[] args) {
7
8         try {
9             System.setSecurityManager(new SecurityManager());
10
11             Registry registry = LocateRegistry.getRegistry (1099);
12             Somme objet = (Somme) registry.lookup (" Somme ");
13
14             int result=objet.add(9,10);
15
16             System.out.println("Le résultat est :"+result);
17
18         }catch (Exception e) {
19             System.out.println("Exception de ClientSomme : " + e);
20         }
21     }
22 }
```

Configuration de la JVM

- Configurer la JVM pour indiquer la politique de sécurité, en faisant click droit sur le projet RMIClient et en faisant **Run as -> Run Configurations**

- Indiquer les configurations suivantes

```
-Djava.security.policy=file:${workspace_loc}/RMIClient/secclient.policy -  
Djava.rmi.server.codebase=file:${workspace_loc}/RMIClient/bin/
```

- Exécutez le client en faisant **Apply -> Run**

