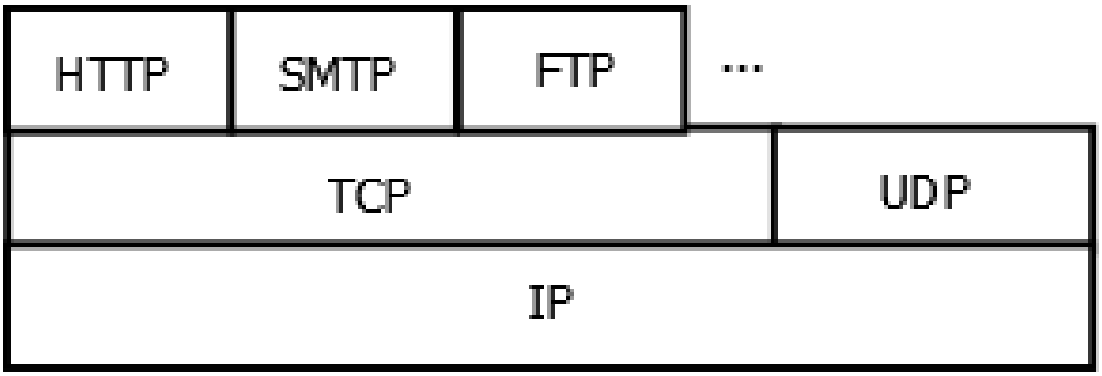
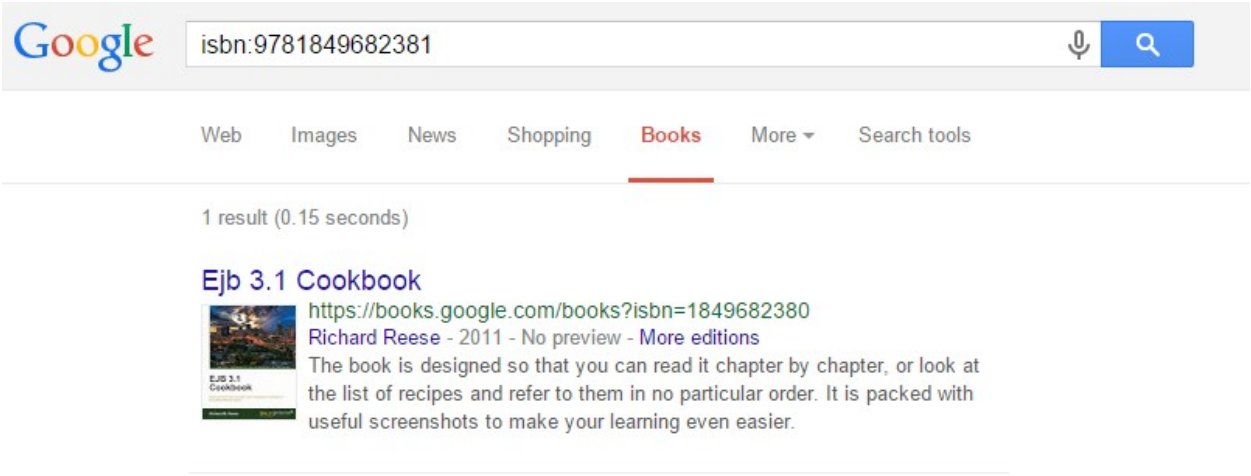
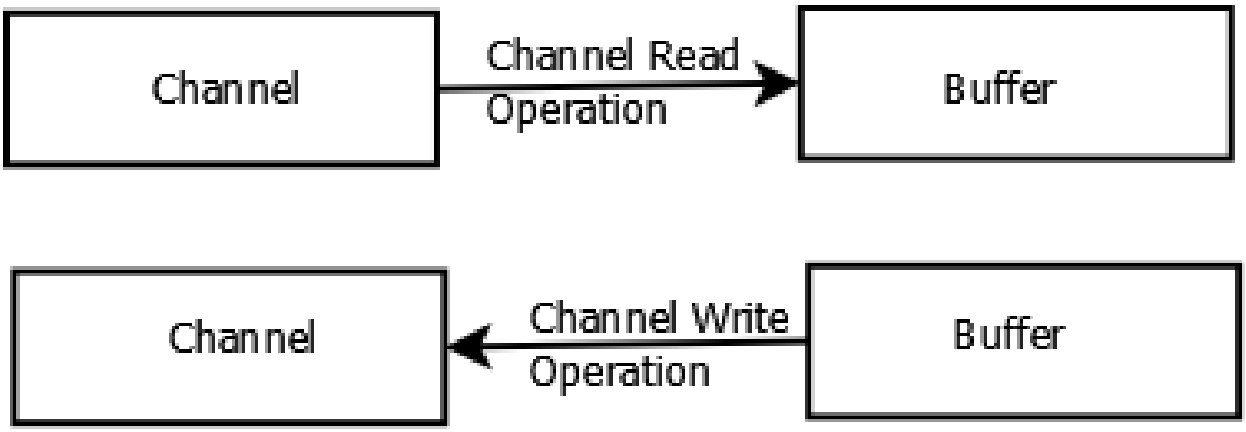
Chapter 2: Network Addressing

