

1. APIs 2

1.1 Auth Service 4

1.2 Center Service 11

1.3 Gateway Service 26

1.4 Map Service 27

APIs

Version ID	Description	Date
2.0.0	1. Change the API to the latest	May 30,2023
1.0.0	1. A basic API plan	May 01,2023

Version 2.0.0

Modules	Sub Modules	APIs	Description
Authentication Service	-	<ol style="list-style-type: none"> 1. User Login 2. User Logout 3. User Register <FOR DEVELOPER TEST ONLY> 4. Get user profile data 5. Update user profile 6. Get team members and info 7. Invite a team member to the team 8. Accept an invitation 9. Get user's avatar 10. Update user's avatar 	Team management, User information, Authentication
Center Service	Dashboard	<ol style="list-style-type: none"> 1. Predict single clinical text 2. Get the item status ratio 	CodeSystem, Task, Visualization, Dashboard
	CodeSystem	<ol style="list-style-type: none"> 1. Create a new code system 2. Get code system concepts of a team 3. Delete a code system 4. Get all versions of code system 	
	Boards	<ol style="list-style-type: none"> 1. Get board lists 2. Modify a board detail 3. Delete a board 4. Create a new task board 5. Get Task Meta Data 6. Get all map tasks 7. Get map task details 8. Create a map task 9. download map task result 10. Curate the map item 	
Mapper Service	-	<ol style="list-style-type: none"> 1. Map Clinical Text to SNOMED CT 2. Predict the texts 3. Retrain the model 4. Reset the model to original 	Map, Curation, Retrain
Gateway Service	-	<ol style="list-style-type: none"> 1. Redirect requests 	Control and manage the request and response

Based on the user stories, to implement maintainable and extendable system architecture, we split the system into five modules. Each of them represents a Flask application in Docker container.

Version 1.0.0

Module	APIs	Description
--------	------	-------------

Auth Service	<ol style="list-style-type: none"> 1. User Login 2. User Logout 3. User forget password 4. User reset password 5. User Register <FOR DEVELOPER TEST ONLY> 	Authentication, team management, user information
Center Service	<ol style="list-style-type: none"> 1. Create a map task 2. Get detail of a map task 3. Delete a map task 4. Get mapped items of a map task 5. Update(Curate) the category of a mapped item in map task 	
Map Service	<ol style="list-style-type: none"> 1. Create a new UIL category 2. Get details of a category 3. Update a category 4. Delete a category 5. Get the UIL list of all version 6. Get the UIL list of a version 	
Common Service	<ol style="list-style-type: none"> 1. Create a Team 2. Add team member into a team 3. Delete a team member from a team 4. Change owner of a team 5. Update permission of a team member 6. Send a email 7. Upload a file 8. Update SNOMED CT version 9. Get Dictionary 	
Gateway Service	<ol style="list-style-type: none"> 1. Redirect requests 	

Auth Service

DigitalHealth

v1.0.0

Base URLs: <http://localhost:8001>

auth

POST Login

POST /auth/login/email

Authenticates a user by their username and password, and if successful, returns an access token (e.g., JWT).

Body Parameters

email: sunkunxi@gmail.com
password: "12345678"

Params

Name	Location	Type	Required	Description
body	body	object	no	none
» email	body	string	no	User email
» password	body	string	no	User password

Response Examples

200 Response

```
{  
  "code": 200,  
  "msg": "ok"  
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline
400	Bad Request	Invalid Email and Password	Inline
401	Unauthorized	Incorrect password	Inline
404	Not Found	User not found	Inline

Responses Data Schema

HTTP Status Code **200**

Name	Type	Required	Restrictions	Title	description
» code	integer	true	none		none
» msg	string	true	none		none

HTTP Status Code **400**

Name	Type	Required	Restrictions	Title	description
» code	integer	true	none		none
» err	string	true	none		none

HTTP Status Code 401

Name	Type	Required	Restrictions	Title	description
» code	integer	true	none		none
» err	string	true	none		none

HTTP Status Code 404

Name	Type	Required	Restrictions	Title	description
» code	integer	true	none		none
» err	string	true	none		none

POST Logout

POST /auth/logout

Logs out the user by invalidating their access token.

Response Examples

200 Response

```
{
  "code": 200,
  "msg": "ok"
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline
400	Bad Request	Not log in	Inline

Responses Data Schema

HTTP Status Code 200

Name	Type	Required	Restrictions	Title	description
» code	integer	true	none		none
» msg	string	true	none		none

HTTP Status Code 400

Name	Type	Required	Restrictions	Title	description
» code	integer	true	none		none
» err	string	true	none		none

POST Invite a team member to a team

GET /auth/team/invite

This system allows users to add a new team member to the existing team by providing the team ID and the new member's user ID along with their role.

