EmployeeContract			
Function Name	**Storage Visibility**	**Parameters**	**Action - Notes**
createTeam	public	- string teamId - int status - string orgnization - string teamType - Employee[] teamMembers	- get sender's orgnization and combine data - register a new team as an asset to the blockchain
queryTeam	public	- string teamId	- return the team data
queryAllTeam	public		- return all teams data
addMember	public	- string teamId - string memberId - string memberName - string role	- check if the user is blong to the team orgnization - add the new member to the teamMembers array
editMember	public	- string teamId - string memberId - string memberName - string role	- check if the user is blong to the team orgnization - update the teamMember in the team data
removeMember	public	- string teamId - string memberId	- check if the user is blong to the team orgnization - remove the member from the teamMembers array
editTeam	public	- string teamId - int status - string orgnization	- check if the user is blong to the team orgnization - update the team data
removeTeam	public	- string teamId	- check if the user is blong to the team orgnization - remove the team
sendProposal	private	- string teamId - string content - string aggrement	- verify the team's status is available - get the team's orgnization - generate a unique proposal id - append the proposal to the shared proposal list - emit event to both parties
acceptProposal	private	- string fromOrgnization - string proposalId	- find the proposal and update its status to accepted - update the team's status to busy
queryProposals	private	- string privateCollectionName	- get all proposals between the sender and the destinated orgniztions

sendPaymentRecord	private	- string proposalId - string paymentAmount - string paymentProof	- find the proposal and attach the payment record to the paymentRecords array
sendTaskResultReport	private	- string proposalId - string resultReport	- find the proposal and attach the result report - update the team's status to available