

```

public interface ITaskRepository
{
    IEnumerable GetTasks();
    @Task GetTaskByID(int taskId);
    void InsertTask(@Task task);
    void DeleteTask(int taskId);
    void UpdateTask(@Task task);
    void Save();
}

public class TaskRepository : ITaskRepository
{
    private ApplicationContext context;

    public object EntityState { get; private set; }

    public TaskRepository(ApplicationContext context)
    {
        this.context = context;
    }

    public IEnumerable<@Task> GetTasks()
    {
        return context.Tasks.ToList();
    }

    public @Task GetTaskByID(int taskId)
    {
        return context.Tasks.Find(taskId);
    }

    public void InsertTask(@Task task)
    {
        context.Tasks.Add(task);
    }

    public void DeleteTask(int taskId)
    {
        Task task = context.Tasks.Find(taskId);

        context.Tasks.Remove(task);
    }

    public void UpdateTask(@Task task)
    {
        context.Entry(task).State = EntityState.Modified;
    }

    public void Save()
    {
        context.SaveChanges();
    }
}

```

```

    }

    IEnumerable ITaskRepository.GetTasks()
    {
        throw new NotImplementedException();
    }
}

public interface IRepository<TEntity> where TEntity : class
{
    void Delete(TEntity entityToDelete);

    void Delete(object id);

    IEnumerable<TEntity> Get(
        Expression<Func<TEntity, bool>> filter = null,
        Func<IQueryable<TEntity>, IOrderedQueryable<TEntity>> orderBy = null,
        string includeProperties = "");

    TEntity GetByID(object id);

    IEnumerable<TEntity> GetWithRawSql(string query,
        params object[] parameters);

    void Insert(TEntity entity);

    void Update(TEntity entityToUpdate);
}

public interface IRepository<TEntity> where TEntity : class
{
    void Delete(TEntity entityToDelete);

    void Delete(object id);

    IEnumerable<TEntity> Get(
        Expression<Func<TEntity, bool>> filter = null,
        Func<IQueryable<TEntity>, IOrderedQueryable<TEntity>> orderBy = null,
        string includeProperties = "");

    TEntity GetByID(object id);

    IEnumerable<TEntity> GetWithRawSql(string query,
        params object[] parameters);

    void Insert(TEntity entity);

    void Update(TEntity entityToUpdate);
}

```

```

}

public class TaskRepository : ITaskRepository
{
    private ApplicationContext context;

    public TaskRepository(ApplicationContext context)
    {
        this.context = context;
    }

    public IEnumerable<@Task> GetTasks()
    {
        return context.Tasks.ToList();
    }

    public @Task GetTaskByID(int taskId)
    {
        return context.Tasks.Find(taskId);
    }

    public void InsertTask(@Task task)
    {
        context.Tasks.Add(task);
    }

    public void DeleteTask(int taskId)
    {
        Task task = context.Tasks.Find(taskId);
        context.Tasks.Remove(task);
    }

    public void UpdateTask(@Task task)
    {
        context.Entry(task).State = EntityState.Modified;
    }

    public void Save()
    {
        context.SaveChanges();
    }
}

```