Params

Name	Location	Type	Required	Description
team_id	query	string	no	The unique identifier of the team.

Response Examples

200 Response

```
{
  "team_id": "string"
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline
400	Bad Request	Bad Request	Inline
404	Not Found	Record Not Found	Inline
500	Internal Server Error	Server Error	Inline

Responses Data Schema

HTTP Status Code 200

Name	Type	Required	Restrictions	Title	description
» team_id	string	true	none		none

HTTP Status Code 400

Name	Type	Required	Restrictions	Title	description
» error	string	true	none		none
» message	string	true	none		none

HTTP Status Code 404

Name	Type	Required	Restrictions	Title	description
» error	string	true	none		none
» message	string	true	none		none

HTTP Status Code 500

Name	Type	Required	Restrictions	Title	description
» error	string	true	none		none
» message	string	true	none		none

GET Get team members and info

GET /auth/team

Get the team members and team informations

Params

Name	Location	Type	Required	Description
team_id	query	string	no	none

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

GET Get User profile data

GET /auth/user

Get user profile information, including name, email, nickname, gender

Params

Name	Location	Type	Required	Description
user_id	query	string	no	none

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

PUT Update user profile

PUT /auth/user

Update the user profile

Body Parameters

```
{
  "first_name": "DI",
  "last_name": "Boxjelly",
  "nickname": "di-boxjelly",
  "gender": "male"
}
```

Params

Name	Location	Type	Required	Description
body	body	object	no	none

» first_name	body	string	no	none
» last_name	body	string	no	none
» nickname	body	string	no	none
» gender	body	string	no	none

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

POST Accept a invitation

POST /auth/team/accept

The invitee can accept the invitation by registering using invitation link.

Body Parameters

```
invite_token: string
username: string
email: string
password: string
first_name: string
last_name: string
gender: string
```

Params

Name	Location	Type	Required	Description
body	body	object	no	none
» invite_token	body	string	no	none
» username	body	string	no	none
» email	body	string	no	none
» password	body	string	no	none
» first_name	body	string	no	none
» last_name	body	string	no	none
» gender	body	string	no	none

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
------------------	---------	-------------	-------------

200	OK	Success	Inline
-----	----	---------	--------

Responses Data Schema

GET Get avatar of a user

GET /auth/user/avatar

Get avatar of a user

Params

Name	Location	Type	Required	Description
avatar	query	string	no	none

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

POST Update the avatar file

POST /auth/user/avatar

Body Parameters

avatar: file:///Users/sunkunxi/Pictures/IMG_4391.JPG

Params

Name	Location	Type	Required	Description
body	body	object	no	none
» avatar	body	string(binary)	yes	none

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

Data Schema

Center Service

DigitalHealth

v1.0.0

Base URLs: http://localhost:8002

center

POST Create a Map Task

POST /center/boards/tasks

Create a new map task for a user in a team with the specified UIL version.

Body Parameters

```
file: file:///Users/sunkunxi/Desktop/test.csv
team_id: 60c879e72cb0e6f96d6b0f65
board_id: 60c879e72cb0e6f96d6b0f65
```

Params

Name	Location	Type	Required	Description
Content-Type	header	string	yes	none
body	body	object	no	none
» file	body	string(binary)	no	none
» team_id	body	string	no	none
» board_id	body	string	no	none

Response Examples

200 Response

```
{
  "data": {
    "id": "645328311abd2039faa81f3a",
    "num": 0,
    "status": "pending"
  },
  "code": 0,
  "msg": "string"
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