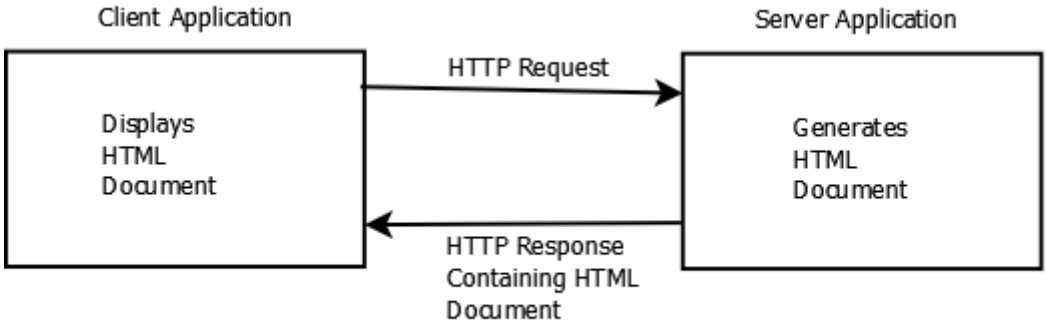


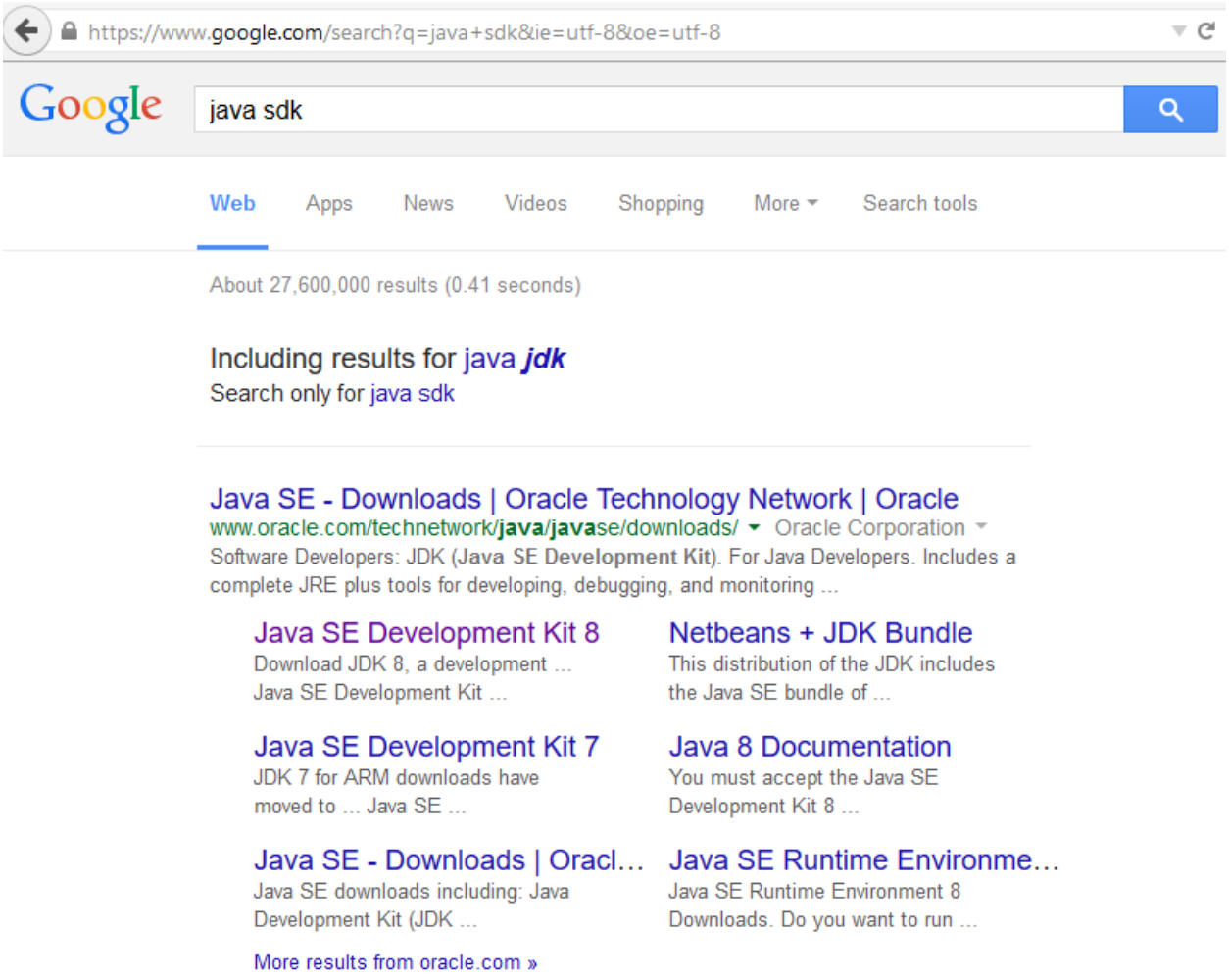


Chapter 3: NIO Support for Networking

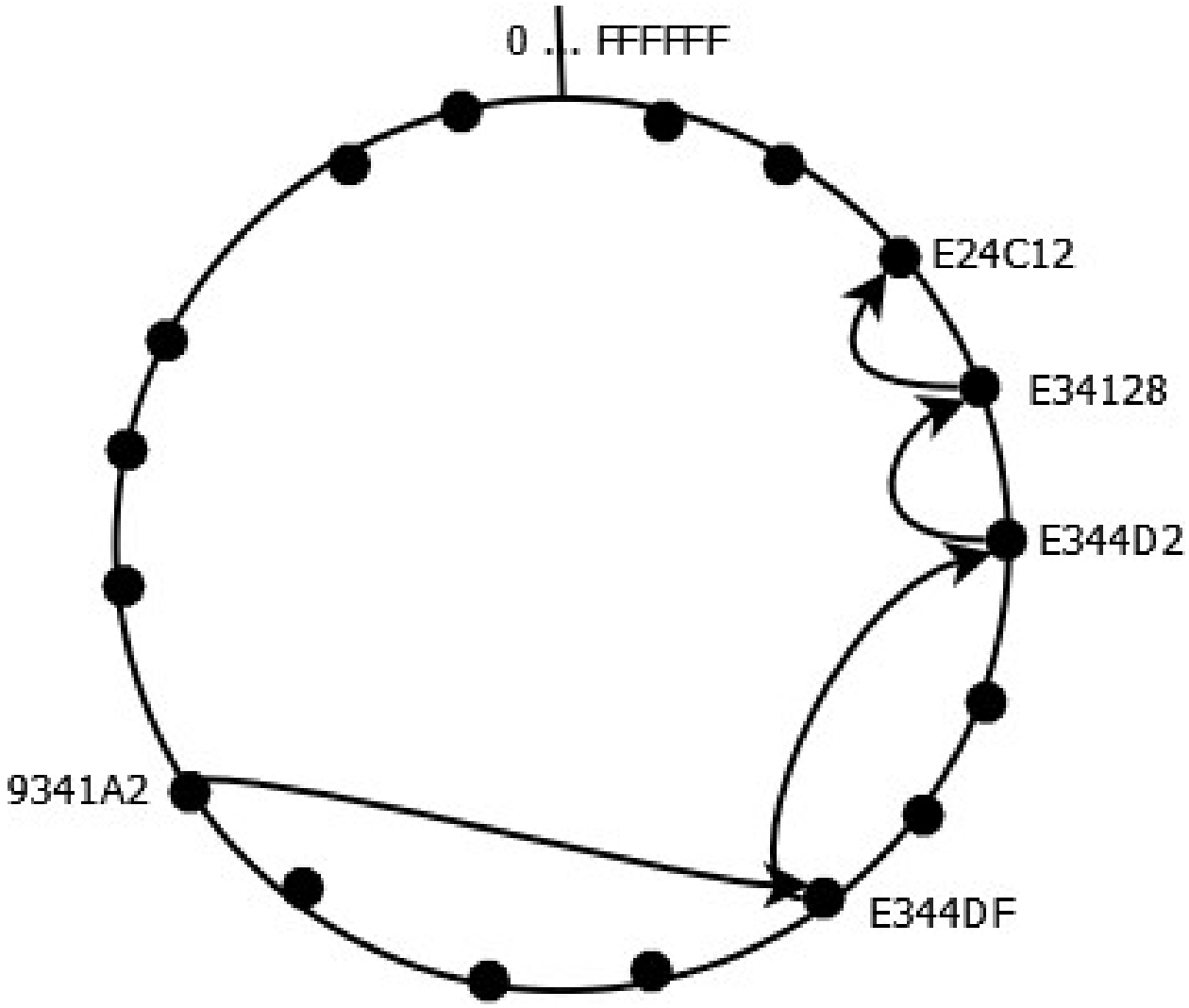


Chapter 4: Client/Server Development

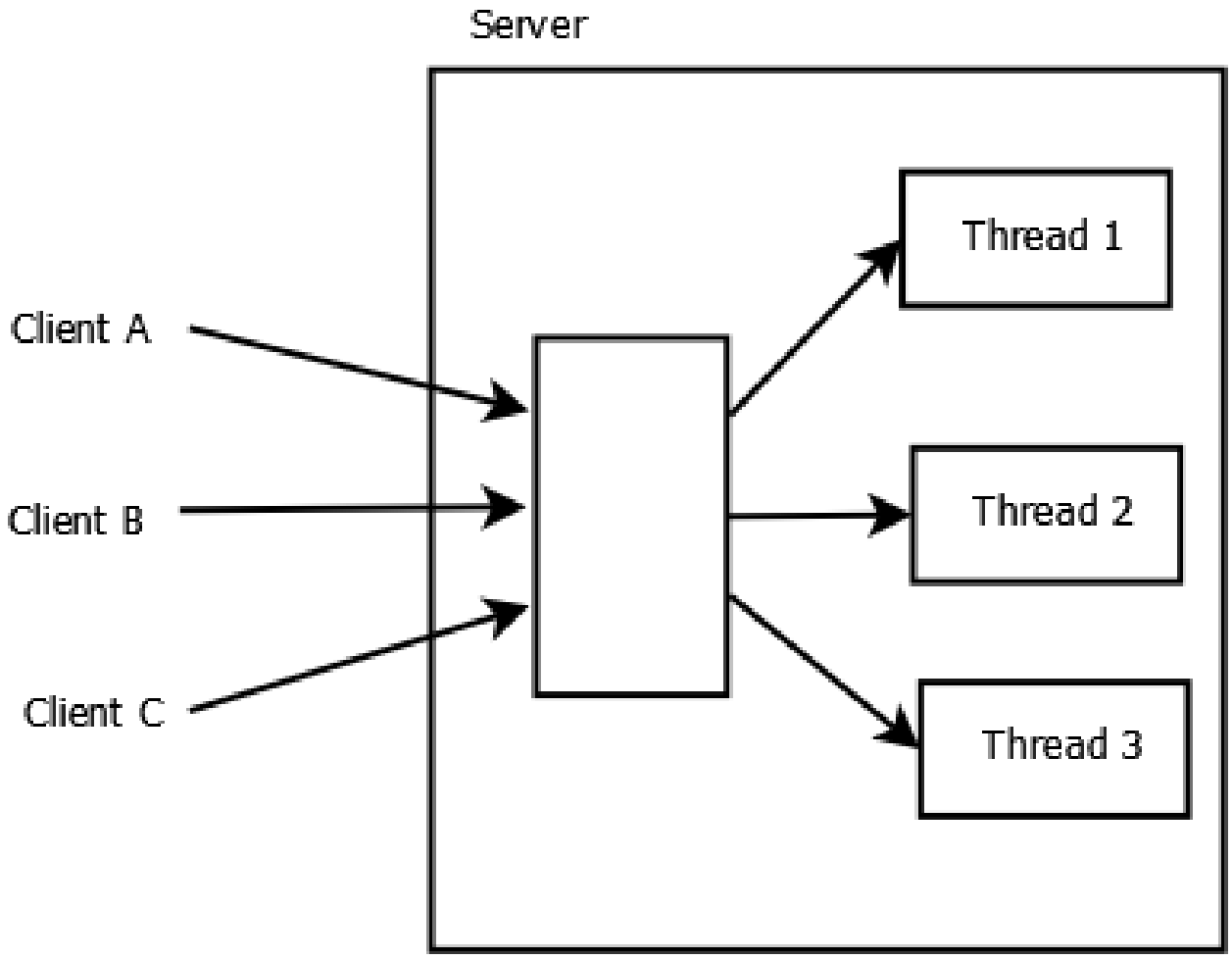


