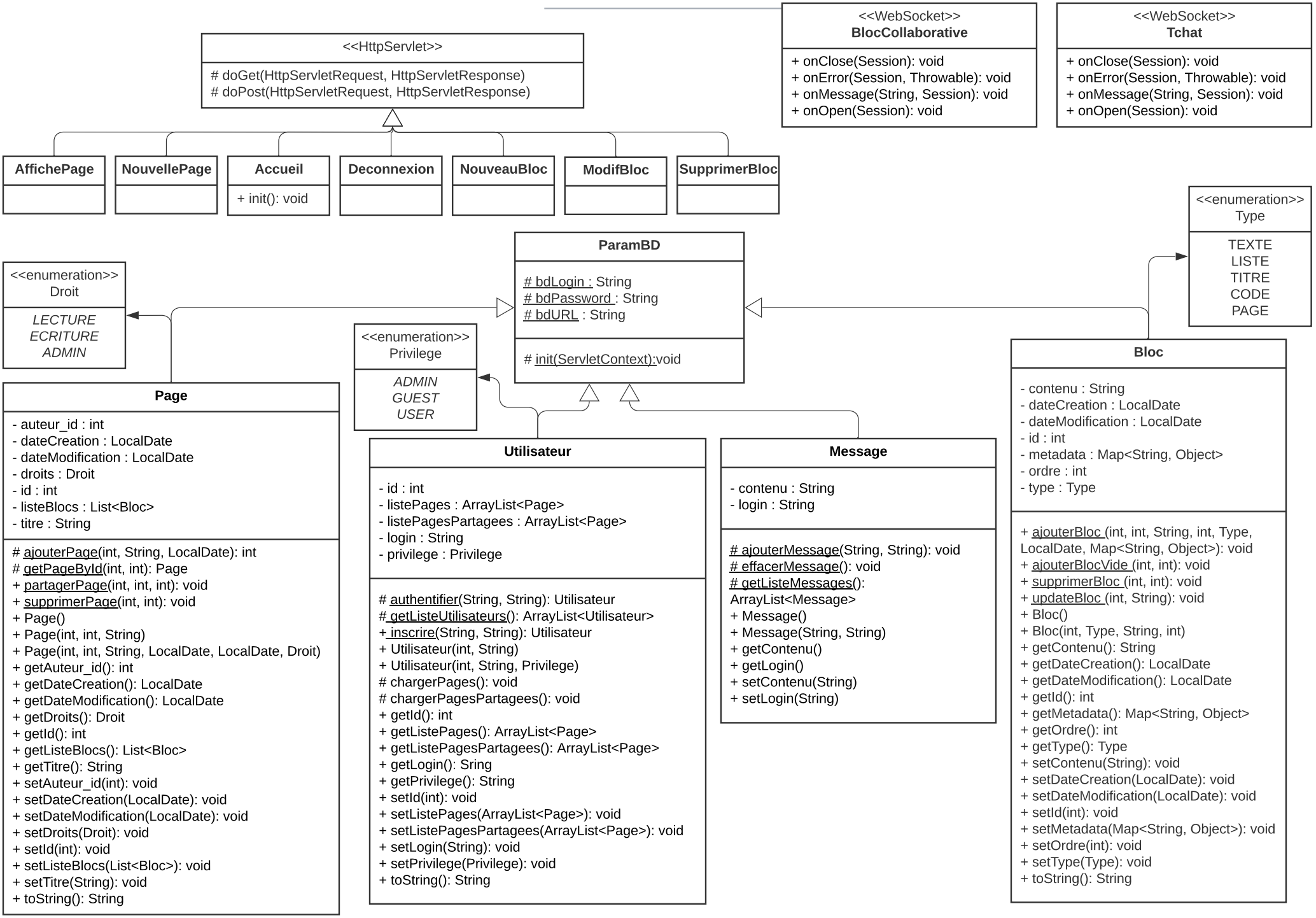
Classe UML


<<WebSocket>>
BlocCollaborative
+ onClose(Session): void
+ onError(Session, Throwable): void
+ onMessage(String, Session): void
+ onOpen(Session): void
<<WebSocket>>
Tchat
+ onClose(Session): void
+ onError(Session, Throwable): void
+ onMessage(String, Session): void
+ onOpen(Session): void
<<HttpServlet>>
doGet(HttpServletRequest, HttpServletResponse)
doPost(HttpServletRequest, HttpServletResponse)
AffichePage
NouvellePage
Accueil
+ init(): void
Deconnexion
NouveauBloc
ModifBloc
SupprimerBloc
<<enumeration>>
Type
TEXTE
LISTE
TITRE
CODE
PAGE
ParamBD
bdLogin : String
bdPassword : String
bdURL : String
init(ServletContext):void
<<enumeration>>
Droit
LECTURE
ECRITURE
ADMIN
<<enumeration>>
Privilege
ADMIN
GUEST
USER
Page
- auteur_id : int
- dateCreation : LocalDate
- dateModification : LocalDate
- droits : Droit
- id : int
- listeBlocs : List<Bloc>
- titre : String
ajouterPage(int, String, LocalDate): int
getPageById(int, int): Page
+ partagerPage(int, int, int): void
+ supprimerPage(int, int): void
+ Page()
+ Page(int, int, String)
+ Page(int, int, String, LocalDate, LocalDate, Droit)
+ getAuteur_id(): int
+ getDateCreation(): LocalDate
+ getDateModification(): LocalDate
+ getDroits(): Droit
+ getId(): int
+ getListeBlocs(): List<Bloc>
+ getTitre(): String
+ setAuteur_id(int): void
+ setDateCreation(LocalDate): void
+ setDateModification(LocalDate): void
+ setDroits(Droit): void
+ setId(int): void
+ setListeBlocs(List<Bloc>): void
+ setTitre(String): void
+ toString(): String
Utilisateur
- id : int
- listePages : ArrayList<Page>
- listePagesPartagees : ArrayList<Page>
