AI 掘金头条–收藏模块

检查新闻收藏状态、添加和取消收藏、收藏列表、清空收藏列表



返回 新闻详情

我国成功发射通信技术试验卫星十一号

央视新闻 2024-01-10T18:20:00 11203 阅读



2024年1月10日15时03分，我国在西昌卫星发射中心使用长征二号丁运载火箭，成功将通信技术试验卫星十一号发射升空。该卫星主要用于开展Ka频段宽带通信、新型天线展开等关键技术验证，可为下一代通信技术积累数据。此次任务是长征系列火箭的第456次飞行。

返回 我的收藏 清空



我国科学家在量子计算领域取得新突破

科技日报 2024-03-01T11:20:00

收藏时间: 2026/1/16 09:55:41



央行宣布降准0.5个百分点

财经网 2024-01-20T16:45:00

收藏时间: 2026/1/16 09:55:35



我国成功发射通信技术试验卫星十一号

央视新闻 2024-01-10T18:20:00

收藏时间: 2026/1/16 09:54:37



国家主席发表2024年新年贺词

新华社 2024-01-01T08:00:00

收藏时间: 2026/1/16 09:54:27

接口实现流程



- ✓ 定义 APIRouter 实例
- ✓ 注册路由 include_router()
- ✓ 参照接口规范文档




```
class Base(DeclarativeBase):
  pass

class Category(Base):
  __tablename__="news_category"
  ......
```

- ✓ 参照数据库表



- ✓ select(模型类)
- ✓ add()
- ✓ update()
- ✓ delete()



- ✓ Depends 注入数据库依赖
- ✓ 调用查询逻辑
- ✓ 响应结果

检查新闻收藏状态

1. 检查新闻收藏状态

- **接口地址**: `GET /api/favorite/check`
- **请求头**: 需要认证
- **请求参数**:

参数名	类型	必填	说明
newsId	integer	是	新闻ID

- **请求示例**:

```
GET /api/favorite/check?newsId=1
```

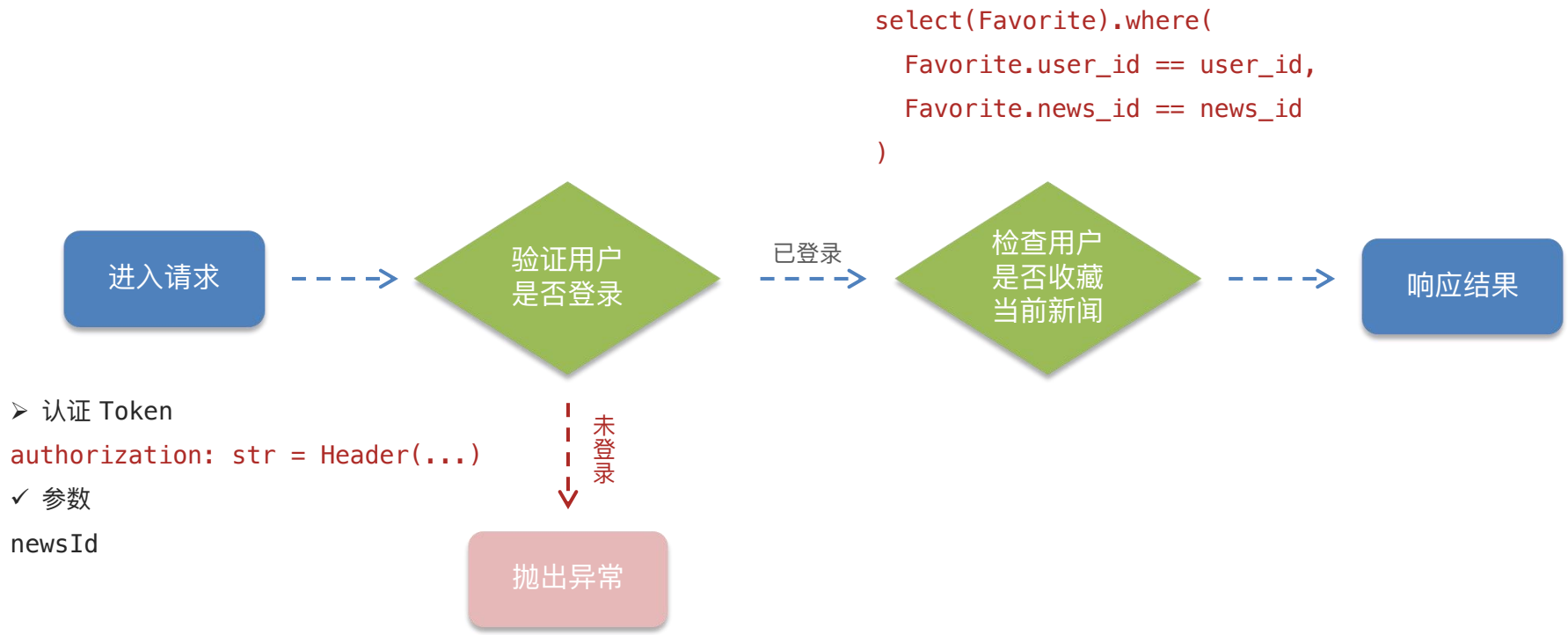
- **响应示例**:

```
{
  "code": 200,
  "message": "success",
  "data": {
    "isFavorite": true
  }
}
```

```python
router = APIRouter(prefix="/api/favorite", tags=["favorite"])

@router.get("/check")
async def check_favorite(
                    news_id: int = Query(..., alias="newsId"),
                    user: User = Depends(get_current_user),
                    db: AsyncSession = Depends(get_db)
):
  return success_response(data=...)
```

