

# RiskHunter API

**Description:** RiskHunter API

**HOST:** localhost:8080

**Contacts:** SparkNJU

**Version:** v1

**Time:** 2025.3.24

**URL:** /v2/api-docs

**Online Access:** Start the RiskHunter backend locally and visit <http://localhost:8080/doc.html#/home> (**Recommended**) or access the online [RiskHunter API Documentation](#).

- [RiskHunter API](#)
- ▼ [Smart Assistant related Interfaces](#)
  - [Get Chat History](#)
  - [Non-Streaming Chat](#)
  - [RAG Streaming Dialogue Interface](#)
  - [RAG Knowledge Base Search Interface](#)
  - [Create Session Interface](#)
  - [Delete Session Interface](#)
  - [Update Session Title](#)
  - [Get All Session IDs of a User](#)
  - [Streaming Chat Interface](#)
- ▼ [User related Interfaces](#)
  - [Get User Information](#)
  - [Update User Information](#)
  - [User Login](#)
  - [User Registration](#)
- ▼ [Risk Signal related Interfaces](#)
  - [Paged Query Risk Signals](#)
  - [Create Risk Signal](#)
  - [Advanced Query Risk Signals](#)
  - [Update Risk Signal](#)
  - [Delete Risk Signal](#)
- ▼ [Risk Cauculation related Interfaces](#)
  - [Get Alert Information](#)
  - [Get Risk Dashboard Information](#)
  - [Get Exposure Analysis Information](#)

- [Get Risk Map Information](#)

# Smart Assistant related Interfaces

## Get Chat History

**url:** /api/chat/history/{sessionId}

**method:** GET

**produces:** application/x-www-form-urlencoded

**consumes:** \*/\*

### Description:

Retrieve chat history for a specified session.

### Params:

Parameter	Description	In	Required	Type	Schema
sessionId	Session ID	path	false	integer(int64)	
userId	User ID	query	false	integer(int64)	

### Status:

Code	Description	Schema
200	OK	ResultVO«List«ChatRecordModel»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

### Response Params:

Parameter	Description	Type	Schema
code	Status code	string	
msg	Message	string	
result	Chat records	array	ChatRecordModel

Parameter	Description	Type	Schema
content	Message content	string	
createTime	Creation time	string(date-time)	
direction	Message direction	boolean	
id	Record ID	integer(int64)	
sessionId	Session ID	integer(int64)	
userId	User ID	integer(int64)	

**Response Example:**

```
{
  "code": "000",
  "msg": null,
  "result": [
    {
      "id": 9,
      "sessionId": 3,
      "userId": 2,
      "direction": true,
      "content": "What is Nanjing University?",
      "createTime": "2025-03-10T10:15:20"
    },
    {
      "id": 10,
      "sessionId": 3,
      "userId": 2,
      "direction": false,
      "content": "<thought>...</thought>Nanjing University (NJU) is one of China's top comprehensive resear",
      "createTime": "2025-03-10T10:15:49"
    }
  ]
}
```

# Non-Streaming Chat

**url:** /api/chat/noStream

**method:** POST

**produces:** application/json

**consumes:** \*/\*

**Note:**

Chat via non-streaming mode.

**Example:**

```
{
  "message": "hello",
  "sessionId": 2,
  "userId": 1
}
```

**Params:**

name	description	in	require	type	schema
chatRequestDTO	Chat request DTO	body	false	ChatRequestDTO	ChatRequestDTO
message			false	string	
sessionId			false	integer(int64)	
userId			false	integer(int64)	

**Status:**

code	description	schema
200	OK	ResultVO«string»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

**Response Params:**

name	description	type	schema
code		string	
msg		string	
result		string	

**Response Example:**

```
{
  "code": "000",
  "msg": null,
  "result": "Hello! Is there anything I can help you with? Whether it's academic questions, data analysis,
}
```

# RAG Streaming Dialogue Interface

**url:** /api/chat/ragChat

**method:** GET

**produces:** application/x-www-form-urlencoded

**consumes:** \*/\*

**Note:**

Conduct a dialogue in a streaming manner via RAG.