- login : String
- privilege : Privilege
authentifier(String, String): Utilisateur
getListeUtilisateurs(): ArrayList<Utilisateur>
+ inscrire(String, String): Utilisateur
+ Utilisateur(int, String)
+ Utilisateur(int, String, Privilege)
# chargerPages(): void
# chargerPagesPartagees(): void
+ getId(): int
+ getListePages(): ArrayList<Page>
+ getListePagesPartagees(): ArrayList<Page>
+ getLogin(): Sring
+ getPrivilege(): String
+ setId(int): void
+ setListePages(ArrayList<Page>): void
+ setListePagesPartagees(ArrayList<Page>): void
+ setLogin(String): void
+ setPrivilege(Privilege): void
+ toString(): String
Message
- contenu : String
- login : String
ajouterMessage(String, String): void
effacerMessage(): void
getListeMessages(): ArrayList<Message>
+ Message()
+ Message(String, String)
+ getContenu()
+ getLogin()
+ setContenu(String)
+ setLogin(String)
Bloc
- contenu : String
- dateCreation : LocalDate
- dateModification : LocalDate
- id : int
- metadata : Map<String, Object>
- ordre : int
- type : Type
+ ajouterBloc (int, int, String, int, Type, LocalDate, Map<String, Object>): void
+ ajouterBlocVide (int, int): void
+ supprimerBloc (int, int): void
+ updateBloc (int, String): void
+ Bloc()
+ Bloc(int, Type, String, int)
+ getContenu(): String
+ getDateCreation(): LocalDate
+ getDateModification(): LocalDate
+ getId(): int
+ getMetadata(): Map<String, Object>
+ getOrdre(): int
+ getType(): Type
+ setContenu(String): void
+ setDateCreation(LocalDate): void
+ setDateModification(LocalDate): void
+ setId(int): void
+ setMetadata(Map<String, Object>): void
+ setOrdre(int): void
+ setType(Type): void
+ toString(): String