1.3 User Profile

Provides user login authentication API to verify username and password, issue Token and return basic user information for API access authentication and user identity recognition.

Develop the following interfaces based on Database Table【Users】

1.3.1 User Login to Obtain Access Token: POST /api/auth/login

Parameter Name	Type	Required	Description
username	String	Yes	User name
password	String	Yes	Password

·Headers:

Content-Type: application/json

·Request Example:

```
{
  "username": "admin",
  "password": "123456"
}
```

·Response requirements:

```
{
    "code": 200,
    "msg": "success",
    "data": {
        "token": "eyJhbGciOiJIU....",
        "expiresIn": 3600,
        "user": {
            "userId": 1,
            "username": "admin",
            "nickName": "Admin",
            "email": "admin@example.com",
            "phone": "13800138000",
            "avatar": "/upload/20260602/639160099670709379.png",
            "createTime": "2026-05-11 15:14:12"
        }
    }
}
```

1.3.2 Get Current Logged-in User Info: GET /api/auth/info

·Headers:

Authorization: Token -- Required. Used for identity verification when calling the API after user login via authentication center.

·Response requirements:


```
{
  "code": 200,
  "msg": "success",
  "data": {
    "user": {
      "userId": 1,
      "username": "admin",
      "nickName": "Admin",
      "email": "admin@example.com",
      "phone": "13800138000",
      "avatar": "/upload/20260602/639160099670709379.png",
      "createTime": "2026-05-11 15:14:12"
    }
  }
}
```


1.3.3 User logout: POST /api/auth/logout

·Headers:

Authorization: Token -- Required. Used for identity verification when calling the API after user login via authentication center.

Content-Type: application/json

·Response requirements:


```
{
    "code": 200,
    "msg": "success",
    "data": "Exit successful"
}
```


1.3.4 Check Current Token Validity：GET /api/auth/check

·Headers:

Authorization: Token -- Required. Used for identity verification when calling the API after user login via authentication center.

© WorldSkills International

·Response requirements:

```
//example: sucess
{
    "code": 200,
    "msg": "success",
    "data": "success"
}
//example: failure
{
    "code": 401,
    "msg": "The token is invalid or expired"
}
```

1.4 Real-time Traffic Status Query

This API provides paginated query for congested road segments. It returns data including the start and end coordinates, addresses, congestion levels, congestion lengths, and estimated delay times of congested segments, providing data support for navigation route planning and real-time traffic analysis.

Develop the following interface based on Database Table【Congestions】

1.4.1 Query the list of congested road segments: GET /system/api/congestion/list

Parameter Name	Type	Required	Description
pageNum	Integer	No	Page number
pageSize	Integer	No	Number of items per page
roadName	String	No	Road name
startLat	Double	Yes	Starting latitude
startLng	Double	Yes	Starting longitude
endLat	Double	Yes	Ending latitude
endLng	Double	Yes	Ending longitude

© WorldSkills International

·Response requirements:


```
{
  "code": 200,
  "msg": "success",
  "data": {
    "total": 10,
    "rows": [{
            "congestionId": 26,
            "roadName": "Central Ave",
            "startLatitude": "39.1500",
            "startLongitude": "-75.4500",
            "startAddress": "Dover Central Ave",
            "endLatitude": "39.1600",
            "endLongitude": "-75.4400",
            "endAddress": "Dover Capitol",
            "congestionLevel": "severe",
            "congestionLength": "900",
            "delayTime": "15 minutes"
        }]
  }
}
```


1.5 Route Planning

This API conducts route planning using the input latitude and longitude coordinates of the start and end points. It returns the distance, travel duration of the destination, and the collection of latitude-longitude points for the path between the two points.

Please refer to the **Map Developer Guide for Session1** to finish this function.

1.5.1 Route Planning: POST /map_test/way

Parameter Name	Type	Required	Description
pathWay	Double[][]	Yes	Array of coordinates, where each element is [longitude, latitude]
mode	String	Yes	Travel mode: driving, cycling, walking

© WorldSkills International

·Headers:

Content-Type: application/json;

·Request Example:

```
{
    "pathWay": [
            [-75.44354248046875, 39.155181884765625],
            [-75.46984100341797, 39.16850280761719]
    ],
    "mode": "cycling"
}
```

·Response requirements:

```
{
    "code": 200,
    "distance": "7942m",
  "msg": "success",
    "time": "1906151ms",
    "path": [
            [-75.44354248046875,39.155181884765625],
            [-75.46910095214844, 39.1687126159668],
            [-75.46984100341797, 39.16850280761719]

    ],
    "startName": "Bayside Drive",
    "endName": "North Little Creek Road",
    "mode": "cycling"
}
```


© WorldSkills International