

Please visit website: <http://cxyroad.com>

EasyExcelMysql

=====

` EfficientFarm ` EasyExcelMySQL

excelEasyExcelEasyPOIJXLJXLS

```
| | | |
|---|---|---|
**EasyExcel** | API | Excel (xls) |
**POI** | ApacheEasyPOI XLSXLSX | API |
**Jxls** | Excel | |
```

EasyExcel

```
> 1
> 2excel
```

```
EasyExcel ****
```

```
---
```

```
-----
```

```
**1. User**
```

```
...
```

```
--
```

```
drop table if exists `user`;
```

```
--
```

```
CREATE TABLE `user` (  
  `id` int NOT NULL AUTO INCREMENT COMMENT "  
  `name` varchar(100) COLLATE utf8mb4_general_ci  
  DEFAULT NULL COMMENT "  
  `phone_num` varchar(20) COLLATE utf8mb4_general_ci  
  DEFAULT NULL COMMENT "  
  `address` varchar(200) COLLATE utf8mb4_general_ci  
  DEFAULT NULL COMMENT "  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;
```

```
...
```

```
**2. **
```

![image.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/061d9f81304a49b49a28bcab689c6c40~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2093&h=927&s=134164&e=png&b=f8f5f5)

3. User

2xlsx100xlsxsheet104xlssheet100M

MySQL100wsql

```
...  
DELIMITER //  
drop procedure IF EXISTS InsertTestData;  
CREATE PROCEDURE InsertTestData()  
BEGIN  
    DECLARE counter INT DEFAULT 1;  
  
    WHILE counter < 1000000 DO  
        INSERT INTO user (id, name, phone_num,  
address) VALUES  
        (counter, CONCAT('name_', counter),  
CONCAT('phone_', counter), CONCAT('add_', counter)) ;  
        SET counter = counter + 1;  
    END WHILE;  
END //  
DELIMITER;  
  
--  
CALL InsertTestData();  
...  
  
---
```

SpringBootEasyExcel

5.1 pom.xml

...

```
<!--lombok-->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>>true</optional>
</dependency>
<!--MyBatis Plus-->
<dependency>
  <groupId>com.baomidou</groupId>
  <artifactId>mybatis-plus-boot-starter</artifactId>
  <version>3.4.0</version>
</dependency>
<!--easyexcel-->
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>easyexcel</artifactId>
  <version>3.3.4</version>
</dependency>
<!-- hutool -->
<dependency>
  <groupId>cn.hutool</groupId>
  <artifactId>hutool-all</artifactId>
  <version>5.8.25</version>
</dependency>
```

...

5.2

...

```
@Data
@AllArgsConstructor
@NoArgsConstructor
@ColumnWidth(25)
public class User {
  /**
   *
```

```
*
* @mbg.generated
*/
@ExcelProperty("id")
private Integer id;

/**
*
*
* @mbg.generated
*/
@ExcelProperty("")
private String name;

/**
*
*
* @mbg.generated
*/
@ExcelProperty("")
private String phoneNum;

/**
*
*
* @mbg.generated
*/
@ExcelProperty("")
private String address;
}
...

```

```
* @ExcelProperty
* @ColumnWidth
```

5.3

`UserMapper`

```

...
/**mybatis-plusBaseMapper*
public interface UserMapper extends BaseMapper<User> {
    int deleteByPrimaryKey(Integer id);

    int insertAll(User record);

    void insertSelective(@Param("list") List<User> list);

    User selectByPrimaryKey(Integer id);

    int updateByPrimaryKeySelective(User record);

    int updateByPrimaryKey(User record);
    Integer countNum();
}

```

...

`UserMapper.xml`

...

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD
Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-
mapper.dtd">
<mapper
namespace="org.javaboy.vhr.mapper.UserMapper">
    <resultMap id="BaseResultMap"
type="org.javaboy.vhr.pojo.User">
        <id column="id" jdbcType="INTEGER" property="id" />
        <result column="name" jdbcType="VARCHAR"
property="name" />
        <result column="phone_num" jdbcType="VARCHAR"
property="phoneNum" />
        <result column="address" jdbcType="VARCHAR"
property="address" />
    </resultMap>
    <sql id="Base_Column_List">
        id, name, phone_num, address
    </sql>
    <select id="selectByPrimaryKey"
parameterType="java.lang.Integer"
resultMap="BaseResultMap">
        select
        <include refid="Base_Column_List" />

```