**Params:**

name	description	in	require	type	schema
message	Message content	query	false	string	
sessionId	Session ID	query	false	integer(int64)	
userId	User ID	query	false	integer(int64)	

**Status:**

code	description	schema
200	OK	Flux«ServerSentEvent«string»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

**Response Params:**

name	description	type	schema
prefetch		integer(int32)	integer(int32)

Response Example:

```
data: Hello,  
data: I am  
data: RiskHunter AI...
```

RAG Knowledge Base Search Interface

url: /api/chat/ragSearch

method: POST

produces: application/json

consumes: \*/\*

Note:

Search the knowledge base via RAG.

Example:

```
{  
  "message": "Search for the main methods of foreign exchange risk management",  
  "sessionId": 1,  
  "userId": 1  
}
```

Params:

name	description	in	require	type	schema
chatRequestDTO	Chat request DTO	body	false	ChatRequestDTO	ChatRequestDTO
message			false	string	
sessionId			false	integer(int64)	
userId			false	integer(int64)	

Status:

code	description	schema
200	OK	ResultVO«string»
201	Created	

code	description	schema
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code		string	
msg		string	
result		string	

Response Example:

```
{
  "code": "000",
  "msg": "",
  "result": "The main methods of foreign exchange risk management include the following: Currency hedging"
}
```

# Create Session Interface

url: /api/chat/session

method: POST

produces: application/json

consumes: \*/\*

Note:

Create a new chat session.

Params:

name	description	in	require	type	schema
userId	Unique identifier of the user (must be greater than 0)	query	false	integer(int64)	

Status:

code	description	schema
200	OK	ResultVO«long»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code		string	
msg		string	
result		integer(int64)	integer(int64)

Response Example:

```
{
  "code": "000",
  "msg": null,
  "result": 6
}
```

## Delete Session Interface

url: /api/chat/session/{sessionId}

method: DELETE

produces: application/x-www-form-urlencoded

consumes: \*/\*

Note:

Delete session by id

Params:

name	description	in	require	type	schema
sessionId	session ID	path	false	integer(int64)	



name	description	in	require	type	schema
userId	user ID	query	false	integer(int64)	

Status:

code	description	schema
200	OK	ResultVO«boolean»
204	No Content	
401	Unauthorized	
403	Forbidden	

Response Params:

name	description	type	schema
code		string	
msg		string	
result		boolean	

Response Example:

```
{
  "code": "",
  "msg": "",
  "result": true
}
```

# Update Session Title

url: /api/chat/session/{sessionId}/title

method: PUT

produces: application/json

consumes: \*/\*

Note:

Update the title of the specified session.

**Params:**

name	description	in	require	type	schema
sessionId	Session ID	path	false	integer(int64)	
title	New title	query	false	string	
userId	User ID	query	false	integer(int64)	

**Status:**

code	description	schema
200	OK	ResultVO«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

**Response Params:**

name	description	type	schema
code		string	
msg		string	
result		boolean	

**Response Example:**

```
{
  "code": "000",
  "msg": null,
  "result": true
}
```

# Get All Session IDs of a User

**url:** /api/chat/sessions

**method:** GET

**produces:** application/x-www-form-urlencoded

**consumes:** \*/\*

**Note:**

Get all session IDs of the specified user.

