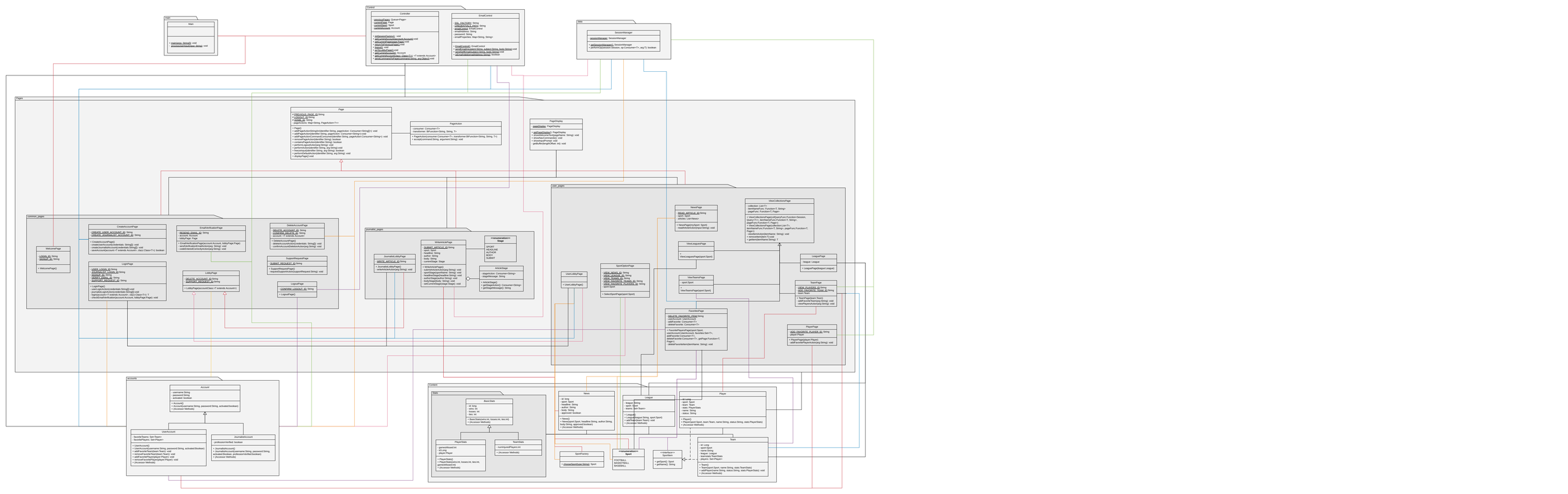
Main
Main
+ main(argv: String[]): void
- processLoginCmd(argv: String): void
Control
Controller
- previousPage(): QueuePage>
- currentPage: Page
- sessionMgr: SessionMgr
- loadedAccount: Account
+ initSessionFactory(): void
+ setLoadedAccount(acc: Account): void
+ setLastVisitedNonLoginPage(p: Page): void
+ returnToPreviousPage(): void
+ getSessionMgr()
+ setTo(toMe:Page): void
+ getLoadedAccount(): Account
+ setLoadedAccount(acc: Account<C>): void
+ setLastVisitedNonLoginPage(p: Page): void
EmailControl
SSL_FACTORY: String
CHECKSTYLE_FIXES: String
- emailHost: String = null
- emailAddress: String
- password: String
- emailProperties: Map<String, String>
+ EmailControl(): EmailControl
- sendEmail(recipient:String, subject:String, body:String): void
- sendMailWithAttachment(recipient:String, subject:String, body:String): void
- isValidEmailAddress(address:String): boolean
Data
SessionManager
- sessionManager: SessionManager
+ getSessionInstance(): SessionManager
+ performOp(session:Session, op:Consumer<T>, arg:T): boolean
Pages
Page
PREVIOUS_PAGE_ID:String
ACCOUNT_ID:String
PAGE_ID: String
- parameters: Map<String, PageAction>>
+ Page()
+ addPageAction(simpleIdentifier:String, pageAction: Consumer<String>[]): void
+ addPageActionCommandConsumer(simpleIdentifier:String, pageAction:Consumer<String>): void
+ addPageActionCommandConsumerIdentifier:String, pageAction:Consumer<String>): void
+ getPageAction(simpleIdentifier:String): PageAction
+ containsPageAction(simpleIdentifier:String): boolean
+ performLogoutAction(arg:String): void
+ performAction(simpleIdentifier:String, arg:String): void
+ hasconsumpsIdentifier:String, arg:String): boolean
+ performDefaultAction(identifier:String, arg:String): void
+ displayPage():void
PageAction
- consumer: Consumer<T>-
- transformer: BiFunction<String, String, T>-
+ PageAction(consumer:Consumer<T>, transformer:BiFunction<String, String, T>)
+ accept(commandString, argument:String): void
PageDisplay
- pageDisplay: PageDisplay
+ getPageDisplay(): PageDisplay
+ showMessage(messageName: String): void
+ showNavCommands(): void
+ getPageHeader(): void
- getBuffer(lengthOfBuf: int): void
user_pages
NewsPage
- READ_ARTICLE_ID:String
- NEWS_ID:String
- articles: List<News>-
+ NewsPage(arg:Spec Sport)
- readArticleAction(arg:String): void
ViewCollectionPage
- collection: List<T>-
- parseFunc:Function<T, String>-
- pageFunc:Function<T, Page>-
+ ViewCollectionPage(sqlQueryFunc:Function<Session, Query<T>>, itemNameFunc:Function<T, String>, pageFunc:Function<T, Page>)
+ ViewCollectionPage(collection:List<T>, itemNameFunc:Function<T, String>, pageFunc:Function<T, Page>)