HTTP Status Code 200

Name	Type	Required	Restrictions	Title	description
» data	object	true	none		none

» » id	string	true	none		none
» » num	integer	true	none		none
» » status	string	true	none		none
» code	number	true	none		none
» msg	string	true	none		none

GET Get All Map Tasks

GET /center/boards/tasks

Params

Name	Location	Type	Required	Description
page	query	integer	no	none
size	query	integer	no	none
team_id	query	string	no	none
board_id	query	string	no	none

Response Examples

success

```
{
  "code": "9",
  "msg": "success",
  "data": {
    "page": 2,
    "size": 10,
    "page_num": 2,
    "tasks": [
      {
        "id": "47",
        "status": "pending",
        "num": 8,
        "create_by": "dolor",
        "create_at": "2003-12-15 15:17:17",
        "update_at": "2006-07-07 11:30:59"
      },
      {
        "id": "52",
        "status": "success",
        "num": 25,
        "create_by": "Duis et nisi dolore ad",
        "create_at": "1981-04-26 06:12:29",
        "update_at": "2007-09-20 23:51:33"
      },
      {
        "id": "35",
        "status": "fail",
        "num": 58,
        "create_by": "exercitation in esse",
        "create_at": "1978-03-07 15:46:58",
        "update_at": "2020-02-01 13:32:27"
      },
      {
        "id": "25",
        "status": "success",
        "num": 10,
        "create_by": "nisi",
        "create_at": "2015-12-17 00:21:47",
        "update_at": "1995-01-16 13:10:39"
      }
    ]
  }
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

HTTP Status Code **200**

Name	Type	Required	Restrictions	Title	description
» code	number	true	none		none
» msg	string	true	none		none
» data	object	true	none		none
» » page	integer	true	none		current page number
» » size	integer	true	none		page size
» » page_num	integer	true	none		total page number
» » tasks	[object]	true	none		none
» » » id	string	true	none		none
» » » status	string	true	none		none
» » » num	integer	true	none		number of mapped items in the task
» » » create_by	string	true	none		none
» » » create_at	string	true	none		none
» » » update_at	string	true	none		none

GET Get Map Task Detail

GET /center/boards/task/detail

Get detail of a map task

Body Parameters

```
{ }
```

Params

Name	Location	Type	Required	Description
page	query	integer	no	none
size	query	integer	no	none
team_id	query	string	no	none
board_id	query	string	no	none
task_id	query	string	no	none
body	body	object	no	none

Response Examples

200 Response

```

{
  "data": {
    "id": "string",
    "status": "pending",
    "create_at": "string",
    "update_at": null,
    "items": [
      {
        "id": "string",
        "mapping_status": 0,
        "original_text": "string",
        "mapped_text": null,
        "confidence": null,
        "source": null,
        "curate": null
      }
    ],
    "num": 0,
    "num_success": 0,
    "num_fail": 0
  },
  "code": "200",
  "msg": "ok"
}

```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

HTTP Status Code 200

Name	Type	Required	Restrictions	Title	description
» data	object	true	none		none
» » id	string	true	none		none
» » status	string	true	none		0: fail 1: pending 2: success
» » create_at	string	true	none		none
» » update_at	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » anonymous	null	false	none		none

or

Name	Type	Required	Restrictions	Title	description
» » » anonymous	string	false	none		none

continued

Name	Type	Required	Restrictions	Title	description
» » items	[object]	true	none		none
» » » id	string	true	none		none
» » » mapping_status	integer	true	none		0: fail 1:success 2: reviewed
» » » original_text	string	true	none		none
» » » mapped_text	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
------	------	----------	--------------	-------	-------------

» » » » <i>anonymous</i>	null	false	none		none
--------------------------	------	-------	------	--	------

or

Name	Type	Required	Restrictions	Title	description
» » » » <i>anonymous</i>	string	false	none		none

continued

Name	Type	Required	Restrictions	Title	description
» » » confidence	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » » <i>anonymous</i>	null	false	none		if map fail

or

Name	Type	Required	Restrictions	Title	description
» » » » <i>anonymous</i>	number	false	none		none

continued

Name	Type	Required	Restrictions	Title	description
» » » source	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » » <i>anonymous</i>	null	false	none		none

or

Name	Type	Required	Restrictions	Title	description
» » » » <i>anonymous</i>	object	false	none		none
» » » » status	integer	true	none		0: snomed, 1: uil
» » » » snomed	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » » » <i>anonymous</i>	null	false	none		none

or

Name	Type	Required	Restrictions	Title	description
» » » » » <i>anonymous</i>	object	false	none		none
» » » » » code	string	true	none		none
» » » » » name	string	true	none		none

continued

Name	Type	Required	Restrictions	Title	description
» » » » uil	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » » » anonymous	null	false	none		none

or

Name	Type	Required	Restrictions	Title	description
» » » » » anonymous	object	false	none		none
» » » » » id	string	true	none		none
» » » » » snomed_code	string	true	none		none
» » » » » name	string	true	none		none

continued

Name	Type	Required	Restrictions	Title	description
» » » curate	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	null	false	none		none

or

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	object	false	none		none
» » » » category_id	string	true	none		none
» » » » category_name	string	true	none		none
» » » » update_at	string	true	none		none

continued

Name	Type	Required	Restrictions	Title	description
» » num	integer	true	none		total number of items
» » num_success	integer	true	none		none
» » num_fail	integer	true	none		none
» code	string	true	none		none
» msg	string	true	none		none

DELETE Delete a Map Task

DELETE /uil/boards/tasks

Params

Name	Location	Type	Required	Description
task_id	query	string	no	none
board_id	query	string	no	none
team_id	query	string	no	none

Response Examples

200 Response