**Params:**

name	description	in	require	type	schema
userId	User ID	query	false	integer(int64)	

**Status:**

code	description	schema
200	OK	ResultVO«List«ChatSessionModel»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

**Response Params:**

name	description	type	schema
code		string	
msg		string	
result		array	ChatSessionModel
createTime	Creation time	string(date-time)	
id	Session ID	integer(int64)	
title	Session title	string	
updateTime	Update time	string(date-time)	
userId	User ID	integer(int64)	

**Response Example:**

```

{
  "code": "",
  "msg": "",
  "result": [
    {
      "createTime": "2025-03-09T23:09:17",
      "id": 1,
      "title": "What is the Citi Cup?",
      "updateTime": "2025-03-09T23:09:17",
      "userId": 2
    },
    {
      "createTime": "2025-03-09T23:28:34",
      "id": 2,
      "title": "",
      "updateTime": "2025-03-09T23:28:34",
      "userId": 2
    },
    {
      "createTime": "2025-03-10T10:15:10",
      "id": 3,
      "title": "What is Nanjing University?",
      "updateTime": "2025-03-10T10:15:20",
      "userId": 2
    },
    {
      "createTime": "2025-03-11T20:03:36",
      "id": 4,
      "title": "",
      "updateTime": "2025-03-11T20:03:36",
      "userId": 2
    },
    {
      "createTime": "2025-03-11T20:21:45",
      "id": 5,
      "title": "Risk signals:...",
      "updateTime": "2025-03-11T20:23:11",
      "userId": 2
    }
  ]
}

```

## Streaming Chat Interface

**url:** /api/chat/stream

**method:** GET

**produces:** application/x-www-form-urlencoded

**consumes:** \*/\*

**Note:**

Conduct a chat in a streaming manner.

**Params:**

name	description	in	require	type	schema
message	Message content	query	false	string	
modelName	Model name	query	false	string	
sessionId	Session ID	query	false	integer(int64)	
userId	User ID	query	false	integer(int64)	

**Status:**

code	description	schema
200	OK	Flux«ServerSentEvent«string»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

**Response Params:**

name	description	type	schema
prefetch		integer(int32)	integer(int32)

**Response Example:**

```
data:<thought>Sure</thought>
data:<thought>,</thought>
data:<thought>how</thought>
// Omit the middle content due to space limitations
data:<thought>adjust</thought>
data:<thought>the</thought>
data:<thought>answer</thought>
data:<thought>direction</thought>
data:<thought>.</thought>
data:Hello!
data:Is there anything
data:I can
data:help you
data:with ? Whether it's
data:academic
data:questions,
data:data analysis,
data:life
data:advice,or
data:just a chat,
data:I'm here
```

# User related Interfaces

## Get User Information

**url:** /api/users

**method:** GET

**produces:** application/x-www-form-urlencoded

**consumes:** \*/\*

**Note:**

Get the information of the currently logged-in user.

**Params:**

None

**Status:**

code	description	schema
200	OK	ResultVO«UserVO»
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code		string	
msg		string	
result		UserVO	UserVO
address		string	
id		integer(int32)	
password		string	
phone		string	
role		integer(int32)	
username		string	

Response Example:

```
{
  "code": "",
  "msg": "",
  "result": {
    "address": "163 Xianlin Avenue",
    "id": 2,
    "password": "123456789",
    "phone": "13333333333",
    "role": 2,
    "username": "SeeCoder"
  }
}
```

# Update User Information

**url:** /api/users

**method:** POST

**produces:** application/json

**consumes:** \*/\*

**Note:**

Update the information of the currently logged-in user.

**Example:**

```
{
  "address": "163 Xianlin Avenue",
  "id": 2,
  "password": "123456789",
  "phone": "13333333333",
  "role": 2,
  "username": "SeeCoder"
}
```

**Params:**

name	description	in	require	type	schema
userVO	User information object	body	false	UserVO	UserVO
address	User's address		false	string	
id	User ID		false	integer(int32)	
password	User password		false	string	
phone	User's phone number		false	string	
role	User role		false	integer(int32)	
username	Username		false	string	

**Status:**

code	description	schema
200	OK	ResultVO«boolean»
201	Created	



code	description	schema
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code		string	
msg		string	
result		boolean	

Response Example:

```
{
  "code": "000",
  "msg": null,
  "result": true
}
```

# User Login

url: /api/users/login

method: POST

produces: application/json

consumes: \*/\*

Note:

User login interface.