```

    from user
    where id = #{id,jdbcType=INTEGER}
</select>
<select id="countNum" resultType="java.lang.Integer">
    select count(*) from user
</select>
<delete id="deleteByPrimaryKey"
parameterType="java.lang.Integer">
    delete from user
    where id = #{id,jdbcType=INTEGER}
</delete>
<insert id="insertAll" keyColumn="id" keyProperty="id"
parameterType="org.javaboy.vhr.pojo.User"
useGeneratedKeys="true">
    insert into user (name, phone_num, address
    )
    values (#{name,jdbcType=VARCHAR},
#{phoneNum,jdbcType=VARCHAR},
#{address,jdbcType=VARCHAR}
    )
</insert>
<insert id="insertSelective"
parameterType="org.javaboy.vhr.pojo.User">
    insert into user
    (id,name, phone_num, address
    )
    values
    <foreach collection="list" item="item" separator=",">
    (#{item.id},#{item.name},#{item.phoneNum},#{item.address
    })
    </foreach>
</insert>
<update id="updateByPrimaryKeySelective"
parameterType="org.javaboy.vhr.pojo.User">
    update user
    <set>
    <if test="name != null">
    name = #{name,jdbcType=VARCHAR},
    </if>
    <if test="phoneNum != null">
    phone_num = #{phoneNum,jdbcType=VARCHAR},
    </if>
    <if test="address != null">
    address = #{address,jdbcType=VARCHAR},
    </if>
    </set>
    where id = #{id,jdbcType=INTEGER}
</update>

```

```
<update id="updateByPrimaryKey"
parameterType="org.javaboy.vhr.pojo.User">
  update user
  set name = #{name,jdbcType=VARCHAR},
      phone_num = #{phoneNum,jdbcType=VARCHAR},
      address = #{address,jdbcType=VARCHAR}
  where id = #{id,jdbcType=INTEGER}
</update>
</mapper>
```

...

Vue

...

```
<template>
  <el-card>
    <div>
      <!-->
      <el-upload
        :show-file-list="false"
        :before-upload="beforeUpload"
        :on-success="onSuccess"
        :on-error="onError"
        :disabled="importDataDisabled"
        style="display: inline-flex;margin-right: 8px"
        action="/employee/excel/import">
        <!-->
        <el-button :disabled="importDataDisabled"
type="success" :icon="importDataBtnIcon">
          {{importDataBtnText}}
        </el-button>
      </el-upload>
      <el-button type="success"
@click="exportEasyExcel" icon="el-icon-download">
        </el-button>
```



```

        <el-button type="success"
        @click="exportExcelTemplate" icon="el-icon-download">

        </el-button>
    </div>
</el-card>

</template>

<script>
import {Message} from 'element-ui';
export default {
    name: "Export",
    data() {
        return {
            importDataBtnText: "",
            importDataBtnIcon: 'el-icon-upload2',
            importDataDisabled: false,
        }
    },
    methods: {
        onError(res) {
            this.importDataBtnText = "";
            this.importDataBtnIcon = 'el-icon-upload2';
            this.importDataDisabled = false;
            console.log(res);
        },
        onSuccess(res) {
            this.importDataBtnText = "";
            this.importDataBtnIcon = 'el-icon-upload2';
            this.importDataDisabled = false;
            console.log(res.msg);
            if (res.msg == ""){
                Message.success("")
            }
            // this.initEmps();
        },
        beforeUpload() {
            this.importDataBtnText = "";
            this.importDataBtnIcon = 'el-icon-loading';
            this.importDataDisabled = true;
        },
        exportEasyExcel() {
            window.open('/employee/excel/easyexcelexport',
'_parent');
        },
        exportExcelTemplate(){

```

```
        window.open('/employee/excel/exporttemplate',
'_parent');
    }
}
```

</script>

<style scoped>

</style>

...

![[image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/5f31ccd7651a48e79479125aa3a91eb4~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=3103&h=184&s=17157&e=png&b=ffffff)]

7.1

1 ` .xlsx` resource

![[image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/88965ab45f564d13981600a9b29b8193~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1141&h=329&s=19809&e=png&b=3c3f41)]

2