```
{
  "success": true,
  "code": "string",
  "msg": "string"
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

HTTP Status Code **200**

Name	Type	Required	Restrictions	Title	description
» success	boolean	true	none		none
» code	string	true	none		none
» msg	string	true	none		none

GET get all mapped items of a task

GET /map/items

Params

Name	Location	Type	Required	Description
id	query	string	no	taskId

Response Examples

200 Response

```
{
  "data": {
    "items": [
      {
        "itemId": "string",
        "mappingSuccess": true,
        "mappingStatus": 0,
        "originalText": "string",
        "mappedText": null,
        "confidence": null,
        "curate": null,
        "uil": null,
        "snomed": null,
        "source": 0
      }
    ],
    "totalNumber": "string",
    "numberOfSuccess": 0,
    "numberOfFail": 0
  },
  "code": "string",
  "msg": "string"
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

HTTP Status Code 200

Name	Type	Required	Restrictions	Title	description
» data	object	true	none		none
» » items	[object]	true	none		none
» » » itemId	string	true	none		none
» » » mappingSuccess	boolean	true	none		none
» » » mappingStatus	integer	true	none		0: success 1:fail 2: reviewed
» » » originalText	string	true	none		none
» » » mappedText	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	null	false	none		none

or

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	string	false	none		none

continued

Name	Type	Required	Restrictions	Title	description
» » » confidence	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	null	false	none		if mapping fail

or

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	number	false	none		none

continued

Name	Type	Required	Restrictions	Title	description
» » » curate	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	null	false	none		none

or

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	object	false	none		none
» » » » categoryName	string	true	none		none
» » » » categoryId	string	true	none		none
» » » » updateAt	string	true	none		none

continued

Name	Type	Required	Restrictions	Title	description
» » » uil	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	null	false	none		none

or

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	object	false	none		none
» » » » id	string	true	none		none
» » » » snomedCode	string	true	none		none
» » » » name	string	true	none		uil category name, should be same as the mapped Text

continued

Name	Type	Required	Restrictions	Title	description
» » » snomed	any	true	none		none

anyOf

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	null	false	none		none

or

Name	Type	Required	Restrictions	Title	description
» » » » anonymous	object	false	none		none
» » » » code	string	true	none		none
» » » » name	string	true	none		none

continued

Name	Type	Required	Restrictions	Title	description
» » » source	number	true	none		0: fail, 1: snomed, 2: uil
» » totalNumber	string	true	none		none
» » numberOfSuccess	number	true	none		none
» » numberOfFail	number	true	none		none
» code	string	true	none		none
» msg	string	true	none		none

POST Curate The Map Item

POST /center/boards/task/curate

Update(Curate) a category of a mapped item in map task

Body Parameters

