

# Le réseau en Java

Licence DANT

PITTON Olivier

# InetAddress et URL

La classe InetAddress représente une adresse IP internet.

La classe URL représente une URL

En live `io.dant.network.cours.InetAddressExample`

# URLConnection

La classe `URLConnection` permet d'effectuer des requêtes HTTP.

API très basique, pratique si la requête est simple

En live `io.dant.network.cours.URLConnectionExample`

# TCP – NetworkInterface

La classe `NetworkInterface` permet de récupérer des informations sur les interfaces reseaux locales

En live `io.dant.network.cours.NetworkInterfaceExample`

# TCP - ServerSocket

ServerSocket est une classe représentant un serveur TCP.

C'est un objet de type socket

Attend des sockets clientes et les accepte

# TCP - Socket

Socket est une classe représentant une connection TCP cliente.

Communication par stream Java standard

Encapsule la communication TCP / SSL ...

En live ☐ `io.dant.network.cours.ServerSocketExample`

En live ☐ `io.dant.network.cours.ClientSocketExample`

# TCP - Socket

Quels est le principal problème avec le code précédent ?



# TCP - Socket

Quels est le principal problème avec le code précédent ?

Le serveur ne peut gérer qu'une seule connection à la fois

En live `io.dant.network.cours.parallel.ParallelServer`





# Le réseau avec NIO / NIO2

Licence DANT

PITTON Olivier

# Modes

Les API IO sont synchrones / bloquantes

Java NIO est synchrone / non bloquant

Java NIO2 est asynchrone / non bloquant



# NIO

Java NIO a introduit la notion de Channel de communication

On trouve donc les SocketChannel, FileChannel, ServerSocketChannel ...

A cela a été ajouté le support des charsets, selectors, Byte Buffers ...



# NIO

Java NIO a introduit la notion de Channel de communication

On trouve donc les SocketChannel, FileChannel,  
ServerSocketChannel ...

A cela a été ajouté le support des charsets, selectors, Byte  
Buffers ...

# Channel

Le code de remplacement, toujours bloquant.

En live `io.dant.network.cours.nio.ServerChannelExample`

En live `io.dant.network.cours.nio.ClientChannelExample`

# Selector

La classe Selector permet de rendre non bloquant les Channel

Elle fonctionne en enregistrant les évènements (accept / connect / read / write)

Et vous la découvrirez en TP 😊

# NIO2

Java NIO2 a introduit la notion d'asynchrone dans les communications

On trouve donc les `AsynchronousSocketChannel`, `AsynchronousFileChannel`, `AsynchronousServerSocketChannel` ...

A cela a été ajouté le support des `Path` ...



# NIO2

NIO2 fonctionne avec les Future / CompletionHandler

Chaque appel, anciennement synchrone, est effectué en tâche de fond

Se rapproche de la programmation en callback

# NIO2

Le code de remplacement en asynchrone.

En live `network.cours.nio2.AsyncServerChannelExample`

En live `network.cours.nio2.AsyncClientChannelExample`

# EchoServer

Un exemple où le serveur continue de tourner indéfiniment

En live `network.cours.full.EchoServer`

En live `network.cours.nio2.EchoClient`