

UserService

- GetUser (user id) : GetUserDto
- UpdateProfile (UpdateProfileDto) : GetUserDto
 - o Kendi useri oldugundan emin olmak lazim,
 - o Password disindaki isler buradan olacak, dark mode falan
- GetAllUsers () : List<GetUserDto>
 - o Lazim olur mu bilmiyom front-ende sormak lazim

CourseService hepsinde instructor auth kontrol edilmeli

- CreateCourse (CreateCourseDto) : returns GetCourseDto
 - o CreateCourseDto : name, courseinfo, lockdate, startdate, enddate, coursesemester, mingroupsize, maxgroupsize, sectionsizes (kac section varsa onu verecek, o kadar section objesini bu method icinde olusturacaz)
- EditCourse (EditCourseDto) : returns GetCourseDto
 - o EditCourseDto : CreateCourseDto ile neredeyse ayni, groupsizelar olmayacak bi de sectionsizes olmayacak
- AddInstructorToCourse (AddInstructorDto) : returns string ya da baska bisi
 - o AddInstructorDto : instructor id, course id icermeli, dto yerine int'lerle de olabilir ama bu daha iyi gibi. Instructor olup olmadigi control edilmeli hem ekleyenin hem eklenenin
- RemoveInstructorFromCourse (instructor id) : returns string
 - o TA'ler atilrsa falan diye. Yalniz verdigi assignmentlarin silinmemesi lazim. Ona dikkat etmek lazim.
- DeleteCourse (course id) : returns string
 - o Bu karisabilir. O coursea bagli olan her seyi silmek lazim. Ef bizim yerimize yapiyor olabilir cagriya sorulmalı. Bu servicei kullanan userin o course icinde instructor olduguna emin olmak lazim.

SectionService

- AddStudentToSection (AddStudentDto) : returns string falan
 - o AddStudentDto: id of student, id of the section
 - o Bunun icinde ogrenciye ait o sectionda bi ProjectGroup acilmali ve gerekli yerlere eklenmeli (user icinde var, section icinde de var, baska var mi control edilmeli).
- DeleteStudentFromSection (DeleteStudentDto) : returns string
 - o DeleteStudentDto: AddStudentDto ile ayni
 - o Bu da karisabilir gibi. O ogrencinin oldugu ProjectGroupun hallolmasi lazim. Tek kisi varsa silinmeli, baskalari varsa sadece kendisi cikmali ve ayriyeten confirmation statei de handle edilmeli (bir kisi cikmis haliyle ogrenciler finalize etmek istemeyebilir. Notification gidebilir belki). Bunu ya instructor cagirabilir ya da userin kendisi

ProjectGroupService

- ConfirmationOfStudent (ConfirmGroupDto) : returns GetGroupDto
 - o ConfirmGroupDto: id of student, id of ProjectGroup
 - o Bir grouptaki herkes confirmlenisse, confirmationstate = true, notification ?
- CancelConfirmationOfStudent (ConfirmGroupDto) : returns GetGroupDto

- ConfirmationStatei false yapar
 - Notification to group members ?
- ForceCancelGroup (ProjectGroup id) : returns string
 - Grubu instructor yetkisiyle tamamen dagitir.
 - Herkes tekli gruplara bolunmus gibi olur
 - Notification ?
- DeleteProjectGroup (ProjectGroupId) : returns string
 - Eger tek kisilik grupsa olur.
 - API ile olmayacak. Diger methodlar cagirabilir sadece
- CompleteJoinRequest (joinrequest id) : returns GetGroupDto
 - Tamami confirm mi control et.
 - API ile olmuyor. Icerden cagrilabiliyor
 - Notification ?
 - Gerekli slime islemlerini halletmek lazim
- CompleteMergeRequest (MergeRequest id) : returns GetGroupDto
 - Tamami confirm mi control et.
 - API ile olmuyor. Icerden cagrilabiliyor
 - Notification ?
 - Gerekli slime ekleme islemleri vs.
- LeaveGroup (projectgroup id) : returns GetGroupDto
 - Kendi user mi cagiriyo control et.
 - Gerekli cikarma islemleri, tek kisiyse cikma islemini iptal et zaten
 - Confirmation state handle
 - Notification to others ?

