

服务地址：http://<server_ip>:8000

数据格式：application/json

返回格式：统一 JSON

```
{ "status": "success", "data": {...}, "message": "" }
```

1. 查询接口 1.1 查询节点

URL:

GET /query_node

参数 (Query Params) :

label (string) 节点标签，例如 Person

key (string) 属性名，例如 name

value (string) 属性值，例如 Alice

请求示例:

GET /query_node?label=Person&key=name&value=Alice

响应示例:

```
{ "status": "success", "data": [ { "id": 1, "name": "Alice", "age": 30 } ] }
```

1.2 查询边

URL:

GET /query_edge

参数:

type (string) 边的类型，例如 KNOWS

(可选) 其他属性过滤，例如 since=2010

请求示例:

GET /query_edge?type=KNOWS&since=2010

响应示例:

```
{ "status": "success", "data": [ { "id": 100, "start": 1, "end": 2, "type": "KNOWS", "since": 2010 } ] }
```

2. 新增接口 2.1 新增节点

URL:

POST /add_node

请求体:

```
{ "label": "Person", "props": { "name": "Bob", "age": 25 } }
```

响应:

```
{ "status": "success", "data": { "id": 2, "message": "Node created" } }
```

2.2 新增边

URL:

POST /add_edge

请求体:

```
{ "start_id": 1, "end_id": 2, "rel_type": "KNOWS", "props": { "since": 2015 } }
```

响应:

```
{ "status": "success", "data": { "id": 101, "message": "Relationship created" } }
```

3. 更新接口 3.1 更新节点属性

URL:

PUT /update_node

请求体:

```
{ "id": 1, "props": { "age": 31 } }
```

响应:

```
{ "status": "success", "message": "Node updated" }
```

3.2 更新边属性

URL:

PUT /update_edge

请求体:

```
{ "id": 101, "props": { "since": 2020 } }
```

响应:

```
{ "status": "success", "message": "Relationship updated" }
```

4. 删除接口 4.1 删除节点

URL:

DELETE /delete_node

参数 (Query Params) :

id (int) 节点 ID

请求示例:

```
DELETE /delete_node?id=1
```

响应:

```
{ "status": "success", "message": "Node deleted" }
```

4.2 删除边

URL:

```
DELETE /delete_edge
```

参数 (Query Params) :

id (int) 边 ID

请求示例:

```
DELETE /delete_edge?id=101
```

响应:

```
{ "status": "success", "message": "Relationship deleted" }
```

5. 额外功能 (可选加分项) 5.1 路径查询 (两节点之间的最短路径)

URL:

```
GET /shortest_path
```

参数:

start_id (int)

end_id (int)

响应示例:

```
{ "status": "success", "data": { "nodes": [1, 5, 8, 2], "edges": [101, 203, 305] } }
```