- viewItemAction(itemName: String): void
+ removeItem(item:T):void
+ getItem(itemName:String): T
ViewLeaguePage
+ ViewLeaguePage(sport:Sport)
LeaguePage
- league: League
+ LeaguePage(league:League)
ViewTeamsPage
- sport:Sport
+ ViewTeamsPage(sport:Sport)
TeamPage
- VIEW_PLAYERS_ID:String
- ADD_FAVORITE_TEAM_ID:String
- team: Team
+ TeamPage(team:Team)
- addFavoriteTeam(arg:String): void
- viewPlayersAction(arg:String): void
FavoritesPage
- DELETE_FAVORITE_ITEM:String
- favorites: List<T>-
- addFavorite: Consumer<T>-
- deleteFavorite: Consumer<T>-
+ FavoritesPage(sport:Sport, userAccount:UserAccount, favorites:Set<T>, addFavorite:Consumer<T>, deleteFavorite:Consumer<T>, getPageFunction:T, Page)
- deleteFavoriteItem(itemName: String): void
PlayerPage
- ADD_FAVORITE_PLAYER_ID: String
- player:Player
+ PlayerPage(player:Player)
- addFavoritePlayerAction(arg:String): void
SportOptionPage
- VIEW_NEWS_ID: String
- VIEW_LEAGUES_ID: String
- VIEW_TEAMS_ID: String
- VIEW_FAVORITE_TEAMS_ID: String
- VIEW_FAVORITE_PLAYERS_ID: String
- sport: Sport
+ SelectSportPage(sport:Sport)
UserLoginPage
+ UserLoginPage()
common_pages
WelcomePage
+ LOGIN_ID: String
- SIGNUP_ID: String
+ WelcomePage()
CreateAccountPage
- CREATE_USER_ACCOUNT_ID: String
- CREATE_JOURNALIST_ACCOUNT_ID: String
+ CreateAccountPage()
- createUserAccount(credentials: String[]): void
- createJournalistAccount(credentials: String[]): void
- saveAccount(account:T extends Account<?>, clazz:Class<T>): boolean
LoginPage
- USER_LOGIN_ID:String
- JOURNALIST_LOGIN_ID:String
- SIGNUP_ID: String
- USER_LOGIN_ID: String
- SUPPORT_REQUEST_ID: String
+ LoginPage()
- userLogin(credentials:String[]):void
- journalistLoginAction(credentials:String[]):void
- signUpAction(account:T extends Account<?>, clazz:Class<T>): T
- checkEmailVerificationAction(account:Account, loginPagePage): void
EmailVerificationPage
- RESEND_EMAIL_ID: String
- account: Account
- loginPage: Page
+ EmailVerificationPage(account:Account, loginPagePage)
- resendVerificationAction(arg:String): void
- codeEnteredCorrectlyAction(arg:String): void
LoginPage
- DELETE_ACCOUNT_ID:String
- SUPPORT_REQUEST_ID:String
+ LoginPage(loginPagePage:Page<T extends Account>)
DeleteAccountPage
- DELETE_ACCOUNT_ID: String
- DELETE_ACCOUNT_ID: String
- account: T extends Account>
+ DeleteAccountPage()
- deleteAccountWithCredentials(credentials: String[]): void
- confirmAccountDeletionAction(arg:String): void
SupportRequestPage
- SUPPORT_REQUEST_ID:String
+ SupportRequestPage()
- requestSupportActionAction(supportRequest:String): void
LogoutPage
- CONFIRM LOGOUT_ID: String
+ LogoutPage()
journalist_pages
JournalistLobbyPage
- WRITE_ARTICLE_ID:String