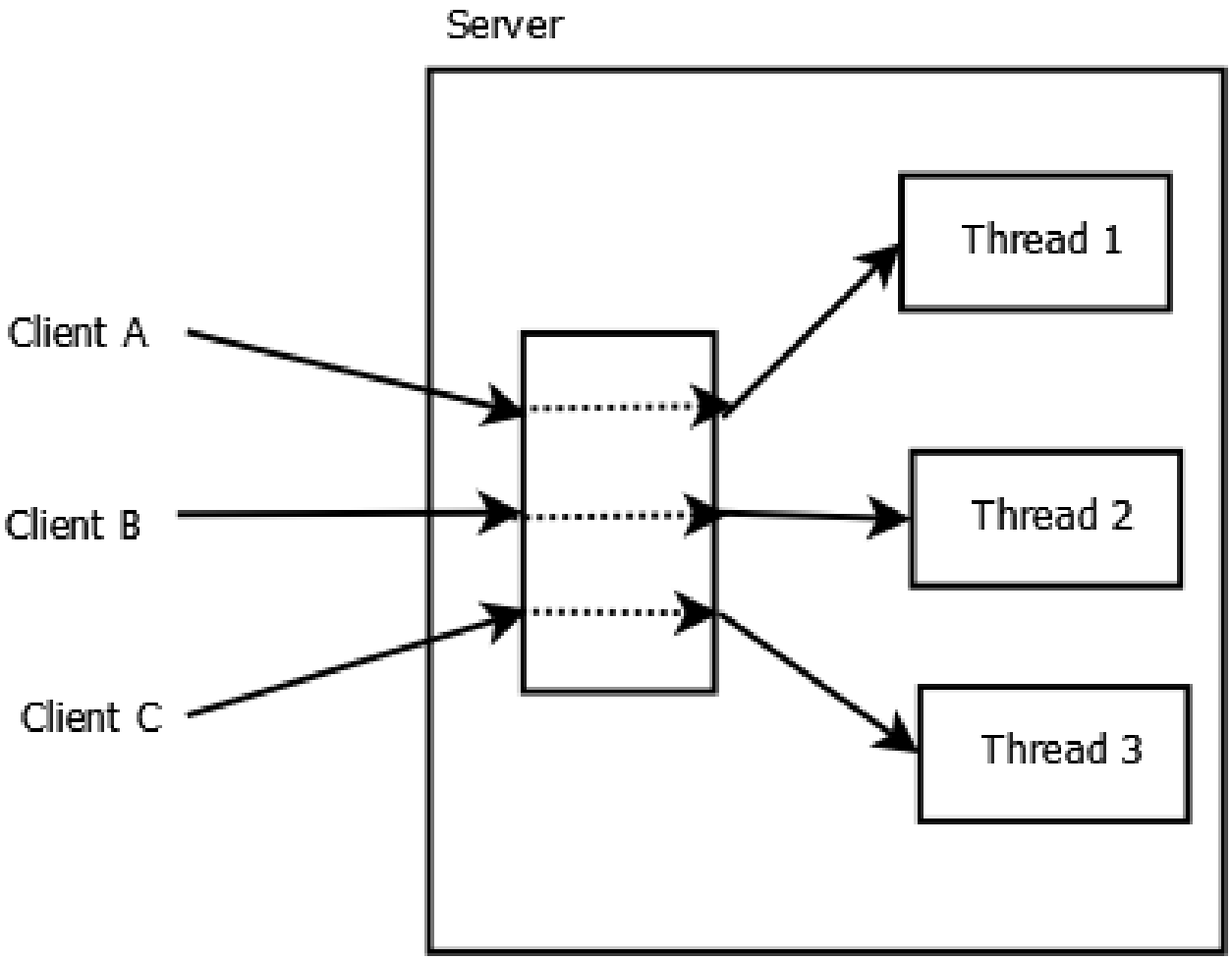


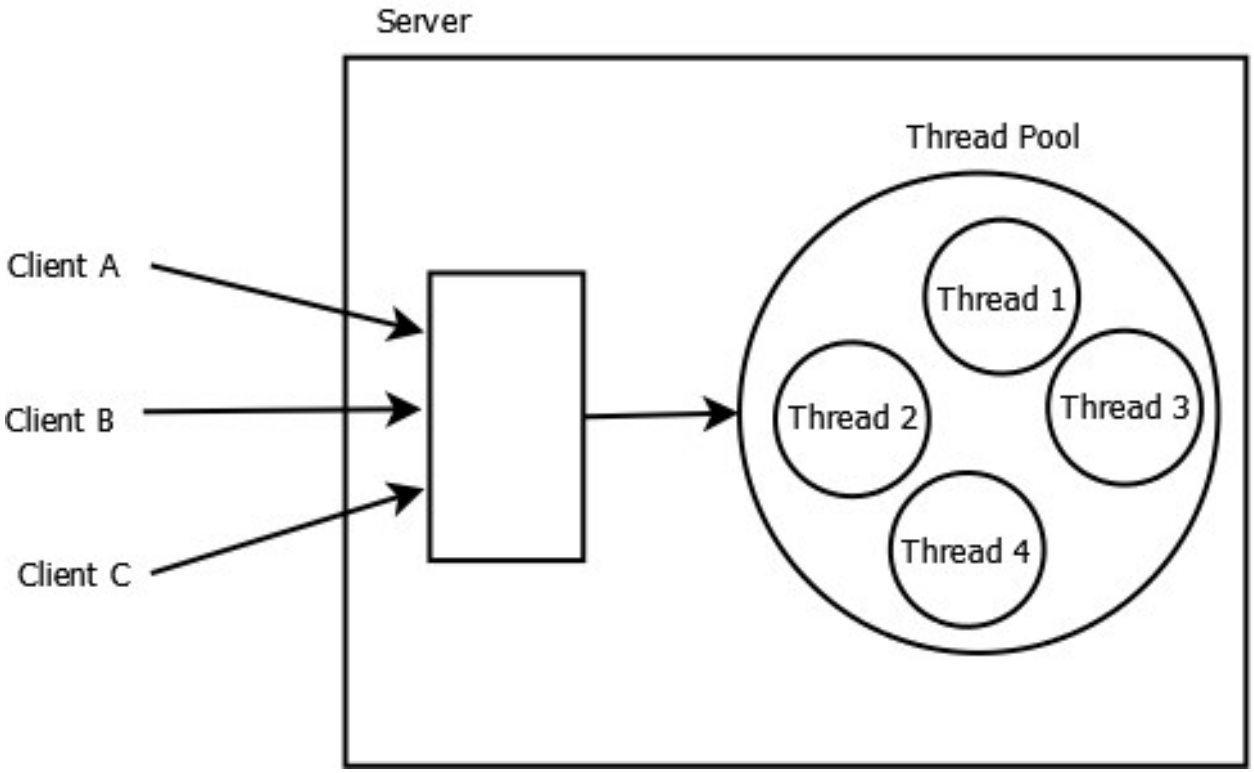
Chapter 5: Peer-to-Peer Networks

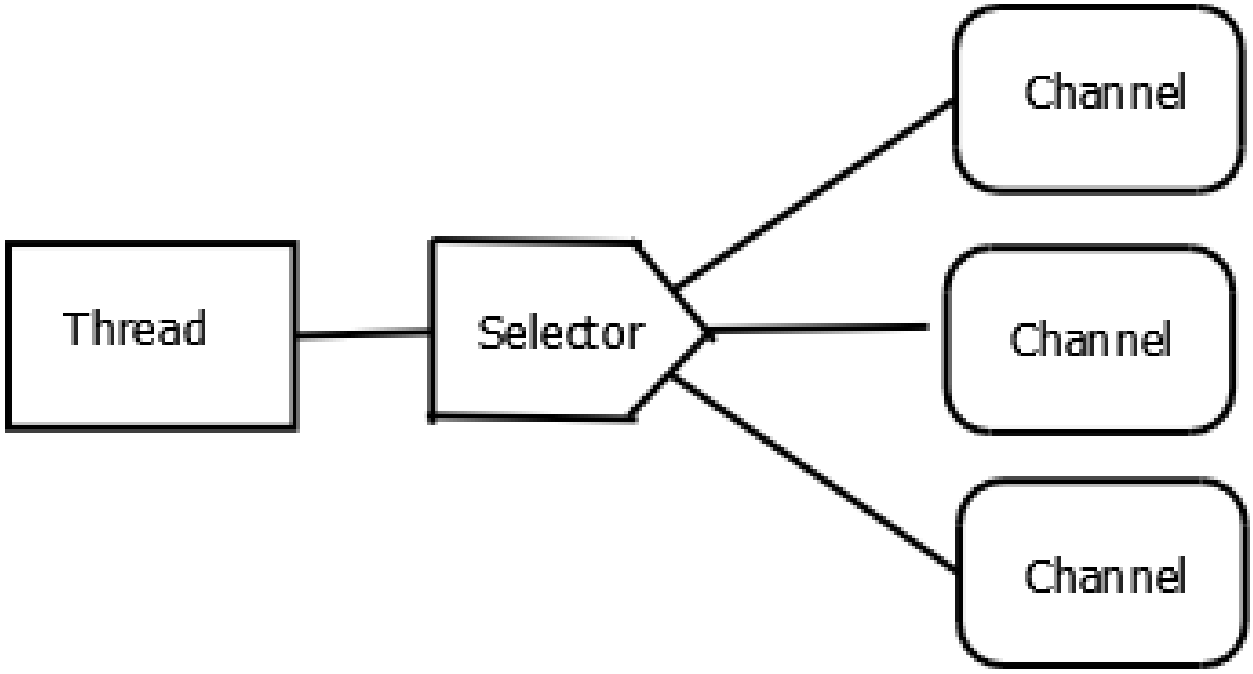


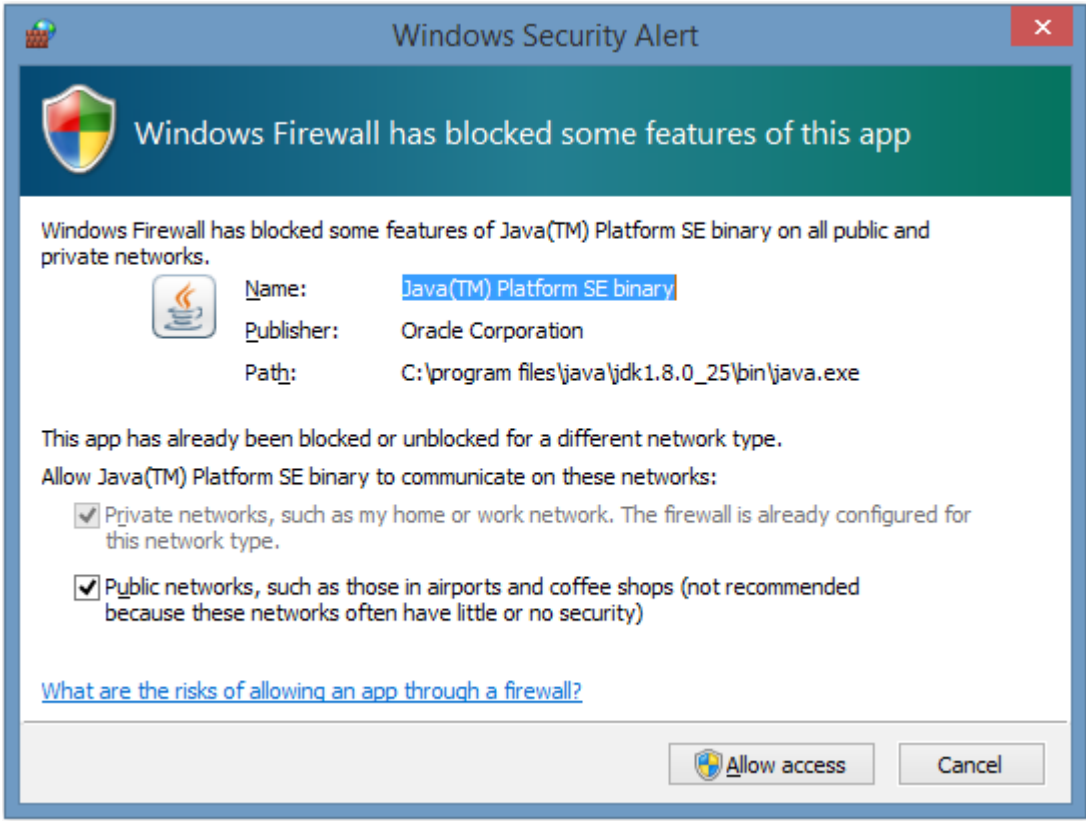
Chapter 7: Network Scalability


Server
Thread 1
Client A
Thread 2