检查新闻收藏状态




```
select(Favorite).where(
  Favorite.user_id == user_id,
  Favorite.news_id == news_id
)
```

- 认证 Token

```
authorization: str = Header(...)
```

- ✓ 参数

```
newsId
```

添加收藏

2. 添加收藏

- **接口地址**: `POST /api/favorite/add`
- **请求头**: 需要认证
- **请求参数**:

参数名	类型	必填	说明
newsId	integer	是	新闻ID

- **请求示例**:

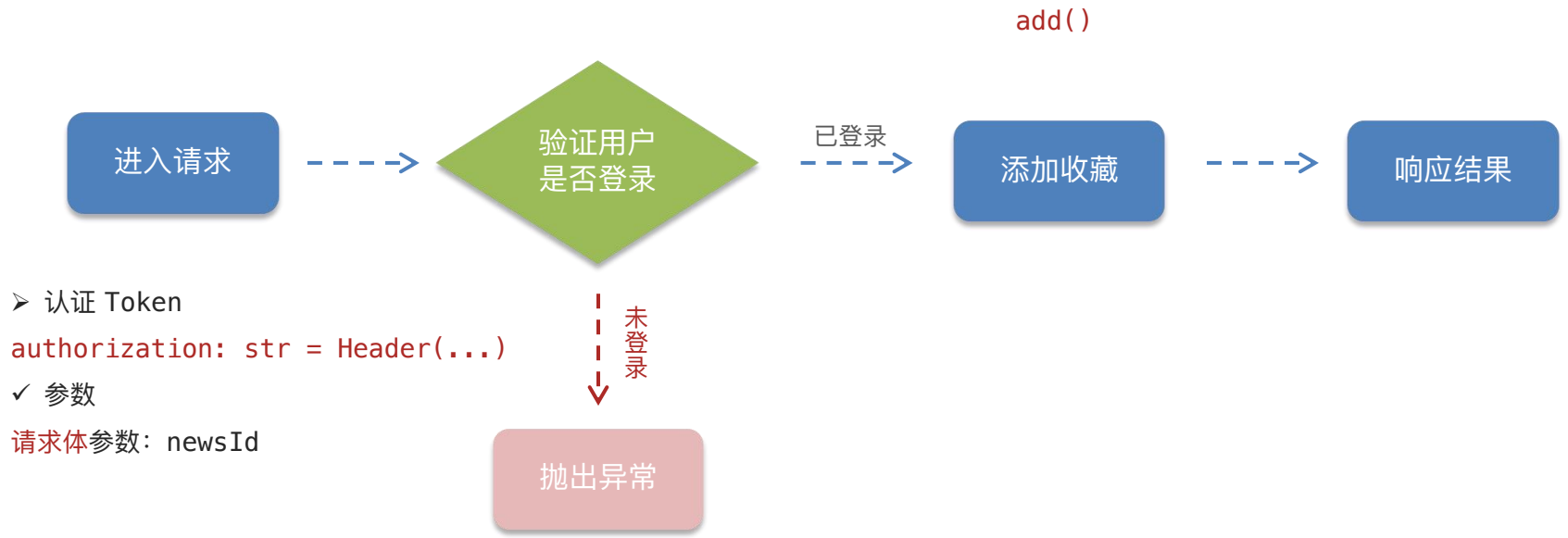
```
{
  "newsId": 1
}
```

- **响应示例**:

```
{
  "code": 200,
  "message": "收藏成功",
  "data": {
    "id": 1,
    "userId": 1,
    "newsId": 1,
    "createTime": "2023-01-01T00:00:00"
  }
}
```

```python
@router.post("/add")
async def add_favorite(
                  data: FavoriteAddRequest,
                  user: User = Depends(get_current_user),
                  db: AsyncSession = Depends(get_db)):
  return success_response(message="收藏成功", data=...)
```

添加收藏



- 认证 Token

```
authorization: str = Header(...)
```

- ✓ 参数

请求体参数：newsId

取消收藏

3. 取消收藏

- **接口地址**: `DELETE /api/favorite/remove`
- **请求头**: 需要认证
- **请求参数**:

参数名	类型	必填	说明
newsId	integer	是	新闻ID

- **请求示例**:

```
DELETE /api/favorite/remove?
newsId=1
```

- **响应示例**:

```
{
  "code": 200,
  "message": "取消收藏成功",
  "data": null
}
```

```python
@router.delete("/remove")
async def remove_favorite(