```

...
@RestController
@RequestMapping("/employee/excel")
@AllArgsConstructor
@Slf4j
public class EasyExcelController {
/**
 *
 * @param response
 */
@RequestMapping("/exporttemplate")
public void downloadTemplate(HttpServletRequest response) {
    try {
        //
        InputStream inputStream =
Thread.currentThread().getContextClassLoader().getReso
urceAsStream("template/.xlsx");
        //,encode
        response.setHeader("Content-disposition",
"attachment;fileName=" + java.net.URLEncoder.encode("
.xlsx", "UTF-8"));
        //
        response.setContentType("application/vnd.ms-
excel;charset=UTF-8");
        OutputStream outputStream =
response.getOutputStream();
        byte[] bytes = new byte[2048];
        int len;
        while((len = inputStream.read(bytes)) != -1){
            outputStream.write(bytes,0,len);
        }
        outputStream.flush();
        outputStream.close();
        inputStream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}
...

```

```

`java.net.URLEncoder.encode(".xlsx", "UTF-8")`

```

7.2

1 EasyExcelController

```
...  
@Autowired  
EasyExcelServiceImpl easyExcel;  
  
/**  
 * excel  
 * @param file  
 * @return  
 */  
@RequestMapping("/import")  
public RespBean easyExcelImport(MultipartFile file){  
    if(file.isEmpty()){  
        return RespBean.error("");  
    }  
    easyExcel.easyExcelImport(file);  
    return RespBean.ok("");  
}  
}
```

...

****RespBean****

...

```
public class RespBean {  
    private Integer status;  
    private String msg;  
    private Object obj;  
  
    public static RespBean build() {  
        return new RespBean();  
    }  
  
    public static RespBean ok(String msg) {  
        return new RespBean(200, msg, null);  
    }  
  
    public static RespBean ok(String msg, Object obj) {  
        return new RespBean(200, msg, obj);  
    }  
}
```

```
}  
  
public static RespBean error(String msg) {  
    return new RespBean(500, msg, null);  
}  
  
public static RespBean error(String msg, Object obj) {  
    return new RespBean(500, msg, obj);  
}  
  
private RespBean() {  
  
private RespBean(Integer status, String msg, Object  
obj) {  
    this.status = status;  
    this.msg = msg;  
    this.obj = obj;  
}  
  
public Integer getStatus() {  
    return status;  
}  
  
public RespBean setStatus(Integer status) {  
    this.status = status;  
    return this;  
}  
  
public String getMsg() {  
    return msg;  
}  
  
public RespBean setMsg(String msg) {  
    this.msg = msg;  
    return this;  
}  
  
public Object getObj() {  
    return obj;  
}  
  
public RespBean setObj(Object obj) {  
    this.obj = obj;  
    return this;  
}  
}
```

...

2 easyExcel.easyExcelImport(file)

...

```
@Service
@Slf4j
@AllArgsConstructor
public class EasyExcelServiceImpl implements
EasyExcelService {

    private final ApplicationContext applicationContext;
    /**
     * excle
     * @param file
     */
    @Override
    public void easyExcelImport(MultipartFile file) {
        try {
            long beginTime = System.currentTimeMillis();
            //
            EasyExcelImportHandler listener =
applicationContext.getBean(EasyExcelImportHandler.class
);
            //easyexcelread
            EasyExcel.read(file.getInputStream(),
User.class, listener).sheet().doRead();
            log.info("{}",(System.currentTimeMillis() -
beginTime)/1000);
        } catch (IOException e) {
            log.error("", e.getMessage(), e);
        }
    }
}
```

...

EasyExcelImportHandler

3

...

```

@Slf4j
@Service
public class EasyExcelImportHandler implements
ReadListener<User> {
    /**/
    private final CopyOnWriteArrayList<User> successList =
new CopyOnWriteArrayList<>();
    /**/
    private final static int BATCH_COUNT = 20000;
    @Resource
    private ThreadPoolExecutor threadPoolExecutor;
    @Resource
    private UserMapper userMapper;

    @Override
    public void invoke(User user, AnalysisContext
analysisContext) {
        if(StringUtils.isNotBlank(user.getName())){
            successList.add(user);
            return;
        }
        if(successList.size() >= BATCH_COUNT){
            log.info("{} ", successList.size());
            saveData();
        }
    }

    /**
     *
     */
    private void saveData() {
        List<List<User>> lists = ListUtil.split(successList,
20000);
        CountDownLatch countDownLatch = new
CountDownLatch(lists.size());
        for (List<User> list : lists) {
            threadPoolExecutor.execute()->{
                try {
                    userMapper.insertSelective(list.stream().map(o -> {
                        User user = new User();
                        user.setName(o.getName());
                        user.setId(o.getId());
                        user.setPhoneNum(o.getPhoneNum());
                        user.setAddress(o.getAddress());
                        return user;
                    }));
                } catch (Exception e) {
                    log.error("saveData error: ", e);
                }
            }
        }
    }
}

```