Client B
Client C
Thread 3

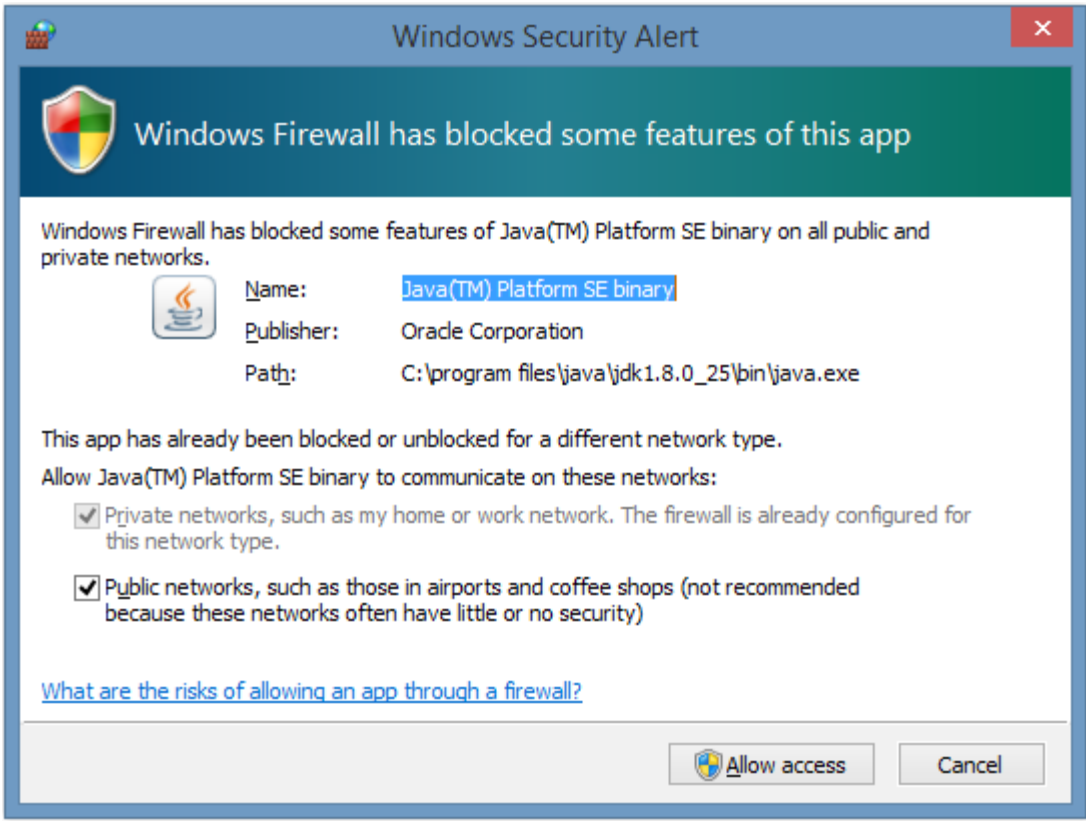

Server
Thread 1
Client A
Thread 2
Client B
Client C
Thread 3

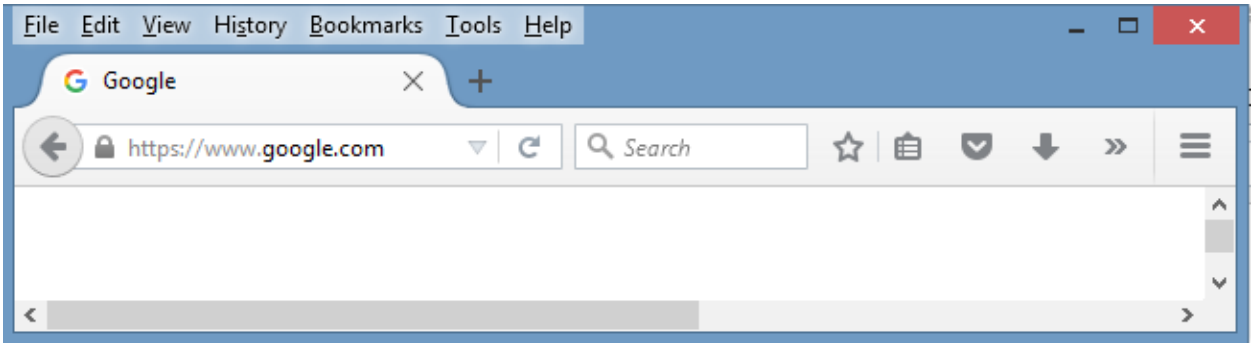

Server
Thread Pool
Client A
Client B
Client C
Thread 1
Thread 2
Thread 3
Thread 4