                    news_id: int = Query(..., alias="newsId"),
                    user: User = Depends(get_current_user),
                    db: AsyncSession = Depends(get_db)):
  return success_response(message="取消收藏成功")
```

取消收藏



- 认证 Token

```
authorization: str = Header(...)
```

- ✓ 参数

```
newsId
```

获取收藏列表

4. 获取收藏列表

- **接口地址**: `GET /api/favorite/list`
- **请求头**: 需要认证
- **请求参数**:

参数名	类型	必填	说明
page	integer	否	页码，默认为1
pageSize	integer	否	每页条数，默认为10，最大值为100

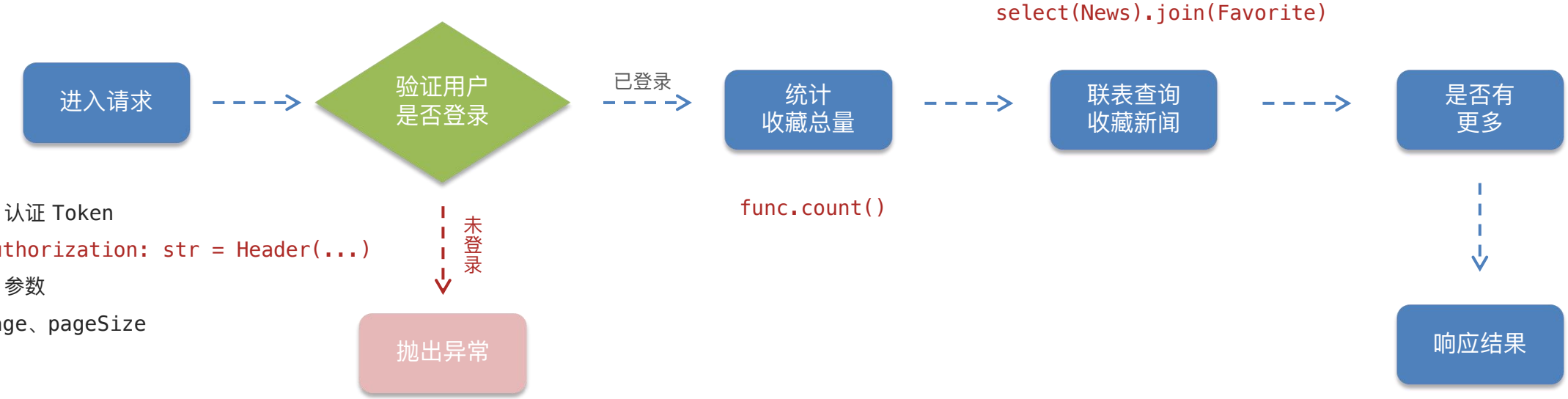
- **响应示例**:

```
{
  "code": 200,
  "message": "success",
  "data": {
    "list": [
      {
        "id": 1,
        "title": "新闻标题",
        "description": "",
        "image": "",
        "author": "",
        "publishTime": "2023-01-01T00:00:00",
        "categoryId": 1,
        "views": 1,
        "favoriteTime": "2023-01-01T00:00:00"
      }
    ],
    "total": 1,
    "hasMore": false
  }
}
```

```python
@router.get("/list")
async def get_favorite_list(
                    page: int = Query(1, ge=1),
                    page_size: int = Query(10, ge=1, le=100, alias="pageSize"),
                    user: User = Depends(get_current_user),
                    db: AsyncSession = Depends(get_db)):


  return success_response(data=...)
```