```

        }).collect(Collectors.toList());
    } catch (Exception e) {
        log.error("e:{}", e.getMessage(), e);
    } finally {
        //1,
        countdownLatch.countDown();
    }
});
}
//
try {
    countdownLatch.await();
} catch (Exception e) {
    log.error("e:{}", e.getMessage(), e);
}
//
successList.clear();
lists.clear();
}
/**
 *
 * @param analysisContext
 */
@Override
public void doAfterAllAnalysed(AnalysisContext
analysisContext) {
    //
    if(CollectionUtils.isNotEmpty(successList)){
        log.info("{} ", successList.size());
        saveData();
    }
}
}
}
...

```

EasyExcelReadListener

```

...
@Configuration
public class EasyExcelThreadPoolExecutor {

```



```
@Bean(name = "threadPoolExecutor")
public ThreadPoolExecutor
easyExcelStudentImportThreadPool() {
    //
    int processors =
Runtime.getRuntime().availableProcessors();
    return new ThreadPoolExecutor(processors + 1,
        processors * 2 + 1,
        10 * 60,
        TimeUnit.SECONDS,
        new LinkedBlockingQueue<>(1000000));
}
}
...

```

ReadListener

```
* invoke()
* doAfterAllAnalysed()
* invokeHead()
* onException()

```

4 100

100excelapplication.yml

```
...
spring:
  servlet:
    multipart:
      max-file-size: 128MB # 10MB
      max-request-size: 128MB # 100MB
...

```

1005008****

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/0e37cf7213d1417eaa40be852f93d58d~tplv-k3u1fbpfcp-ij-mark:3024:0:0:0:q75.awebp#?w=2339&h=296&s=80888&

e=png&b=2b2b2b)

7.3

1 EasyExcelController

```
...
/**
 * excel
 * @param response
 */
@RequestMapping("/easyexcelexport")
public void easyExcelExport(HttpServletRequest response)
response){
    try {
        //
        response.setContentType("text/csv");
        //
        response.setCharacterEncoding("utf-8");
        //
        String fileName = URLEncoder.encode("", "UTF-
8");
        //
        response.setHeader("Content-disposition",
"attachment;filename="+ fileName + ".xlsx");
        Integer total = userMapper.countNum();
        if (total == 0) {
            log.info("");
            return;
        }
        //class
        ExcelWriter build =
EasyExcel.write(response.getOutputStream(),
User.class).build();
        //sheet
        WriteSheet writeSheet =
EasyExcel.writerSheet("sheet").build();
        long pageSize = 10000;
        long pages = total / pageSize;
        long startTime = System.currentTimeMillis();
        //
        if(pages < 1){
            List<User> users = userMapper.selectList(null);
            build.write(users, writeSheet);
        }
    }
}
```

```

    }
    //
    for (int i = 0; i <= pages; i++) {
        Page<User> page = new Page<>();
        page.setCurrent(i + 1);
        page.setSize(pageSize);
        Page<User> userPage =
userMapper.selectPage(page, null);
        build.write(userPage.getRecords(), writeSheet);
    }
    build.finish();
    log.info("/ms:"+(System.currentTimeMillis()-
startTime)+" "+total);
} catch (Exception e) {
    log.error("easyExcele:{}",e.getMessage(),e);
}
}
}

```

...

sheetxlsheet

Mybatis-Plus

```

...
@Configuration
public class MybatisPlusPageConfig {
    /**
     *
     */
    @Bean
    public MybatisPlusInterceptor mybatisPlusInterceptor() {
        MybatisPlusInterceptor mybatisPlusInterceptor = new
MybatisPlusInterceptor();
        mybatisPlusInterceptor.addInnerInterceptor(new
PaginationInnerInterceptor());
        return mybatisPlusInterceptor;
    }
}

```

...

10040

![[image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/8740d6a264f8423eba75c6ede3f32dfd~tplv-k3u1fbpfcp-ij-mark:3024:0:0:0:q75.awebp#?w=2111&h=214&s=56164&e=png&b=2b2b2b)]

SpringBootEasyExcel

![[image.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/5935e2ae310440f2a45235f33006da89~tplv-k3u1fbpfcp-ij-mark:3024:0:0:0:q75.awebp#?w=2111&h=214&s=56164&e=png&b=2b2b2b)]

: <https://juejin.cn/post/7366839695769124874>