Thread
Selector
Channel
Channel
Channel

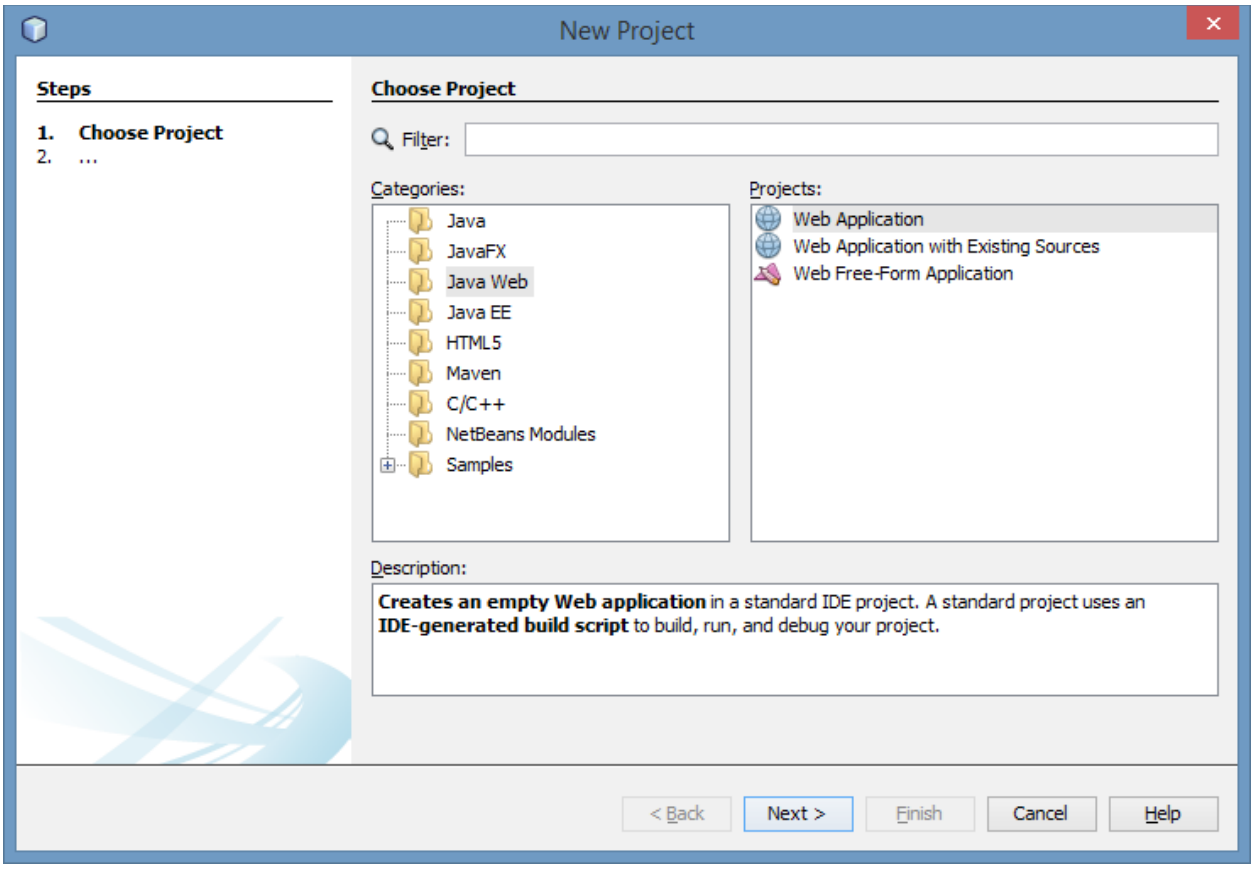


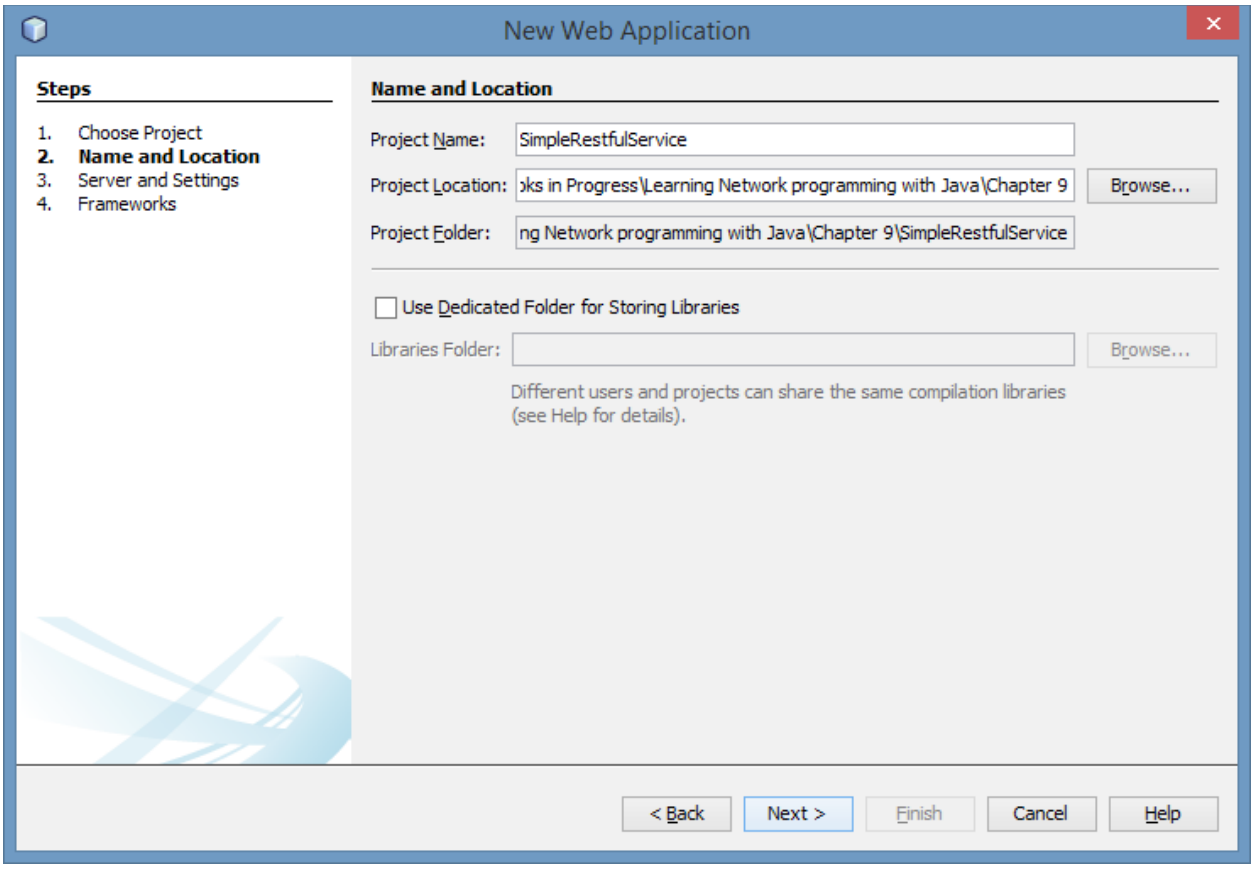
Chapter 8: Network Security

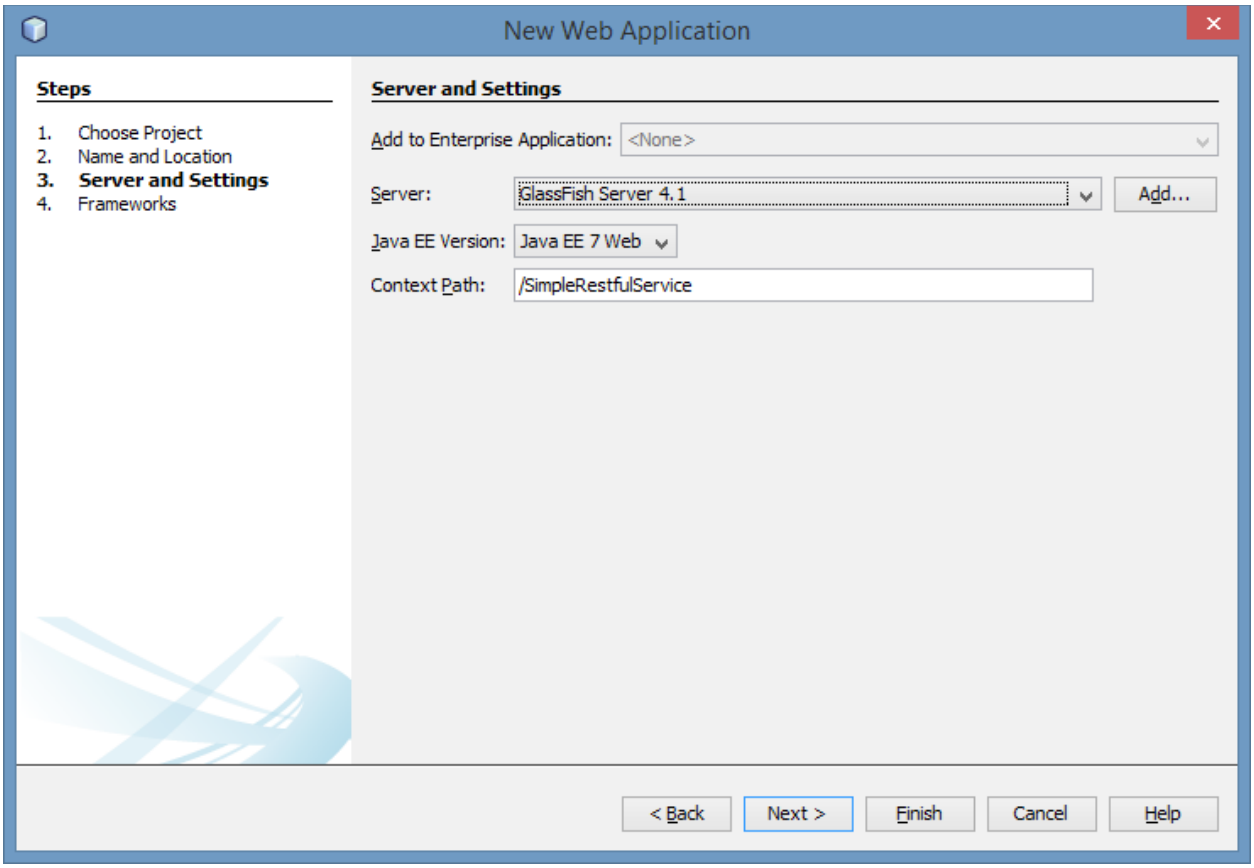


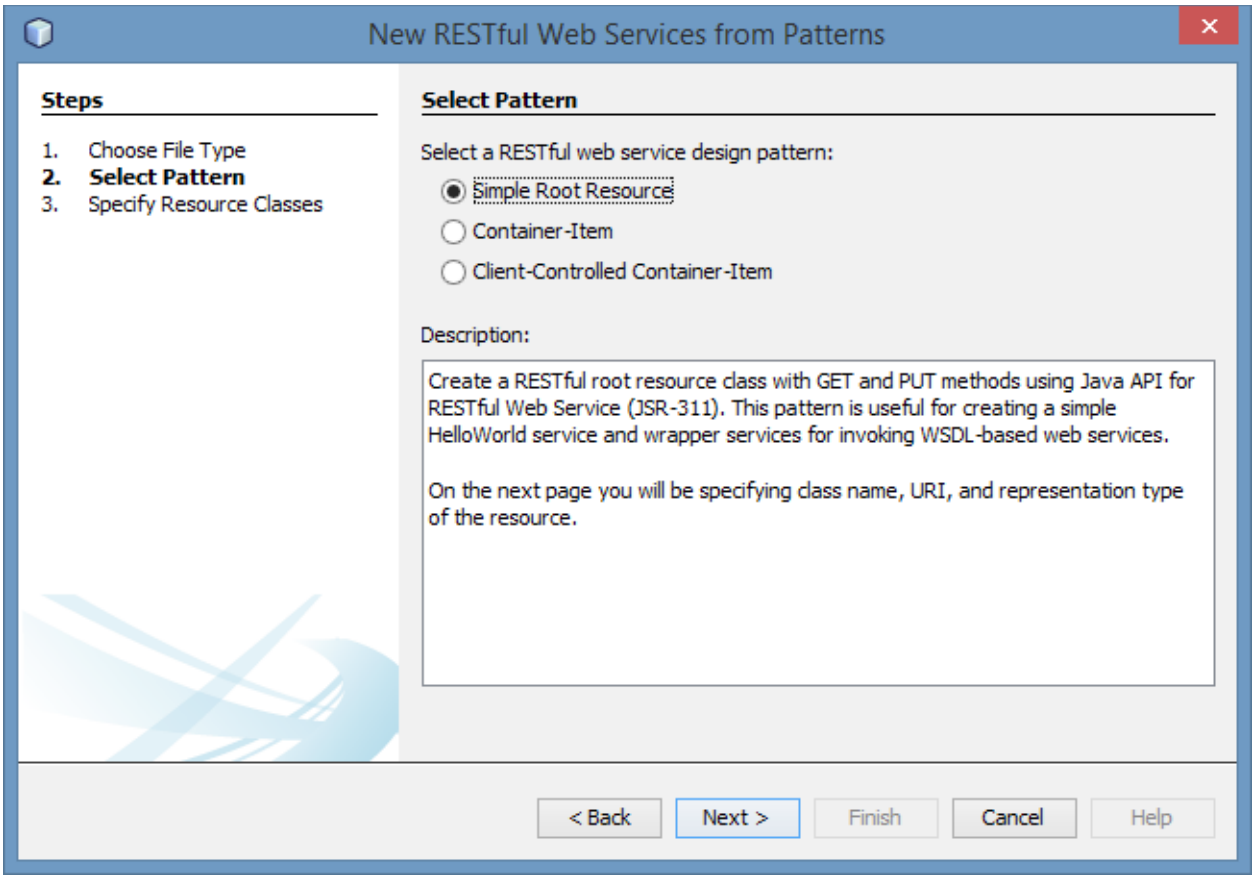


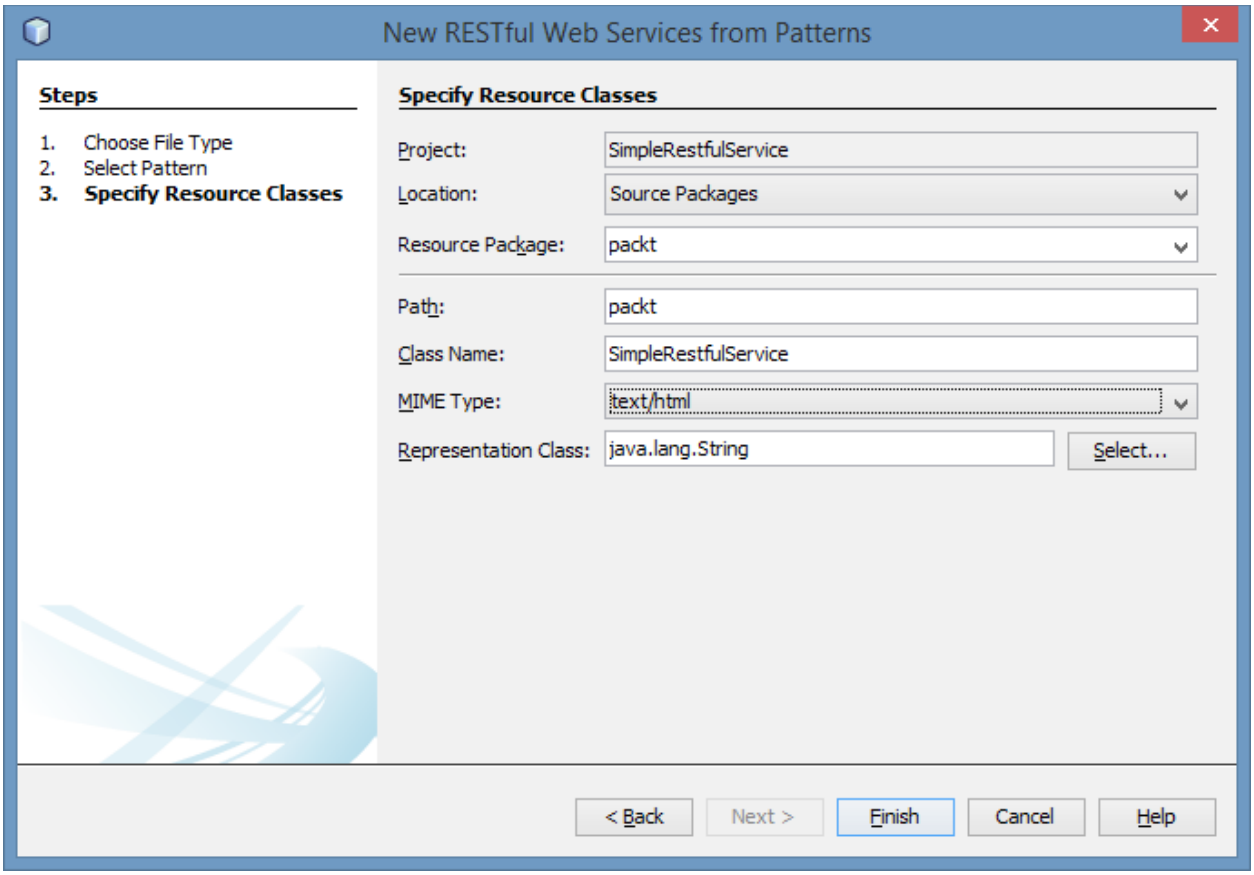
Chapter 9: Network Interoperability

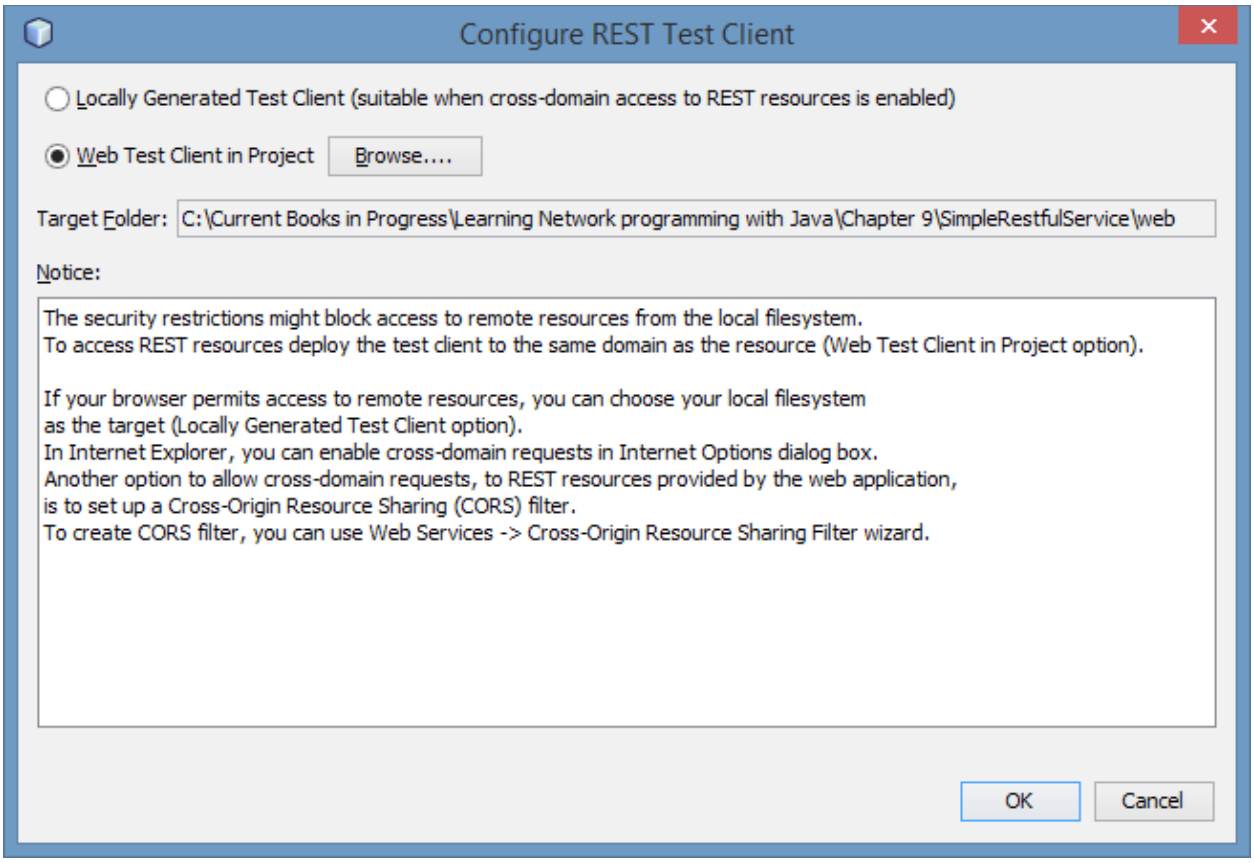

New Project
Steps
1. Choose Project
2. ...
Choose Project
Filter:
Categories:
Java
JavaFX
Java Web
Java EE
HTML5
Maven
C/C++
NetBeans Modules
Samples
Projects:
Web Application
Web Application with Existing Sources
Web Free-Form Application
Description:
Creates an empty Web application in a standard IDE project. A standard project uses an IDE-generated build script to build, run, and debug your project.
< Back
Next >
Finish
Cancel
Help

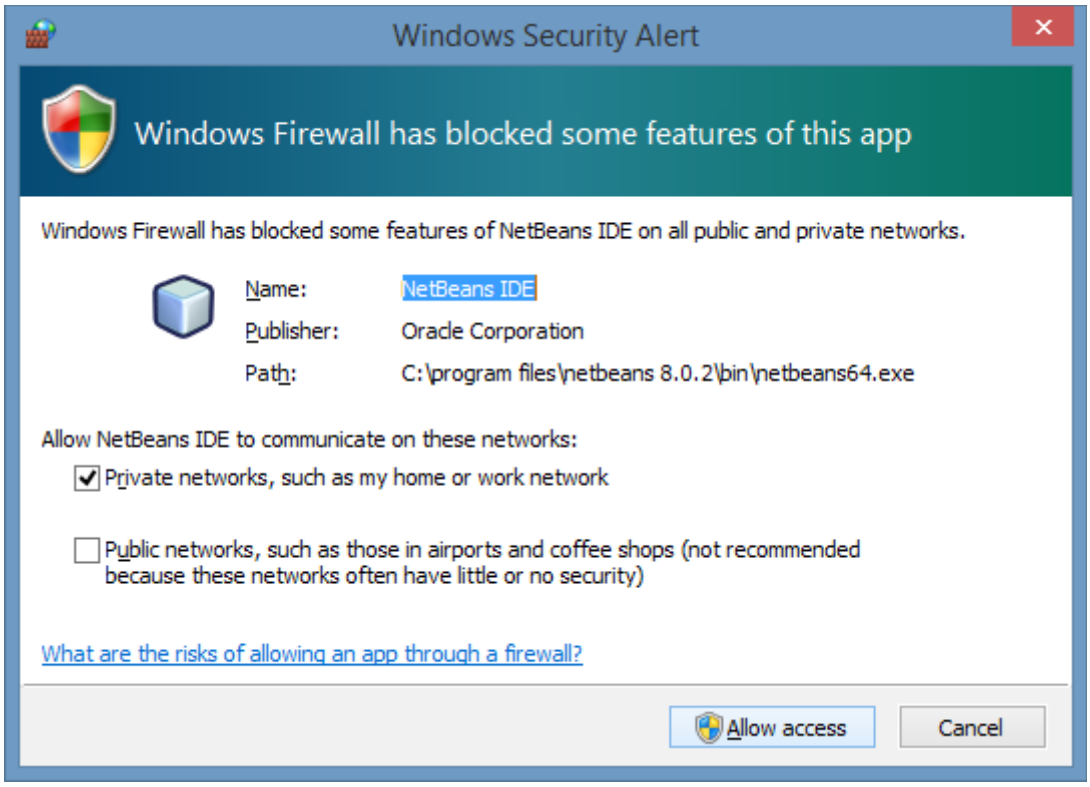

New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Name and Location
Project Name: SimpleRestfulService
Project Location: ›ks in Progress\Learning Network programming with Java\Chapter 9
Browse...
Project Folder: ng Network programming with Java\Chapter 9\SimpleRestfulService