获取收藏列表



- 认证 Token

```
authorization: str = Header(...)
```

- ✓ 参数

page、pageSize

```
func.count()
```

清空收藏列表

5. 清空所有收藏

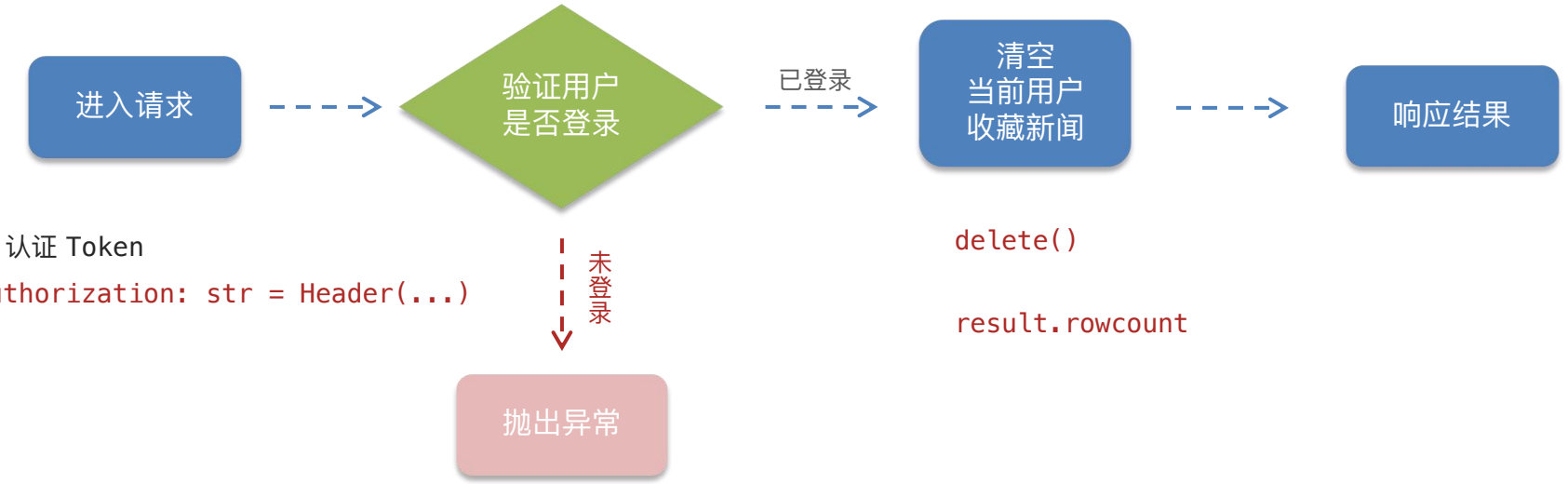
- **接口地址**: `DELETE /api/favorite/clear`
- **请求头**: 需要认证
- **响应示例**:

```
{
  "code": 200,
  "message": "成功删除1条收藏记录",
  "data": null
}
```

```python
@router.delete("/clear")
async def clear_favorite(
            user: User = Depends(get_current_user),
            db: AsyncSession = Depends(get_db)):

return success_response(message=f"清空{xx}条收藏记录")
```