Params:

Parameter	Description	In	Required	Type	Schema
password	User password	query	true	string	
phone	User phone	query	true	string	

Status:

code	description	schema
200	OK	ResultVO«string»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code		string	
msg		string	
result		string	

Response Example:

```
{
  "code": "000",
  "msg": null,
  "result": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyIiwiaXhwIjozNzQyMzAxOTQ2fQ.zQr01009QUXESy5w2w"
}
```

# User Registration

url: /api/users/register

method: POST

produces: application/json

consumes: \*/\*

Note:

Register a new user

Example:

```
{
  "address": "Nanjing University",
  "id": 6,
  "password": "123456",
  "phone": "1331234567891",
  "role": 1,
  "username": "Test"
}
```

Params:

Parameter Name	Parameter Description	Request Type	Is Required	Data Type	Schema
userVO	User information object	body	true	UserVO	UserVO
address	User address		false	string	
id	User ID		false	integer(int32)	
password	User password		false	string	
phone	User phone number		false	string	
role	User role		false	integer(int32)	
username	Username		false	string	

Status:

code	description	schema
200	OK	ResultVO«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code		string	
msg		string	
result		boolean	

Response Example:

```
{
  "code": "000",
  "msg": null,
  "result": true
}
```

# Risk Signal related Interfaces

## Paged Query Risk Signals

url: /api/risk-signals

method: GET

produces: application/x-www-form-urlencoded

consumes: \*/\*

Note:

Query risk signal list in pages based on time range

Params:

Parameter Name	Parameter Description	Request Type	Is Required	Data Type	Schema
endTime	End time	query	false	string(date-time)	
page	Page number	query	false	integer(int32)	
size	Page size	query	false	integer(int32)	
startTime	Start time	query	false	string(date-time)	

Status:

Status Code	Description	Schema
200	OK	ResultVO«Page«RiskSignalModel»»
401	Unauthorized	

Status Code	Description	Schema
403	Forbidden	
404	Not Found	

Response Params:

Parameter Name	Parameter Description	Type	Schema
code	Status code	string	
msg	Message	string	
result	Risk signal list	Page	Page«RiskSignalModel»
countId	Count ID	string	
current	Current page	integer(int64)	
maxLimit	Max limit	integer(int64)	
optimizeCountSql	Optimize count SQL	boolean	
orders	Sorting orders	array	OrderItem
asc	Ascending	boolean	
column	Column name	string	
pages	Total pages	integer(int64)	
records	Risk signal records	array	RiskSignalModel
advice	Advice	string	
analysis	Analysis result	string	
baseCurrency	Base currency	integer	
emp	EMP value	number	
exchangeRate	Exchange rate	number	
fxReserves	Foreign exchange reserves	number	
id	Signal ID	integer	
interestRate	Interest rate	number	
targetCurrency	Target currency	integer	
time	Time	string	

Parameter Name	Parameter Description	Type	Schema
searchCount	Search count	boolean	
size	Page size	integer(int64)	
total	Total records	integer(int64)	

**Response Example:**

```

{
  "code": "000",
  "msg": null,
  "result": {
    "records": [
      {
        "id": 1,
        "time": "2023-01-01T00:00:00",
        "emp": 100.5,
        "exchangeRate": 1.12,
        "interestRate": 2.5,
        "fxReserves": 500,
        "analysis": "Initial analysis: Stable economy.",
        "advice": "Monitor inflation closely.",
        "targetCurrency": 2,
        "baseCurrency": 1
      },
      // Omitted due to length
      {
        "id": 20,
        "time": "2023-10-15T00:00:00",
        "emp": 100.5,
        "exchangeRate": 1.12,
        "interestRate": 2.5,
        "fxReserves": 495,
        "analysis": "Economy returning to normal, exchange rate stabilizing.",
        "advice": "Maintain prudent fiscal and monetary policies.",
        "targetCurrency": 4,
        "baseCurrency": 3
      }
    ],
    "total": 0,
    "size": 10,
    "current": 1,
    "orders": [],
    "optimizeCountSql": true,
    "searchCount": true,
    "maxLimit": null,
    "countId": null,
    "pages": 0
  }
}