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Browse...
Different users and projects can share the same compilation libraries
(see Help for details).
< Back
Next >
Finish
Cancel
Help

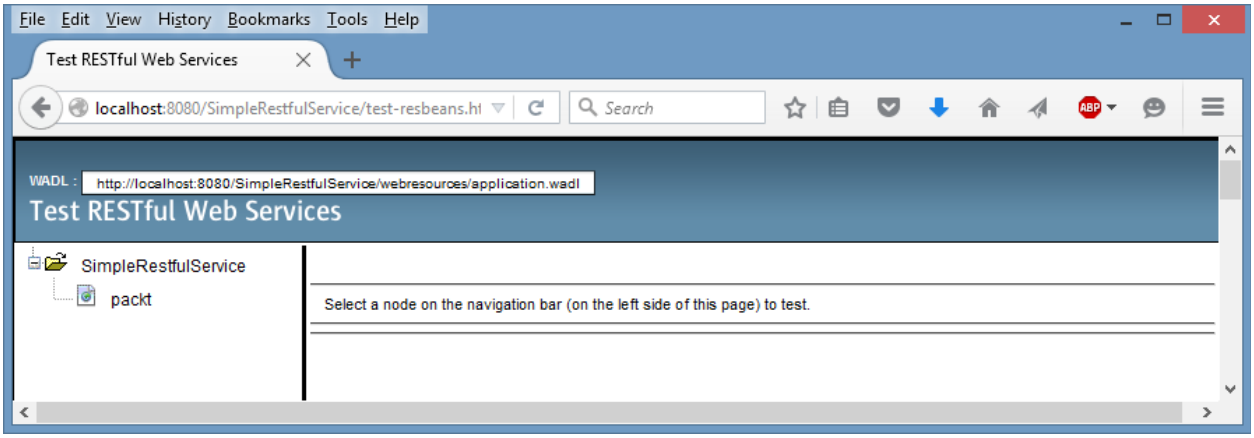

New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Server and Settings
Add to Enterprise Application: <None>
Server: GlassFish Server 4.1
Add...
Java EE Version: Java EE 7 Web
Context Path: /SimpleRestfulService
< Back
Next >
Finish
Cancel
Help

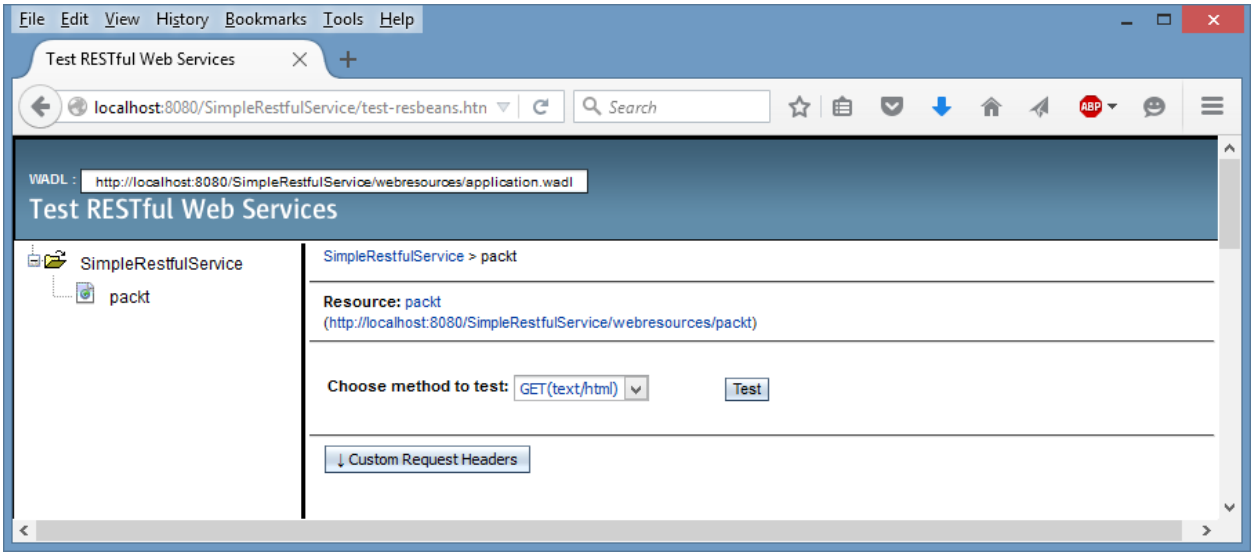

New RESTful Web Services from Patterns
Steps
1. Choose File Type
2. Select Pattern
3. Specify Resource Classes
Select Pattern
Select a RESTful web service design pattern:
Simple Root Resource
Container-Item
Client-Controlled Container-Item
Description:
Create a RESTful root resource class with GET and PUT methods using Java API for RESTful Web Service (JSR-311). This pattern is useful for creating a simple HelloWorld service and wrapper services for invoking WSDL-based web services.
On the next page you will be specifying class name, URI, and representation type of the resource.
< Back
Next >
Finish
Cancel
Help