清空收藏列表



- 认证 Token

```
authorization: str = Header(...)
```

```
delete()

result.rowcount
```

本章总结



返回 新闻详情

我国成功发射通信技术试验卫星十一号

央视新闻 2024-01-10T18:20:00 11203 阅读



2024年1月10日15时03分，我国在西昌卫星发射中心使用长征二号丁运载火箭，成功将通信技术试验卫星十一号发射升空。该卫星主要用于开展Ka频段宽带通信、新型天线展开等关键技术验证，可为下一代通信技术积累数据。此次任务是长征系列火箭的第456次飞行。

返回 我的收藏 清空



我国科学家在量子计算领域取得新突破

科技日报 2024-03-01T11:20:00

收藏时间: 2026/1/16 09:55:41



央行宣布降准0.5个百分点

财经网 2024-01-20T16:45:00

收藏时间: 2026/1/16 09:55:35



我国成功发射通信技术试验卫星十一号

央视新闻 2024-01-10T18:20:00

收藏时间: 2026/1/16 09:54:37



国家主席发表2024年新年贺词

新华社 2024-01-01T08:00:00

收藏时间: 2026/1/16 09:54:27

本章总结

- ORM 联表查询的写法

  `select(` 查询的主体模型类, 字段重命名 `).join(` jion 的模型类, join 的条件`)`

```
select(
      News,
      Favorite.created_at.label("favorite_time"),
      Favorite.id.label("favorite_id")
)
.join(Favorite, Favorite.news_id == News.id)
```

- 模型类中，`UniqueConstraint` 的作用

  唯一约束

```
class Favorite(Base):
  # 创建索引
  __table_args__ = (
    UniqueConstraint('user_id', 'news_id', name='user_news_unique')
    ......
  )
```

接口实现流程



- ✓ 定义 APIRouter 实例
- ✓ 注册路由 include_router()
- ✓ 参照接口规范文档




```
class Base(DeclarativeBase):
  pass

class Category(Base):
  __tablename__="news_category"
  ......
```

- ✓ 参照数据库表



- ✓ select(模型类)
- ✓ add()
- ✓ update()
- ✓ delete()