+ JournalistLobbyPage()
- writeArticleAction(arg:String): void
WriteArticlePage
- SUBMIT_ARTICLE_ID:String
- sport: Sport
- headline: String
- author: String
- body: String
- currentStage: Stage
+ WriteArticlePage()
- saveArticleAction(arg:String): void
- saveStageSportHeadline: String): void
- saveStageSportHeadline: String, author): void
- authorStageAuthor:String): void
- bodyStageBody: String): void
- setCurrentStage(stage:Stage): void
<<enumeration>>
Stage
SPORT
HEADLINE
AUTHOR
BODY
SUBMIT
ArticleStage
- stageAction: Consumer<String>-
- stageMessage: String
+ ArticleStage()
+ ArticleStage(stage: Consumer<String>
+ getStageMessage(): String
Content
Stats
BaseStats
- id: long
- loses: int
- ties: int
- wins: int
+ BaseStats(wins:int, losses:int, ties:int)
+ (Accessor Methods)
PlayerStats
- gamesWonLost:int
- id:Long
- player:Player
+ PlayerStats()
+ PlayerStats(gamesWonLost:int, losses:int, ties:int, wins:int)
+ (Accessor Methods)
TeamStats
- numGroupsPlayers:int
+ (Accessor Methods)
News
- id: long
- sport: Sport
- headline: String
- author: String
- body: String
- approved: boolean
+ News()
+ News(sport:Sport, headline:String, author:String, body:String, approved:boolean)
+ (Accessor Methods)
League
- league:String
- sport: Sport
- teams: Set<Team>-
+ League()
+ League(league:String, sport:Sport)
+ addTeam(team:Team): void
+ (Accessor Methods)
Player
- id:Long
- sport: Sport
- team: Team
- stats: PlayerStats
- name: String
- status: String
+ Player()
+ Player(sport:Sport, team:Team, name:String, status:String, stats:PlayerStats)
+ (Accessor Methods)
SportFactory
+ chooseSport(sport:String): Sport
<<enumeration>>
Sport
FOOTBALL
HOCKEY
BASEBALL
<<interface>>
Team<T>
+ getSport(): Sport
+ getName(): String
Team
- id: Long
- sport: Sport
- name:String
- league: League
- stats: TeamStats
- players: Set<Player>-
+ Team()
+ Team(sport:Sport, name:String, stats:TeamStats)
+ addPlayer(name:String, status:String, stats:PlayerStats): void
+ (Accessor Methods)
accounts
Account
- username:String
- password:String
- activated: boolean
+ Account()
+ Account(username:String, password:String, activated:boolean)
+ (Accessor Methods)
UserAccount
- favoriteTeams: Set<Team>-
- favoritePlayers: Set<Player>-
+ UserAccount()
+ UserAccount(username:String, password:String, activated:Boolean)
- addFavoriteTeam(team:Team): void
- removeFavoriteTeam(team:Team): void
- addFavoritePlayer(player:Player): void
- removeFavoritePlayer(player:Player): void
+ (Accessor Methods)
JournalistAccount
- professionalVerified: boolean
+ JournalistAccount()
+ JournalistAccount(username:String, password:String, activated:Boolean, professionalVerified:boolean)
+ (Accessor Methods)