

# 基于mybatis的数据持久层 研究

汇报人：

2024-01-25

## 目录

- 引言
- MyBatis框架概述
- 数据持久层技术原理
- 基于MyBatis的数据持久层设计与实践
- 性能优化与扩展性研究
- 总结与展望

01

引言

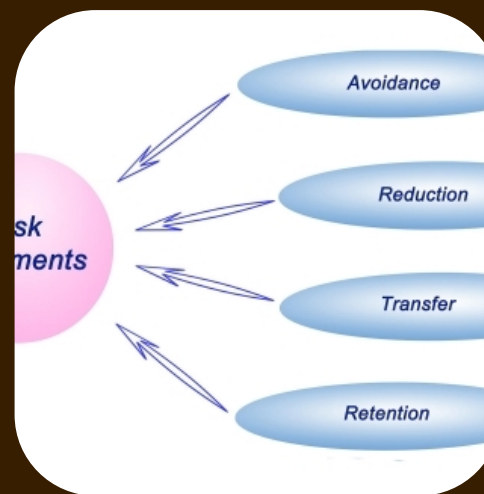
# 研究背景与意义



互联网应用快速发展，数据持久层作为应用重要组成部分，其性能直接影响应用整体表现。



MyBatis作为一款优秀的持久层框架，具有灵活、高效、易用等特点，在Java领域得到广泛应用。



研究基于MyBatis的数据持久层，对于提高应用性能、优化数据库操作、降低开发成本等具有重要意义。



# 研究目的和内容



01

研究目的

02

探究MyBatis框架的原理和机制。

03

分析MyBatis在数据持久层应用中的优势和不足。



# 研究目的和内容

- 提出基于MyBatis的数据持久层优化方案。





# 研究目的和内容

01

研究内容

02

MyBatis框架核心原理及运行机制。

03

MyBatis与数据库交互方式及性能分析。

04

基于MyBatis的数据持久层优化策略与实践。



# 国内外研究现状及发展趋势

1

国内外研究现状

2

目前，国内外对MyBatis框架的研究主要集中在原理分析、性能优化、扩展应用等方面。

3

针对MyBatis的不足，一些研究者提出了改进方案，如引入缓存机制、优化SQL语句等。



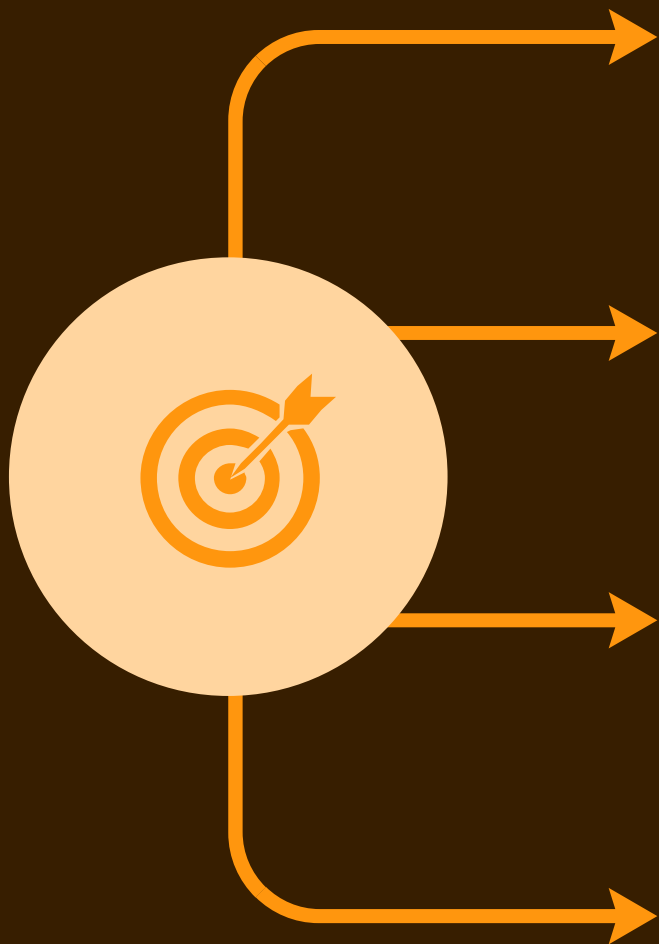


# 国内外研究现状及发展趋势





# 国内外研究现状及发展趋势



01

发展趋势

02

随着云计算、大数据等技术的不断发展，数据持久层将面临更多挑战和机遇。

03

未来，基于MyBatis的数据持久层研究将更加注重性能优化、分布式处理、智能化管理等方面。

04

同时，随着新技术的不断涌现，MyBatis框架本身也将不断完善和发展，以适应不断变化的市场需求和技术趋势。

02

# MyBatis框架概述



# MyBatis框架介绍

MyBatis是一个优秀的持久层框架，它支持自定义SQL、存储过程以及高级映射。



MyBatis避免了几乎所有的JDBC代码和手动设置参数以及获取结果集。



MyBatis可以使用简单的XML或注解来配置和映射原始类型、接口和Java的POJO ( Plain Old Java Objects , 普通的Java对象 ) 为数据库中的记录。



```
        content="author">
        on" content="description">
        rds" content="keywords">
        8">
        pe="image/icon" href="favicon.ico">
        meet" type="text/css" href="style.css">

        { margin:0; padding:0; }
        { clear:both; }
        { content:" "; display:block; height:0; clear:both; visibility:hidden; }