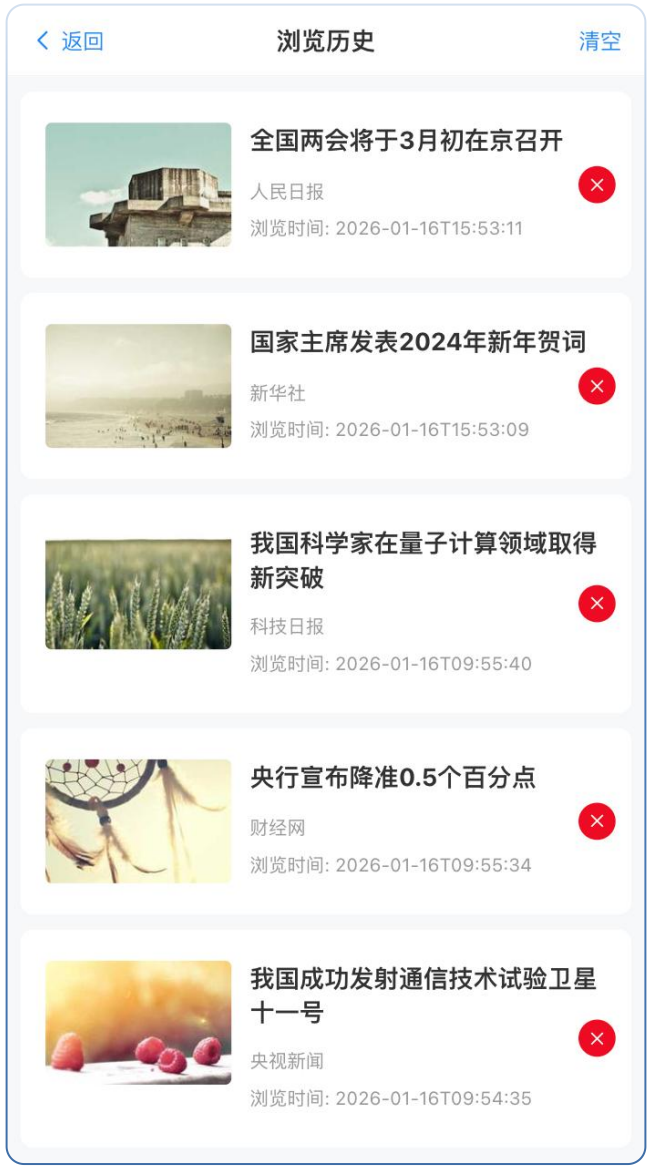


- ✓ Depends 注入数据库依赖
- ✓ 调用查询逻辑
- ✓ 响应结果

AI 掘金头条–浏览历史

添加浏览记录、获取浏览历史列表、删除单条记录、清空浏览历史




返回
浏览历史
清空
全国两会将于3月初在京召开
人民日报
浏览时间: 2026-01-16T15:53:11
国家主席发表2024年新年贺词
新华社
浏览时间: 2026-01-16T15:53:09
我国科学家在量子计算领域取得新突破
科技日报
浏览时间: 2026-01-16T09:55:40
央行宣布降准0.5个百分点
财经网
浏览时间: 2026-01-16T09:55:34
我国成功发射通信技术试验卫星十一号
央视新闻
浏览时间: 2026-01-16T09:54:35

收藏 VS 浏览历史

收藏

添加收藏
取消收藏
获取收藏列表
清空收藏列表

浏览历史

添加浏览记录
删除单条浏览记录
获取浏览历史列表
清空浏览历史列表


爆肝敲代码

单击此处添加标题



示例代码　　**讲义**　　**思路分析视频**　　**素材代码片段**

添加浏览记录

1. 添加浏览记录

- **接口地址**: `POST /api/history/add`
- **请求头**: 需要认证
- **请求参数**:

参数名	类型	必填	说明
newsId	integer	是	新闻ID

- **请求示例**:

```
{
  "newsId": 1
}
```

- **响应示例**:

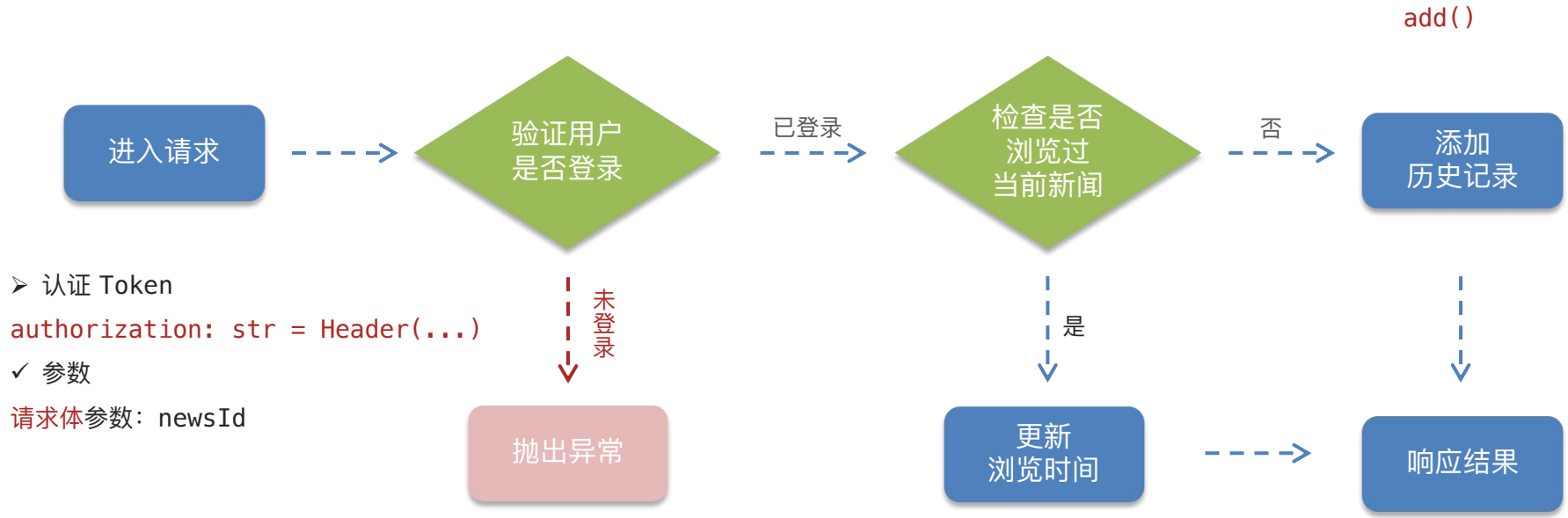
```
{
  "code": 200,
  "message": "添加成功",
  "data": {
    "id": 1,
    "userId": 1,
    "newsId": 1,
    "viewTime": "2023-01-01T00:00:00"
  }
}
```

```python
router = APIRouter(prefix="/api/history", tags=["history"])

@router.post("/add")
async def add_history(
            data: HistoryAddRequest,
            user: User = Depends(get_current_user),
            db: AsyncSession = Depends(get_db)):

  return success_response(message="添加成功", data=xx)
```