AssignmentService

- CreateAssignment (CreateAssignmentDto) : returns GetAssignmentDto
 - GetAssignmentDto istemememizin sebebi olusturduktan sonra UI'de gostermek lazim.
 - Kullanan user ya TA ya instructor olmalı (o kurs icinde)
 - CreateAssignmentDto: course id (Assignment modelinde section var course yok, normalde bi assignment tum kursa verilmiyor muydu?), AssignmentDescriptionFile, DueDate, CreatedAt, AcceptedTypes, maxFileSizeInBytes, VisibilityOfSubmission, CanBeGradedByStudents, IsItGraded
 - GetAssignmentDto da asagi yukari ayni
- UpdateAssignment (UpdateAssignmentDto) : returns GetAssignmentDto
 - UpdateAssignmentDto : CreateAssignmentDto gibi, course id, createdAt haric.
- DeleteAssignment (Assignment id yeter) : returns string
 - Silerken icindeki tum submissionlar da silinmeli, submission icindeki commentler de
- GetAssignment (assignment id) : returns GetAssignmentDto
- GetAllAssignments (course id) : returns List<GetAssignmentDto>
 - Isteyen kisi o course icindeyse dondurulur, sadece o courseun assignmentlari ama

SubmissionService

- Cagri yapmis zaten. Review edilse sarabilir. Yine de aklima gelenler:
- GetAllSubmissions (assignment id) : returns List<GetSubmissionDto>
 - Eger assignmentin visibilityOfSubmission == false ise sadece kendininkileri, degilse herkesinkini. Eger TA veya Instructor yapıyorsa bu eylemi, bu kural gecерli degil.

- GetMySubmissions () : List<GetSubmissionDto>
 - o Bu gerekli mi emin degilim ama front-endde vardi sanki boyle bisi
 - o Userin oldugu tum gruplardan yapilmis submissionlar donulmeli
- SubmitAssignment falan duz zaten
- DeleteSubmission ()
 - o Bu yoktu service icinde

CommentService

- SubmitComment (AddCommentDto) : returns GetCommentDto
 - o Dto'lar gerekli infolar olsun zaten demeye gerek yok
 - o Corresponding Assignmentin CanBeGradedByStudents == false ise sadece Instructor ve TA'lere izin ver
- DeleteComment (DeleteCommentDto) : returns string
- GetAllComments (submission id) : returns List<GetCommentDto>
 - o Buna kimler erisebiliyordu? Herkes visible submissionlara gelen commentleri gorebiliyor muydu sadece o gruptakiler mi? ona gore auth verilmeli
 - o TA ve Instructor zaten gorecek

JoinRequestService

- CreateJoinRequest (AddJoinRequestDto) : GetJoinRequestDto
 - o Max sizein disina cikamaz
 - o Kendi oldugu gruba atamaz
- CancelJoinRequest (CnancelJoinRequestDto) : string
 - o Join requesti atan yapabilir sadece
 - o !!!!!!!! herkes confirmlediginde yapacaz, auth olarak ya user yapacak ya da grubun tamamı confirm mi diye control edecez (OGUZHAN)
- Vote (VoteJoinRequestDto) : GetJoinRequestDto
 - o Hepsi okeyse merge islemi olacak (bu olurken de max sizei asip asmadigina bakmak lazim)
 - o Grup sizei maxsa diger join requestleri de silebiliriz.
 - o Bunun icin de auth gerekiyor (OGUZHAN)
- GetJoinRequest (join request id) : GetJoinRequestDto

MergeRequestService

- CreateMergeRequest (AddMergeRequestDto) : GetMergeRequestDto
- GetMergeRequest (Merge Request id) : GetMergeRequestDto
 - o Bunun icinde kimler confirmlemis donse sarar, grup uyeleriyle beraber
- Vote (VoteMergeRequestDto) : GetMergeRequestDto
 - o Accept veya cancel olacak
 - o Hepsi tamamsa merge request olacak
- CancelMergeRequest (CancelMergeRequestDto) : string
 - o !!!!!!!! herkes confirmlediginde yapacaz, auth olarak ya user yapacak ya da grubun tamamı confirm mi diye control edecez (OGUZHAN)

ProjectGradeService

- CreateProjectGrade (AddProjectGradeDto) : GetProjectGradeDto

- Icinde projectGroup id ve user id olabilir (user id olmasa da olur cunku zaten servicei cagirana bakiyoz, grade (negatif olmamasi ve max gradedden yuksek olmamasini control etmek lazim) ve additional comments olsun)
- GetProjectGrade (project grade id) : GetProjectDto
- GetProjectGrade (grader id, project group id) : GetProjectGradeDto
 - Farkli bi yolla da erisilebilir olmasi gerekli sanki
- DeleteProjectGrade (Project Grade id) : string
- UpdateProjectGrade (UpdateProjectGradeDto) : GetProjectGradeDto

PeerGradeService

- Add
- Edit
- Delete

PeerGradeAssignmentService

Bunlari sadece relevant kursun instructorlari yapabilir!!

- CreatePeerGradeAssignment(AddPeerGradeAssignmentDto) : GetPeerGradeAssignmentDto
 - Gerekli propertyler iste
- EditPeerGradeAssignment (UpdatePeerGradeAssignmentDto): GetPeerGradeAssignmentDto
- DeletePeerGradeAssignment (Peer Grade id)