```

## Create Risk Signal

url: /api/risk-signals

**method:** POST

**produces:** application/json

**consumes:** \*/\*

**Note:**

Create a new risk signal

**Example:**

```
{
  "advice": "hard to say",
  "analysis": "what can i say",
  "baseCurrency": 1,
  "emp": 123,
  "exchangeRate": 456,
  "fxReserves": 101,
  "id": 21,
  "interestRate": 10,
  "targetCurrency": 3,
  "time": "2023-10-15T00:00:00"
}
```

**Params:**

Parameter Name	Parameter Description	Request Type	Is Required	Data Type	Schema
signal	Risk signal object	body	true	RiskSignalModel	RiskSignalModel
advice	Advice		false	string	
analysis	Analysis result		false	string	
baseCurrency	Base currency		false	integer(int32)	
emp	EMP value		false	number(double)	
exchangeRate	Exchange rate		false	number(double)	
fxReserves	Foreign exchange reserves		false	number(double)	
id	Signal ID		false	integer(int64)	
interestRate	Interest rate		false	number(double)	



Parameter Name	Parameter Description	Request Type	Is Required	Data Type	Schema
targetCurrency	Target currency		false	integer(int32)	
time	Time		false	string(date-time)	

Status:

Status Code	Description	Schema
200	OK	ResultVO«RiskSignalModel»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

Parameter Name	Parameter Description	Type	Schema
code	Status code	string	
msg	Message	string	
result	Risk signal details	RiskSignalModel	RiskSignalModel
advice	Advice	string	
analysis	Analysis result	string	
baseCurrency	Base currency	integer(int32)	
emp	EMP value	number(double)	
exchangeRate	Exchange rate	number(double)	
fxReserves	Foreign exchange reserves		number(double)
id	Signal ID	integer(int64)	
interestRate	Interest rate	number(double)	
targetCurrency	Target currency	integer(int32)	
time	Time	string(date-time)	

## Response Example:

```
{
  "code": "000",
  "msg": null,
  "result": {
    "id": 21,
    "time": "2023-10-15T00:00:00",
    "emp": 123,
    "exchangeRate": 456,
    "interestRate": 10,
    "fxReserves": 101,
    "analysis": "what can i say",
    "advice": "hard to say",
    "targetCurrency": 3,
    "baseCurrency": 1
  }
}
```

## Advanced Query Risk Signals

**url:** /api/risk-signals/search

**method:** POST

**produces:** application/json

**consumes:** \*/\*

**Note:**

Advanced query based on multiple conditions.

**Example:**

```

{
  "baseCurrency": 0,
  "endTime": "",
  "keyword": "",
  "maxEmp": 0,
  "minEmp": 0,
  "minExchangeRate": 0,
  "page": 0,
  "size": 0,
  "startTime": "",
  "targetCurrency": 0
}

```

Params:

Parameter Name	Parameter Description	Request Type	Is Required	Data Type	Schema
queryDTO	Query condition object	body	true	RiskSignalQueryDTO	RiskSignalQueryDTO
baseCurrency	Base currency		false	integer(int32)	
endTime	End time		false	string(date-time)	
keyword	Keyword		false	string	
maxEmp	Max EMP value		false	number(double)	
minEmp	Min EMP value		false	number(double)	
minExchangeRate	Min exchange rate		false	number(double)	
page	Page number		false	integer(int32)	
size	Page size		false	integer(int32)	
startTime	Start time		false	string(date-time)	
targetCurrency	Target currency		false	integer(int32)	