添加浏览记录



- 认证 Token

```
authorization: str = Header(...)
```

- ✓ 参数

请求体参数：newsId

获取浏览历史列表

2. 获取浏览历史列表

- **接口地址**: `GET /api/history/list`
- **请求头**: 需要认证
- **请求参数**:

参数名	类型	必填	说明
page	integer	否	页码，默认为1
pageSize	integer	否	每页条数，默认为10，最大值为100

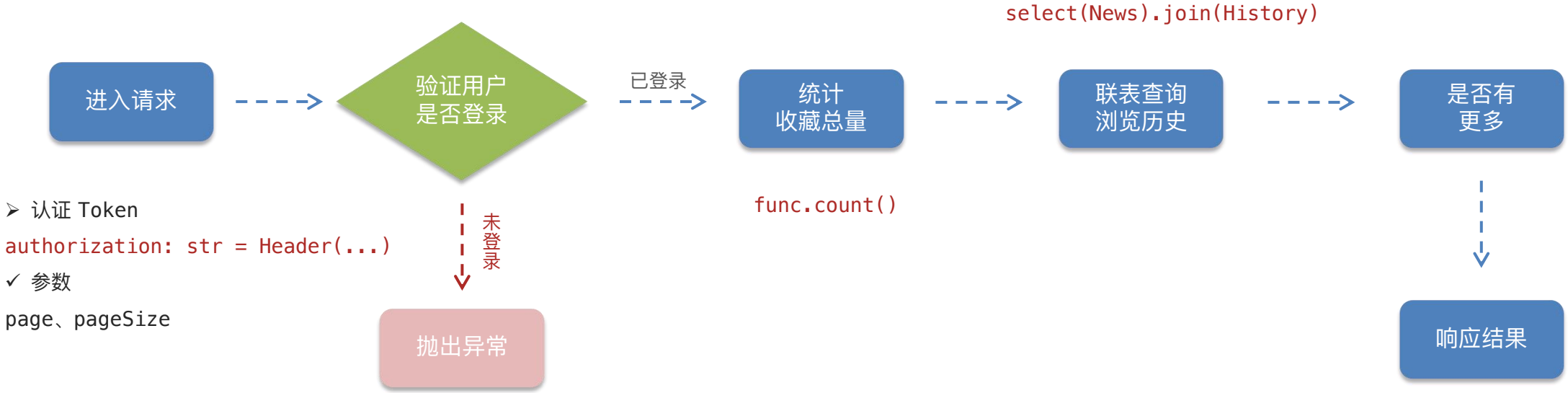
- **响应示例**:

```
{
  "code": 200,
  "message": "success",
  "data": {
    "list": [
      {
        "id": 1,
        "title": "新闻标题",
        "description": "",
        "image": "",
        "author": "",
        "publishTime": "2023-01-01T00:00:00",
        "categoryId": 1,
        "views": 1,
        "viewTime": "2023-01-01T00:00:00"
      }
    ],
    "total": 1,
    "hasMore": false
  }
}
```

```python
@router.get("/list")
async def get_history_list(
                    page: int = Query(1, ge=1),
                    page_size: int = Query(
                          10, ge=1, le=100, alias="pageSize"),
                    user: User = Depends(get_current_user),
                    db: AsyncSession = Depends(get_db)):

  return success_response(data=result)
```