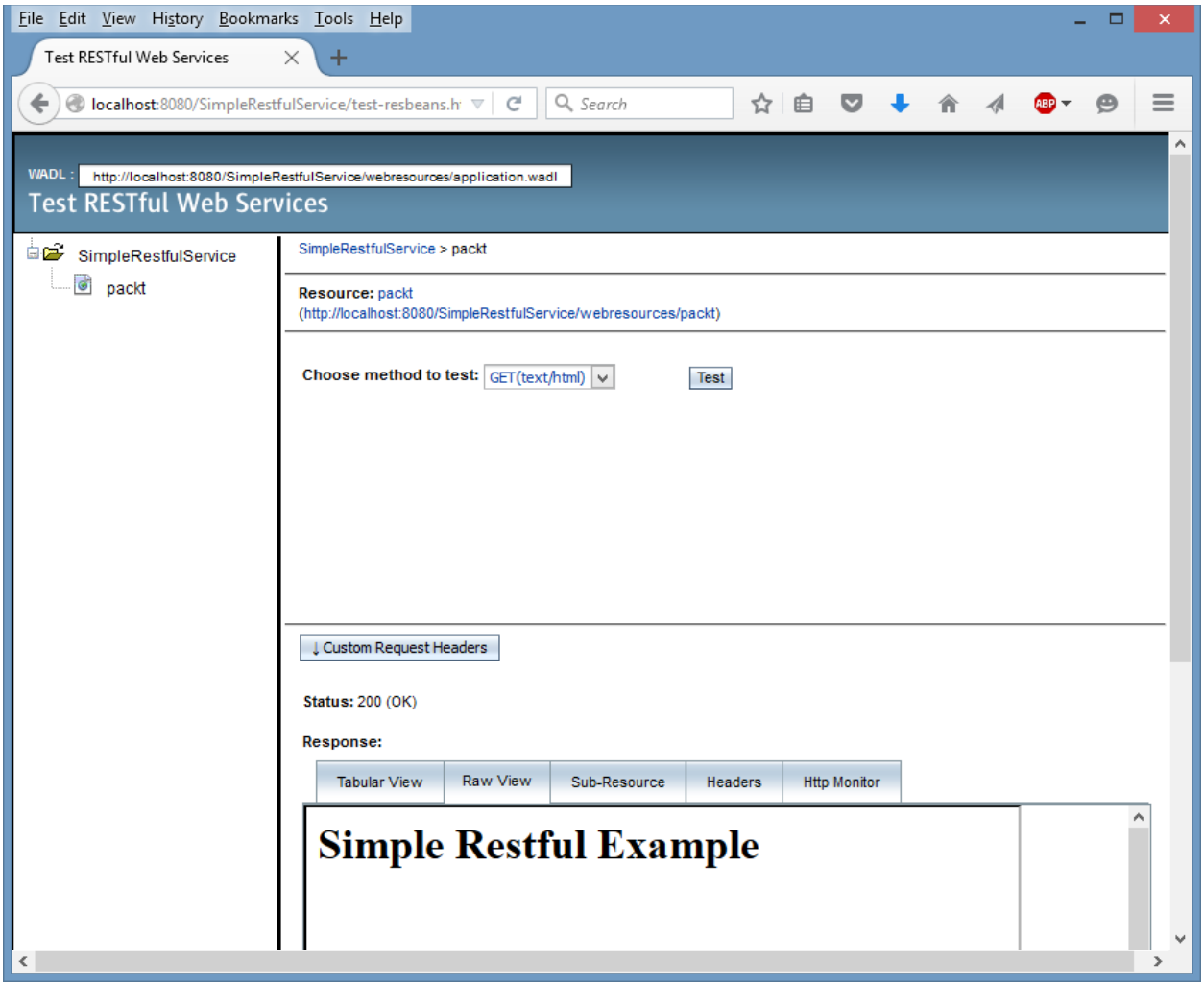

New RESTful Web Services from Patterns
Steps
1. Choose File Type
2. Select Pattern
3. Specify Resource Classes
Specify Resource Classes
Project:
SimpleRestfulService
Location:
Source Packages
Resource Package:
packt
Path:
packt
Class Name:
SimpleRestfulService
MIME Type:
text/html
Representation Class:
java.lang.String
Select...
< Back
Next >
Finish
Cancel
Help

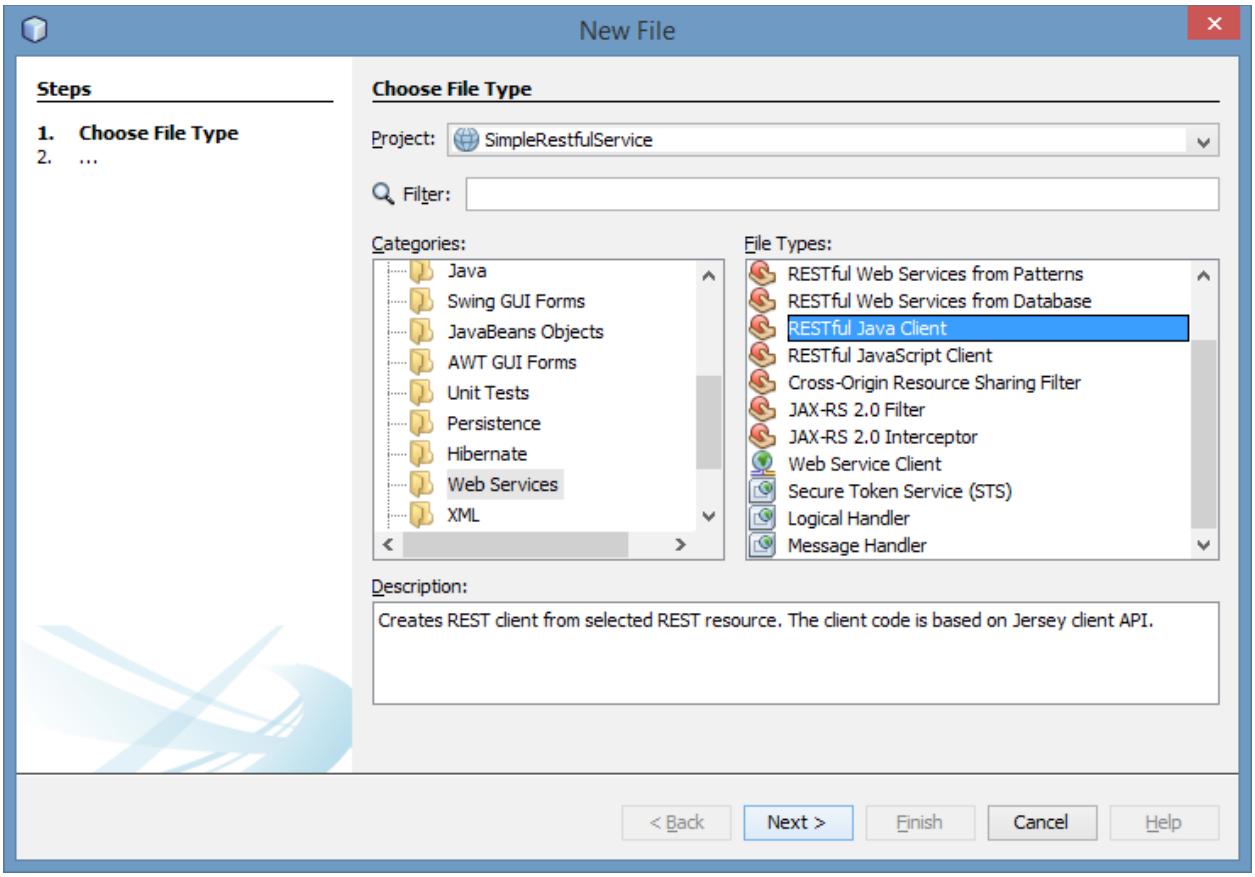

Configure REST Test Client
Locally Generated Test Client (suitable when cross-domain access to REST resources is enabled)
Web Test Client in Project
Browse....
Target Folder:
C:\Current Books in Progress\Learning Network programming with Java\Chapter 9\SimpleRestfulService\web
Notice:
The security restrictions might block access to remote resources from the local filesystem.
To access REST resources deploy the test client to the same domain as the resource (Web Test Client in Project option).
If your browser permits access to remote resources, you can choose your local filesystem
as the target (Locally Generated Test Client option).
In Internet Explorer, you can enable cross-domain requests in Internet Options dialog box.
Another option to allow cross-domain requests, to REST resources provided by the web application,
is to set up a Cross-Origin Resource Sharing (CORS) filter.
To create CORS filter, you can use Web Services -> Cross-Origin Resource Sharing Filter wizard.
OK
Cancel

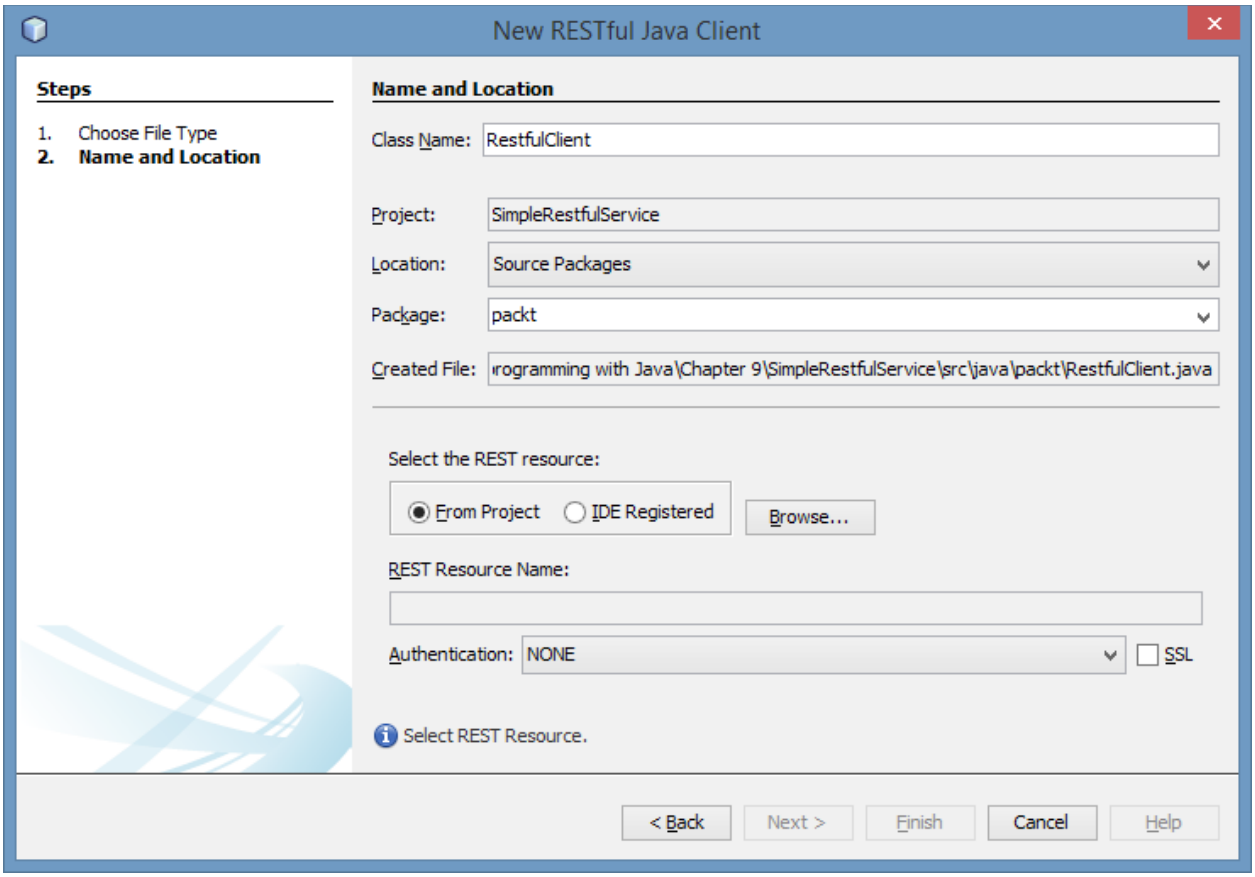

Windows Security Alert
Windows Firewall has blocked some features of this app
Windows Firewall has blocked some features of NetBeans IDE on all public and private networks.
Name: NetBeans IDE
Publisher: Oracle Corporation
Path: C:\program files\netbeans 8.0.2\bin\netbeans64.exe
Allow NetBeans IDE to communicate on these networks:
Private networks, such as my home or work network
Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)
What are the risks of allowing an app through a firewall?
Allow access
Cancel

