

Florian DURAND
Benjamin JEDROCHA

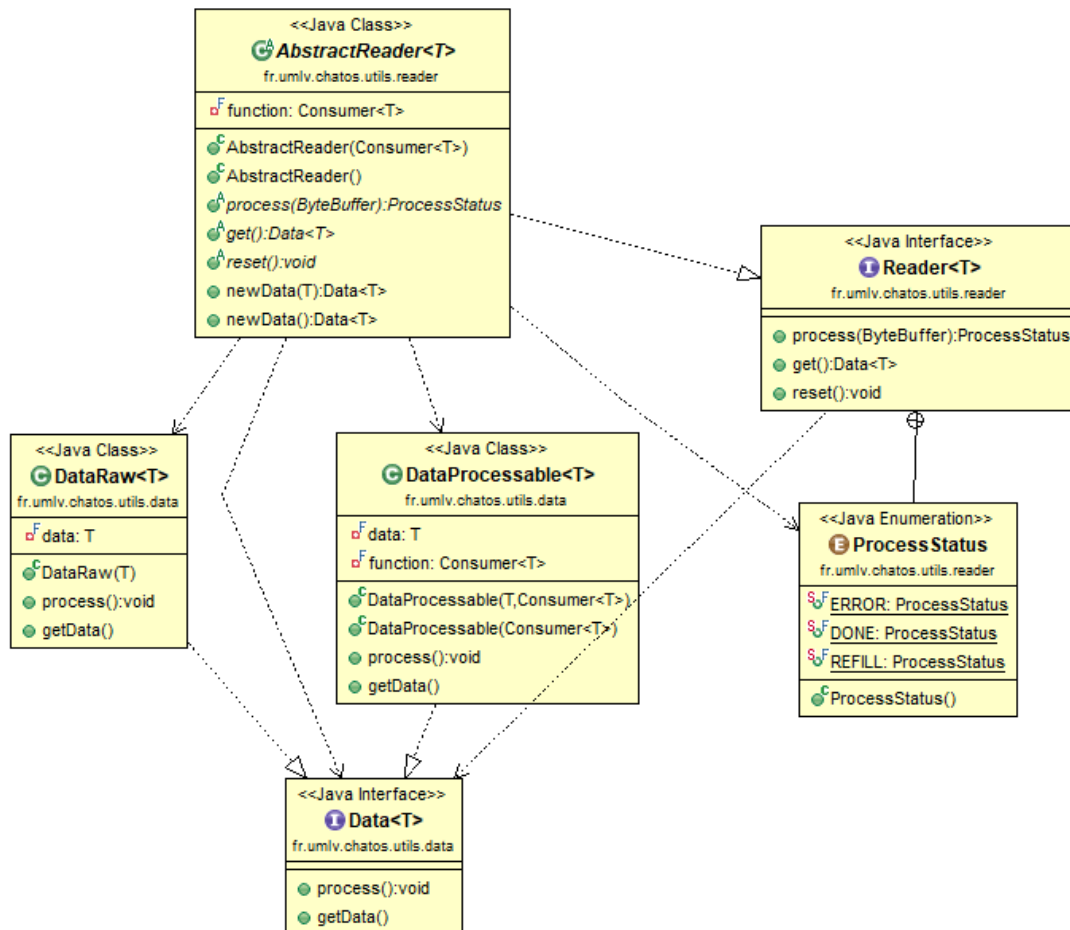
Rapport ChatOS

Architecture :	3
Reader	3
Data	3
Context	4
ReaderProcessor	4
HTTPReader	5
Fonctionnalités :	6
Evolution demandées après la soutenance bêta :	7
Console Thread	7
Code Reader	7
Accepter plusieurs demandes de connexion privée	7
Difficultés :	8
Architecture	8
HTTPReader	8
Setup des connexions privées	8
Tests	8

Architecture :

Reader

Nous avons choisi de créer des Reader génériques, c'est-à-dire qu'il n'y a pas un Reader par paquet. Il y a différents types de Reader à dispositions en fonction de ce qu'il faut lire dans un paquet. Pour créer un Reader il faut lui fournir un Consumer qui s'exécutera une fois le paquet entièrement reçu. Ce Consumer permet de faire ce que l'on veut avec les données récupérées dans le paquet reçu avec un objet Data.