Status:

Status Code	Description	Schema
200	OK	ResultVO«Page«RiskSignalModel»»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

Parameter Name	Parameter Description	Type	Schema
code	Status code	string	
msg	Message	string	
result	Risk signal list	Page	Page«RiskSignalModel»
countId	Count ID	string	
current	Current page	integer(int64)	
maxLimit	Max limit	integer(int64)	
optimizeCountSql	Optimize count SQL	boolean	
orders	Sorting orders	array	OrderItem
asc	Ascending	boolean	
column	Column name	string	
pages	Total pages	integer(int64)	
records	Risk signal records	array	RiskSignalModel
advice	Advice	string	
analysis	Analysis result	string	
baseCurrency	Base currency	integer	
emp	EMP value	number	
exchangeRate	Exchange rate	number	
fxReserves	Foreign exchange reserves	number	

Parameter Name	Parameter Description	Type	Schema
id	Signal ID	integer	
interestRate	Interest rate	number	
targetCurrency	Target currency	integer	
time	Time	string	
searchCount	Search count	boolean	
size	Page size	integer(int64)	
total	Total records	integer(int64)	

**Response Example:**

```

{
  "code": "",
  "msg": "",
  "result": {
    "countId": "",
    "current": 0,
    "maxLimit": 0,
    "optimizeCountSql": true,
    "orders": [
      {
        "asc": true,
        "column": ""
      }
    ],
    "pages": 0,
    "records": [
      {
        "advice": "",
        "analysis": "",
        "baseCurrency": 0,
        "emp": 0,
        "exchangeRate": 0,
        "fxReserves": 0,
        "id": 0,
        "interestRate": 0,
        "targetCurrency": 0,
        "time": ""
      }
    ],
    "searchCount": true,
    "size": 0,
    "total": 0
  }
}

```

## Update Risk Signal

**url:** /api/risk-signals/{id}

**method:** PUT

**produces:** application/json

**consumes:** \*/\*

**Note:**

Update risk signal by ID

Example:

```
{
  "advice": "hard to say",
  "analysis": "what can i say",
  "baseCurrency": 1,
  "emp": 123,
  "exchangeRate": 456,
  "fxReserves": 101,
  "id": 15,
  "interestRate": 10,
  "targetCurrency": 3,
  "time": "2023-10-15T00:00:00"
}
```

Params:

Parameter Name	Parameter Description	Request Type	Is Required	Data Type	Schema
id	Risk signal ID	path	true	integer(int64)	
signal	Risk signal object	body	true	RiskSignalModel	RiskSignalModel
advice	Advice		false	string	
analysis	Analysis result		false	string	
baseCurrency	Base currency		false	integer(int32)	
emp	EMP value		false	number(double)	
exchangeRate	Exchange rate		false	number(double)	
fxReserves	Foreign exchange reserves		false	number(double)	
id	Signal ID		false	integer(int64)	
interestRate	Interest rate		false	number(double)	
targetCurrency	Target currency		false	integer(int32)	
time	Time		false	string(date-time)	

Status:

Status Code	Description	Schema
200	OK	ResultVO«RiskSignalModel»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

Parameter Name	Parameter Description	Type	Schema
code	Status code	string	
msg	Message	string	
result	Updated risk signal	RiskSignalModel	RiskSignalModel
advice	Advice	string	
analysis	Analysis result	string	
baseCurrency	Base currency	integer(int32)	
emp	EMP value	number(double)	
exchangeRate	Exchange rate	number(double)	
fxReserves	Foreign exchange reserves		number(double)
id	Signal ID	integer(int64)	
interestRate	Interest rate	number(double)	
targetCurrency	Target currency	integer(int32)	
time	Time	string(date-time)	

Response Example:



```
{
  "code": "000",
  "msg": null,
  "result": {
    "id": 15,
    "time": "2023-10-15T00:00:00",
    "emp": 123,
    "exchangeRate": 456,
    "interestRate": 10,
    "fxReserves": 101,
    "analysis": "what can i say",
    "advice": "hard to say",
    "targetCurrency": 3,
    "baseCurrency": 1
  }
}
```

# Delete Risk Signal

**url:** /api/risk-signals/{id}

**method:** DELETE

**produces:** application/x-www-form-urlencoded

**consumes:** \*/\*

**Note:**

Delete risk signal by ID.