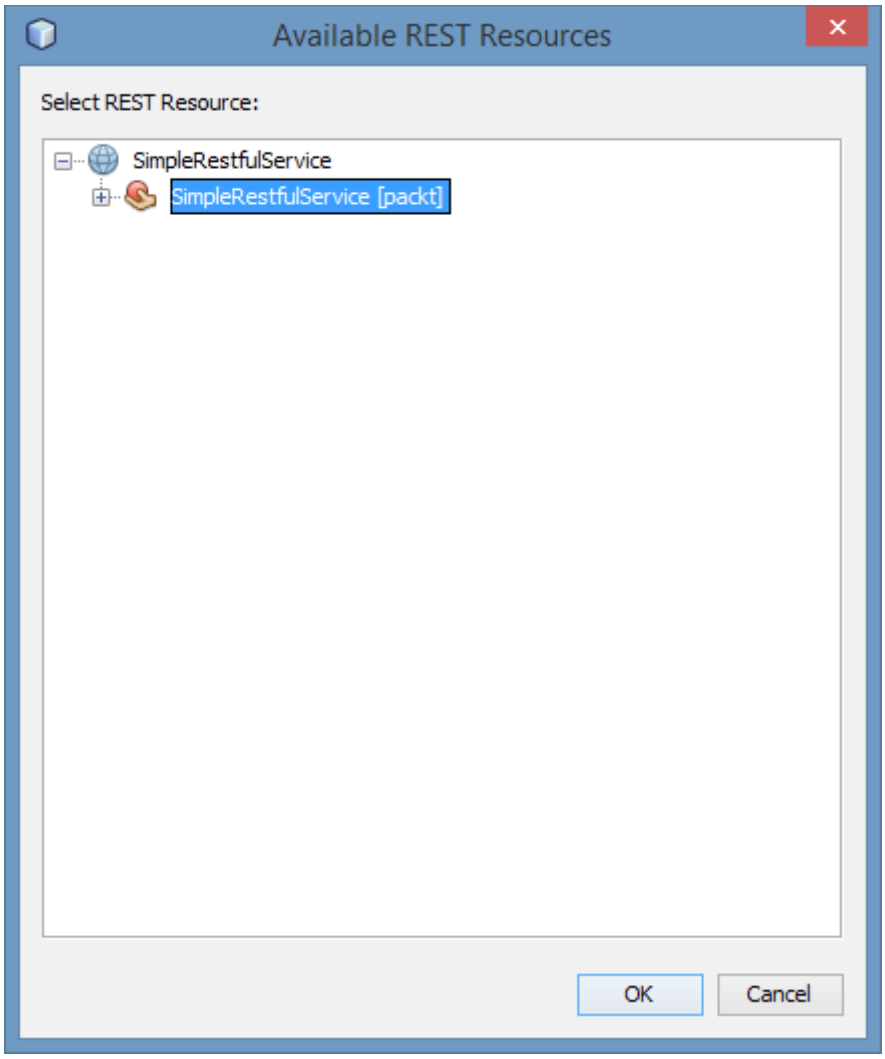

File Edit View History Bookmarks Tools Help
Test RESTful Web Services
localhost:8080/SimpleRestfulService/test-resbeans.ht
Search
WADL : http://localhost:8080/SimpleRestfulService/webresources/application.wadl
Test RESTful Web Services
SimpleRestfulService
packt
Select a node on the navigation bar (on the left side of this page) to test.

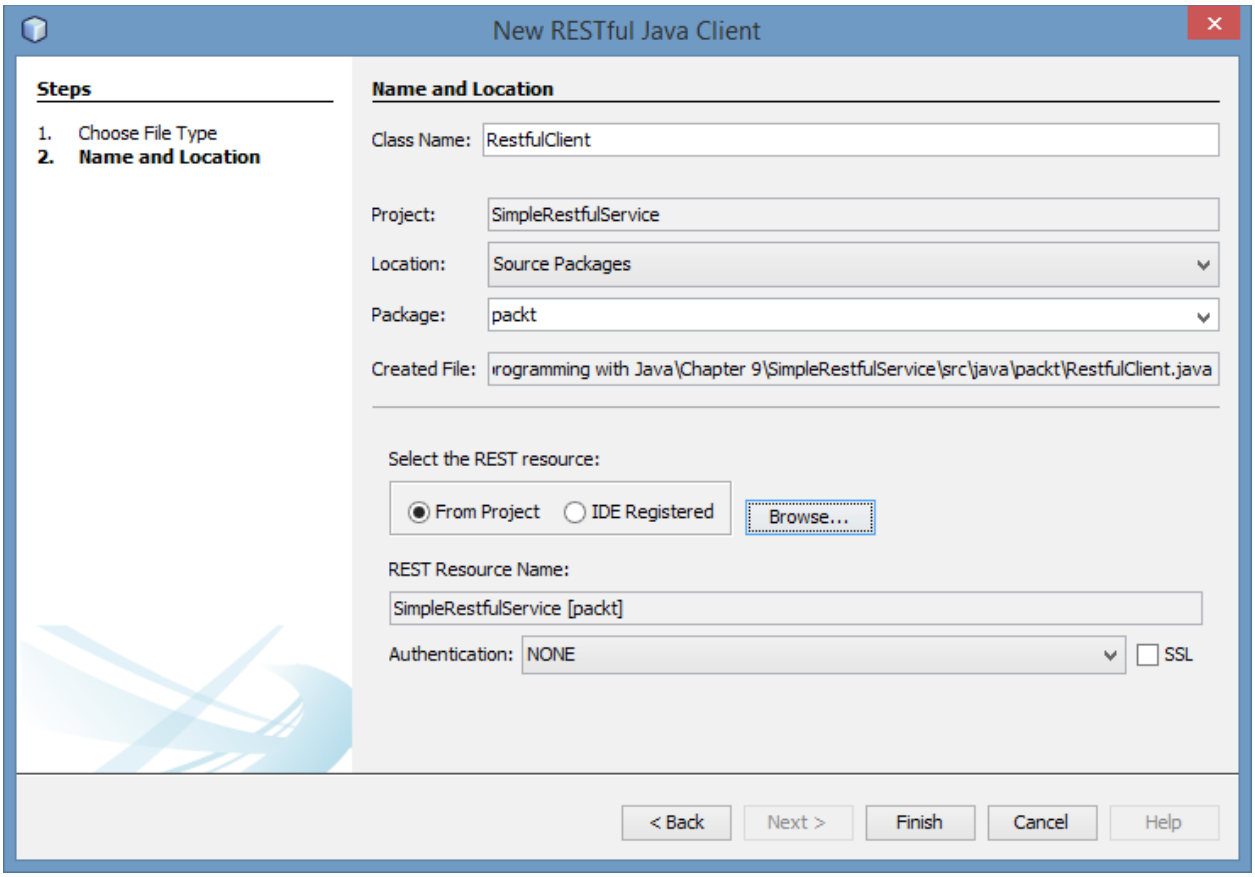

File Edit View History Bookmarks Tools Help
Test RESTful Web Services
localhost:8080/SimpleRestfulService/test-resbeans.htn
Search
WADL : http://localhost:8080/SimpleRestfulService/webresources/application.wadl
Test RESTful Web Services
SimpleRestfulService
packt
SimpleRestfulService > packt
Resource: packt
(http://localhost:8080/SimpleRestfulService/webresources/packt)
Choose method to test: GET(text/html)
Test
↓ Custom Request Headers


File Edit View History Bookmarks Tools Help
Test RESTful Web Services
localhost:8080/SimpleRestfulService/test-resbeans.h
Search
WADL : http://localhost:8080/SimpleRestfulService/webresources/application.wadl
Test RESTful Web Services
SimpleRestfulService
packt
SimpleRestfulService > packt
Resource: packt
(http://localhost:8080/SimpleRestfulService/webresources/packt)
Choose method to test: GET(text/html)
Test
Custom Request Headers
Status: 200 (OK)
Response:
Tabular View
Raw View
Sub-Resource
Headers
Http Monitor
Simple Restful Example


New File
Steps
1. Choose File Type
2. ...
Choose File Type
Project:
SimpleRestfulService
Filter:
Categories:
Java
Swing GUI Forms
JavaBeans Objects
AWT GUI Forms
Unit Tests
Persistence
Hibernate
Web Services
XML
File Types:
RESTful Web Services from Patterns
RESTful Web Services from Database
RESTful Java Client
RESTful JavaScript Client
Cross-Origin Resource Sharing Filter
JAX-RS 2.0 Filter
JAX-RS 2.0 Interceptor
Web Service Client
Secure Token Service (STS)
Logical Handler
Message Handler
Description:
Creates REST client from selected REST resource. The client code is based on Jersey client API.
< Back
Next >
Finish
Cancel
Help


New RESTful Java Client
Steps
1. Choose File Type
2. Name and Location
Name and Location
Class Name: RestfulClient
Project: SimpleRestfulService
Location: Source Packages
Package: packt
Created File: rogramming with Java\Chapter 9\SimpleRestfulService\src\java\packt\RestfulClient.java
Select the REST resource:
From Project
IDE Registered
Browse...
REST Resource Name:
Authentication: NONE
SSL
Select REST Resource.
< Back
Next >
Finish
Cancel
Help


Available REST Resources
Select REST Resource:
SimpleRestfulService
SimpleRestfulService [packt]
OK
Cancel


New RESTful Java Client
Steps
1. Choose File Type
2. Name and Location
Name and Location
Class Name: RestfulClient
Project: SimpleRestfulService
Location: Source Packages
Package: packt
Created File: rogramming with Java\Chapter 9\SimpleRestfulService\src\java\packt\RestfulClient.java
Select the REST resource:
From Project
IDE Registered
Browse...
REST Resource Name:
SimpleRestfulService [packt]
Authentication: NONE
SSL
< Back
Next >
Finish
Cancel
Help