

springboot 整合 mybatis-plus 实现 CRUD

CRUD

mybatis-plus 实现 CRUD 的 Service 和 ServiceImpl

实现 CRUD 接口

实现 Entity 和 DTO

实现 Mapper

```
/*
 * @Description: 实现 CRUD
 * @Author: donglietao@163.com
 * @ENT: 实体
 * @DTO: DTO
 */

public interface CrudService<ENT, DTO> {

    List<DTO> list(@Param("ew") Wrapper<ENT> queryWrapper);

}
```

实现 BaseMap<ENT> 接口

```
/*
 * @description: CrudServiceImpl
 * @author: donglietao@163.com
 * */
@Service
```

```

@Slf4j
public class CrudServiceImpl<ENT, DTO> implements CrudService<ENT, DTO> {

    // 数据库mapper
    @Autowired
    protected BaseMapper<ENT> baseMapper;

    private Class<DTO> getDTOClass() {
        return (Class<DTO>) ReflectionKit.getSuperClassGenericType(this.getClass(),
CrudService.class, 1);
    }

    public List<DTO> list(@Param("ew") Wrapper<ENT> queryWrapper) {

        List<ENT> list = baseMapper.selectList(queryWrapper);
        if (list == null) {
            return null;
        }
        // 数据库mapper
        List<DTO> listDTO = new ArrayList<>(list.size());
        Class<DTO> dtoClass = getDTOClass();
        for (ENT ent : list) {
            try {
                DTO dto = dtoClass.newInstance();
                BeanUtils.copyProperties(ent, dto);
                listDTO.add(dto);
            } catch (Exception ex) {
                log.error("CrudServiceImpl.list() error:", ex);
            }
        }
        return listDTO;
    }
}

```

数据库CRUD

```

public interface UserService extends CrudService<UserEntity, UserDTO> {
}

@Service
public class UserServiceImpl extends CrudServiceImpl<UserEntity, UserDTO> implements
UserService {

}

```



```

@Slf4j
public class ConvertUtils {

    public static <T> T sourceToTarget(Object source, Class<T> target) {
        if (source == null) {
            return null;
        }
        T targetObject = null;
        try {
            targetObject = target.newInstance();
            BeanUtils.copyProperties(source, targetObject);
        } catch (Exception e) {
            log.error("convert error ", e);
        }

        return targetObject;
    }

    public static <T> List<T> sourceToTarget(Collection<?> sourceList, Class<T> target) {
        if (sourceList == null) {
            return null;
        }
    }
}

```

```

List targetList = new ArrayList<>(sourceList.size());
try {
    for (Object source : sourceList) {
        T targetObject = target.newInstance();
        BeanUtils.copyProperties(source, targetObject);
        targetList.add(targetObject);
    }
} catch (Exception e) {
    log.error("convert error ", e);
}

return targetList;
}
}

```



```

@Service
@Slf4j
public class CrudServiceImpl<ENT, DT0> implements CrudService<ENT, DT0> {

    @Autowired
    protected BaseMapper<ENT> baseMapper;

    private Class<DT0> getDT0Class() {
        return (Class<DT0>) ReflectionKit.getSuperClassGenericType(this.getClass(),
CrudService.class, 1);
    }

    public List<DT0> list(@Param("ew") Wrapper<ENT> queryWrapper) {

        List<ENT> list = baseMapper.selectList(queryWrapper);
        // 
        List<DT0> listDT0 = ConvertUtils.sourceToTarget(list, getDT0Class());
        return listDT0;
    }
}

```

□□ #3

□ □□ □□ 2 □ 2025 03:36:44

□ □□ □□ 2 □ 2025 04:12:44