ProjectRepositoryEloquent

// METHODS TO RELATIONS WITH USERS  
**public function** findWithOwnerAndMember($userId)  
{  
 **return** $this->scopeQuery(**function**($queryBuilder) **use**($userId) {  
 **return** $queryBuilder->select('projects.\*')  
 ->leftJoin('project\_members', 'project\_members.project\_id', '=', 'projects.id')  
 ->where('project\_members.member\_id', '=', $userId)  
 ->union($this->model->query()->getQuery()->where('owner\_id', '=', $userId));  
 })->all();  
}

ProjectControlerr

*\*/***public function** \_\_construct(ProjectRepository $repository, ProjectService $service)  
{  
 $this->repository = $repository;  
 $this->service = $service;  
 $this->middleware('check.project.owner', ['except' => ['index', 'store', 'show']]);  
 $this->middleware('check.project.permission', ['except' => ['index', 'store', 'update', 'destroy']]);  
}  
  
  
**public function** index()  
{  
 //return $this->service->all();  
 **return** $this->repository->findWithOwnerAndMember(Authorizer::*getResourceOwnerId*());  
}