获取浏览历史列表



- 认证 Token

`authorization: str = Header(...)`

- ✓ 参数

page、pageSize

删除单条浏览记录

3. 删除单条浏览记录

- **接口地址**: `DELETE /api/history/delete/{history_id}`
- **请求头**: 需要认证
- **路径参数**:

参数名	类型	必填	说明
history_id	integer	是	历史记录ID

- **请求示例**:

```
DELETE /api/history/delete/1
```

- **响应示例**:

```
{
  "code": 200,
  "message": "删除成功",
  "data": null
}
```

```python
@router.delete("/delete/{history_id}")
async def delete_history(
                    history_id: int,
                    user: User = Depends(get_current_user),
                    db: AsyncSession = Depends(get_db)):

  return success_response(message="删除成功")
```

删除单条浏览记录



- 认证 Token

```
authorization: str = Header(...)
```

- ✓ 参数

```
history_id
```

清空浏览历史列表

4. 清空浏览历史

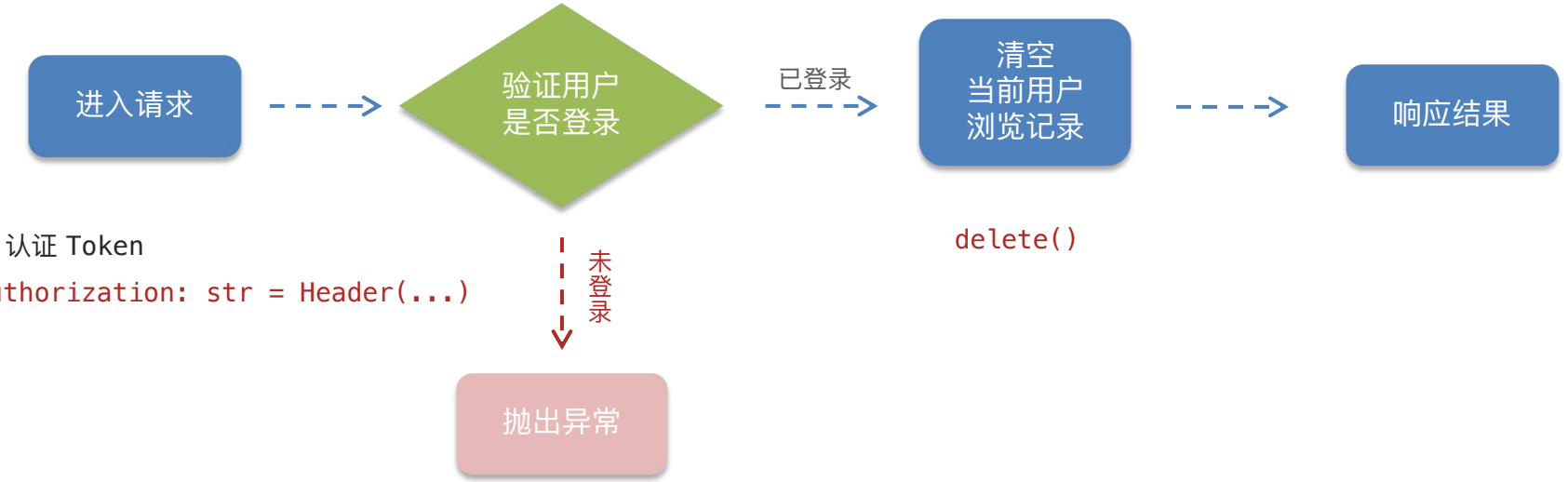
- **接口地址**: `DELETE /api/history/clear`
- **请求头**: 需要认证
- **响应示例**:

```
{
  "code": 200,
  "message": "清空成功",
  "data": null
}
```

```python
@router.delete("/clear")
async def clear_history(
            user: User = Depends(get_current_user),
            db: AsyncSession =Depends(get_db)):

return success_response(message="清空成功")
```

清空浏览历史列表



- 认证 Token

```
authorization: str = Header(...)
```

```
delete()
```



传智教育旗下高端IT教育品牌