Data

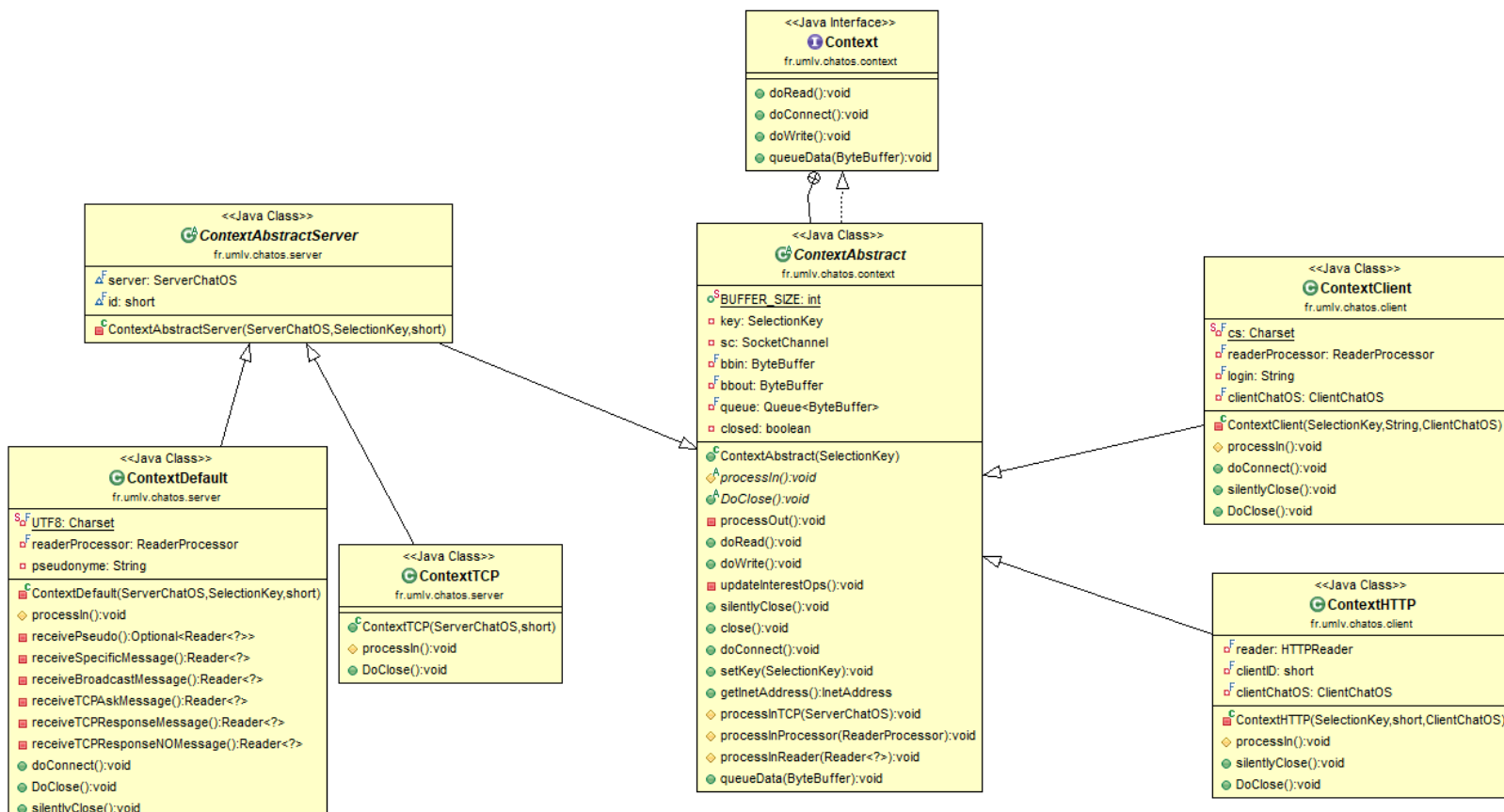
Il y a deux types de Data (voir ci-dessus) :

DataRaw pour les données qui ne seront pas traitées avec un Consumer.