**Params:**

Parameter Name	Parameter Description	Request Type	Is Required	Data Type	Schema
id	Risk signal ID	path	true	integer(int64)	

**Status:**

code	description	schema
200	OK	ResultVO«Void»
204	No Content	
401	Unauthorized	
403	Forbidden	

**Response Params:**

name	description	type	schema
code		string	
msg		string	

**Response Example:**

```
{
  "code": "000",
  "msg": null,
  "result": null
}
```

# Risk Cauculation related Interfaces

## Get Alert Information

**url:** /api/calculate/alert

**method:** GET

**produces:** application/x-www-form-urlencoded

**consumes:** \*/\*

**Note:**

Get Alert Information

**Params:**

null

**Status:**

code	description	schema
200	OK	AlertVO
401	Unauthorized	
403	Forbidden	

code	description	schema
404	Not Found	

Response Params:

name	description	type	schema
content		string	
level		string	
title		string	
updateTime		string	

Response Example:

```
[
  {
    "content": "",
    "level": "",
    "title": "",
    "updateTime": ""
  }
]
```

# Get Risk Dashboard Information

**url:** /api/calculate/dashboard

**method:** GET

**produces:** application/x-www-form-urlencoded

**consumes:** \*/\*

**Note:**

Get Risk Dashboard data

**Params:**

null

**Status:**

code	description	schema
200	OK	RiskDashboardVO
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
factorBreakdown		array	RiskFactor
name		string	
value		number(double)	
name		string	
riskStatus		string	
score		number(double)	number(double)
trend		ScoreTrend	ScoreTrend
direction		string	
value		number(double)	
updateTime		string	

Response Example:

```
{
  "factorBreakdown": [
    {
      "name": "",
      "value": 0
    }
  ],
  "name": "",
  "riskStatus": "",
  "score": 0,
  "trend": {
    "direction": "",
    "value": 0
  },
  "updateTime": ""
}
```

# Get Exposure Analysis Information

**url:** /api/calculate/exposure

**method:** GET

**produces:** application/x-www-form-urlencoded

**consumes:** \*/\*

**Note:**

Get Exposure Analysis data

**Params:**

null

**Status:**

code	description	schema
200	OK	ExposureMatrixVO
401	Unauthorized	
403	Forbidden	
404	Not Found	

**Response Params:**

name	description	type	schema
terms		array	Term
amount		number(double)	
currency		integer(int32)	
range		string	
riskLevel		integer(int32)	

Response Example:

```
[
  {
    "terms": [
      {
        "amount": 0,
        "currency": 0,
        "range": "",
        "riskLevel": 0
      }
    ]
  }
]
```

# Get Risk Map Information

url: /api/calculate/map

method: GET

produces: application/x-www-form-urlencoded

consumes: \*/\*

Note:

Get Risk Map data

Params:

null

Status:

code	description	schema
200	OK	RiskMapVO
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
regions		RegionRisk	RegionRisk
currencyPair		integer(int32)	
currentRate		number(double)	
hotNews		array	NewsItem
date		string	
title		string	
url		string	
rateChange		number(double)	
riskLevel		integer(int32)	
suggestions		array	string

Response Example:

```
{
  "regions": {
    "additionalProperties1": {
      "currencyPair": 0,
      "currentRate": 0,
      "hotNews": [],
      "rateChange": 0,
      "riskLevel": 0,
      "suggestions": []
    }
  }
}
```