

«interfaz» ICommentService
+ addComment(userId:long, productId:long, txt:string, labelNames:List<string>): long + addLabels(commentId:long, labelNames:List<string>, userId:long): long + getLabelsByCommentId(commentId:long): List<Label> + getLabels(start:int, cnt:int): List<Label> + getCommentsByLabelId(labelId:long, start:int, cnt:int): List<Comment> + getCommentById(commentId:long): Comment + getNumberOfCommentsByLabelId(labelId:long): long + removeLabels(commentId:long, List<string> labelNames) + removeComment(userProfileId:long, commentId:long) + modifyComment(commentId:long, txt:string, userId:long, tags: List<string>) + getComment(productId:long, start:int, cnt:int): List<Comment> + getNumberOfComments(productId:long): long

«interfaz» IFavouriteService
+ getUserIdAndProductId(userId:long, productId:long): Favourite + addFavourite(userId:long, productId:long, name:string, comment:string): long + getFavouriteByUserId(userId:long, start:int, cnt:int): List<Favourite> + getNumberFavouriteByUserProfileId(userProfileId:long): long + removeFromList(favId:long, userId:long)

«interfaz» IProductService
+ searchProductByKeywords(keywords:string, start:int, size:int): List<Product> + searchProductByKeywordsXML(keywords:string, start:int, size:int): XPathNavigator + getNumberOfProducts(keyWords:string): int + searchProductByIdXML(productId:long): XPathNavigator + getUrl2Bid(productId:long): string

«interfaz» IUserService
+ RegisterUser(loginName:string, clearPassword:string, userProfileDetails:userProfileDetails): long + Login(loginName:string, password:string, passwordIsEncrypted:string):LoginResult + FindUserProfileDetails(userProfileId:long): UserProfileDetails + UpdateUserProfileDetails(userProfileId:long, userProfileDetails:UserProfileDetails) + ChangePassword(userProfileId:long, oldClearPassword:String, newClearPassword:String)

«interfaz» IValuationServie
+ addValuation(userId:long, sellerId:string, productId:long, score:long, txt:string): long + listAllVySeller(seller:string, start:int, cnt:int): List<Valuation> + getAverageAndNumberOfValuations(sellers:List<string>): Dictionary<string, AverageAndNumberOfValuations> + getAverageAndNumberOfValuations(seller:string): AverageAndNumberOfValuations + getByUserIdAndProductId(userId:long, productId:long): Valuation