DataProcessable pour les données qui seront traitées avec un Consumer.

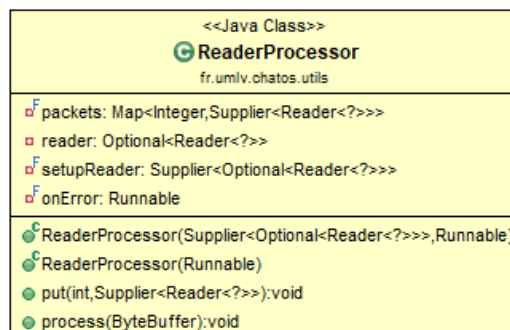
Context

Afin de traiter les connexions privées et les clients basiques nous avons choisi de créer deux Context différents dans le client et dans le serveur afin de bien déléguer les fonctionnalités à chaque context.



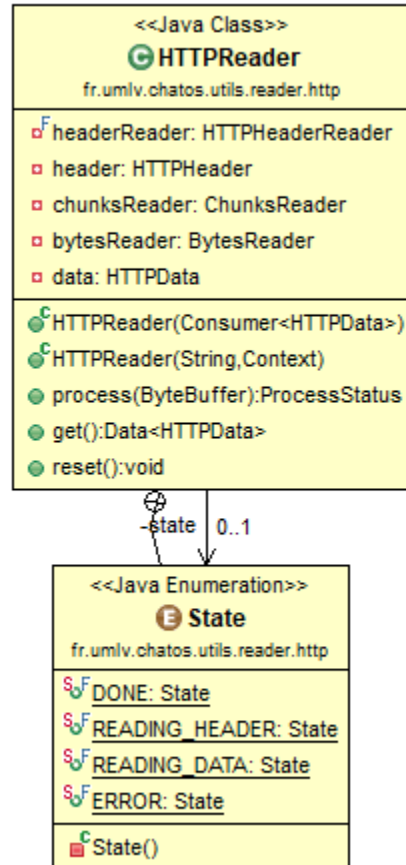
ReaderProcessor

Afin de gérer les différents reader en fonction de l'opcode lu au début des paquets nous avons créé une classe ReaderProcessor qui contient une map qui lie un opcode à un Reader.



HTTPReader

HTTPReader est capable à la fois de recevoir des requêtes GET et des réponses HTTP. En fonction de la première partie du message soit "GET", soit la version HTTP le reader s'adapte pour lire une requête ou une réponse.



Fonctionnalités :

L'ensemble des fonctionnalités demandées dans le sujet du projet ont été implémentées sans bug à notre connaissance.

Nous avons ajouté des fonctionnalités tel que l'affichage de la liste des clients, l'affichage des connexions privées ou encore l'affichage des demandes de connexions privées. Ainsi qu'une commande pour accepter ou refuser une connexion privée.

Evolution demandées après la soutenance bêta :

Console Thread

Nous avons un problème de concurrence qui n'avait pas été traité au niveau de la console et de la queue. Pour régler cela nous avons ajouté des blocs synchronized.

Code Reader

Nous avons externalisé les codes des Reader afin de ne plus les avoir dans les Context.

Accepter plusieurs demandes de connexion privée

Il n'était pas possible d'accepter plusieurs demandes de connexion privée. Maintenant c'est possible avec une liste qui enregistre les demandes qui arrivent. Le client peut accepter ou refuser les demandes à n'importe quel moment.

Difficultés :

Architecture

Nous avons eu des difficultés pour trouver une architecture qui nous convienne mais une fois que tout avait été mis en place, c'était très facile de rajouter les différentes fonctionnalités.

HTTPReader

Nous avons eu du mal à comprendre comment réussir à avoir des contextes HTTP capables à la fois de recevoir des requêtes et des réponses HTTP sans avoir à ajouter un opcode ou un état spécifique pour le contexte.

Setup des connexions privées

Nous avons eu du mal à gérer le setup des connexions privées, avec tous les scénarios possibles c'était compliqué de bien gérer tous les problèmes probables. Et surtout réussir à enregistrer les demandes de connexions et accepter les connexions des nouvelles sockets.

Tests

Réussir à tester tous les cas possibles pour les différentes fonctionnalités était vraiment compliqué.