        { float:right; }
        { float:left; }
        { border:0; }
        { max-width:100%; }
        { display:block; }
        { margin:0;padding:0; }

        src="script.js"></script>
        ument).ready(function(){

        /header>
        >
        article>
```



# MyBatis核心组件



01

## SqlSession

作为MyBatis工作的主要顶层API，表示和数据库交互的会话，完成必要数据库增删改查功能。

02

## Executor

MyBatis执行器，是MyBatis调度的核心，负责SQL语句的生成和查询缓存的维护。

03

## StatementHandler

封装了JDBC Statement操作，负责对JDBC statement的操作，如设置参数、将Statement结果集转换成List集合。



# MyBatis核心组件

## ParameterHandler

负责对用户传递的参数转换成JDBC Statement 所需要的参数。

## TypeHandler

用于Java类型和JDBC类型之间的转换。

## ResultSetHandler

负责将JDBC返回的ResultSet结果集对象转换成List类型的集合。

## MappedStatement

一个MappedStatement代表了一个增删改查的SQL语句，包括SQL语句本身、SQL语句的输入参数类型和输出结果的映射关系等。



# MyBatis工作流程

01

## 加载配置并初始化

触发条件有加载配置文件、将SQL的配置信息加载成为一个MappedStatement对象（包括了传入参数映射配置、执行的SQL语句、结果映射配置）、加载映射文件等。

02

## 接收调用请求

触发条件为调用MyBatis提供的API，传入SQL的标识以及传入参数对象。

03

## 处理操作请求

触发条件为根据SQL的标识找到对应的MappedStatement对象，根据传入参数对象解析MappedStatement对象，得到最终要执行的SQL和执行传入参数。

04

## 返回处理结果

触发条件为执行SQL，将所得结果映射成Java对象并返回。



03

# 数据持久层技术原理





# 数据持久层概念及作用

## 数据持久层概念

数据持久层是应用程序中专门负责与数据库进行交互的一层，它封装了对数据库的所有操作，为上层应用提供统一的数据访问接口。

## 数据持久层作用

数据持久层的主要作用是实现数据的持久化存储，将内存中的数据保存到数据库等持久化存储介质中，同时提供数据的查询、更新、删除等操作。通过数据持久层，应用程序可以实现对数据的统一管理和访问，提高数据的安全性和可维护性。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：  
<https://d.book118.com/498121032065006077>