```
{
  "board_id": "string",
  "team_id": "string",
  "map_item_id": "string",
  "concept_id": "string"
}
```

Params

Name	Location	Type	Required	Description
id	query	string	no	taskId
body	body	object	no	none
» board_id	body	string	yes	none
» team_id	body	string	yes	none
» map_item_id	body	string	yes	the id of the item
» concept_id	body	string	yes	the id (code) of the curated uil

Response Examples

200 Response

```
{
  "code": "string",
  "msg": "string",
  "success": true
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

HTTP Status Code **200**

Name	Type	Required	Restrictions	Title	description
» code	string	true	none		none
» msg	string	true	none		none
» success	boolean	true	none		none

GET Get Task Meta Data

GET /uil/boards/task/meta

Num of success, number of fail, number of reviewed....

Params

Name	Location	Type	Required	Description
task_id	query	string	no	none

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
------------------	---------	-------------	-------------

200	OK	success	Inline
-----	----	---------	--------

Responses Data Schema

GET Get task board list

GET /uil/boards

Get the task board of a team

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

POST Create a new task board

POST /uil/boards

Get the task board of a team

Body Parameters

```
{
  "team_id": "string",
  "name": "string",
  "description": "string"
}
```

Params

Name	Location	Type	Required	Description
body	body	object	no	none
» team_id	body	string	yes	none
» name	body	string	yes	none
» description	body	string	no	none

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

PUT Modify a board detail

PUT /uil/boards

Modify a board's name and description

Body Parameters

```
{}
```

Params

Name	Location	Type	Required	Description
body	body	object	no	none

Response Examples

200 Response

```
{
  "team_id": "string",
  "board_id": "string",
  "new_name": "string",
  "new_description": "string"
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

HTTP Status Code **200**

Name	Type	Required	Restrictions	Title	description
» team_id	string	true	none		none
» board_id	string	true	none		none
» new_name	string	true	none		none
» new_description	string	true	none		none

GET /center/dashboard/item-status-ratio

GET /center/dashboard/item-status-ratio

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
------------------	---------	-------------	-------------

200	OK	success	Inline
-----	----	---------	--------

Responses Data Schema

center/dashboard

POST Predict single clinical text

POST /center/dashboard/predict

Body Parameters

```
{
  "text": "string"
}
```

Params

Name	Location	Type	Required	Description
body	body	object	no	none
» text	body	string	yes	none

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

center/codesystem

GET Get Code System Concepts of a Team

GET /center/codesystem

Retrieves the UIL category list detail for a specific version.

Params

Name	Location	Type	Required	Description
version	query	string	no	none

Response Examples

200 Response

```

{
  "code": "string",
  "msg": "string",
  "data": {
    "version": "string",
    "categories": [
      {
        "id": "string",
        "name": "string"
      }
    ]
  }
}

```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

HTTP Status Code **200**

Name	Type	Required	Restrictions	Title	description
» code	string	true	none		none
» msg	string	true	none		none
» data	object	true	none		none
» » version	string	true	none		The UIL list version.
» » categories	[object]	true	none		An array of categories for the specified version.
» » » id	string	true	none		none
» » » name	string	true	none		none

POST Create a New Code System

POST /center/codesystem

When a new team is created, create a new code sytem

Response Examples

200 Response

```

{}

```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

DELETE Delete a code system

DELETE /center/codesystem

Delete the code system given a unique version

Params

Name	Location	Type	Required	Description
version	query	string	no	Version of code system

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

GET Get all versions of codesystem

GET /center/codesystem/versions

Get all version of code system from database

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

Data Schema

Gateway Service

Gateway service redirect all APIs to its sub-services.

All APIs requires JWT token, except three API from authentication service, login, logout and register

Redirect request

POST/GET/PUT/DELETE /<path:path>

<Redirect all APIs>

Map Service

DigitalHealth

v1.0.0

Base URLs: http://localhost:8003

map

POST Map Clinical Text To Snomed CT

POST /map/translate

Body Parameters

```
{
  "texts": [
    "Pharyngitis, streptococcal",
    "Abscess - left lower conjunctiva"
  ]
}
```

Params

Name	Location	Type	Required	Description
body	body	object	no	none
» texts	body	[string]	yes	none

Response Examples

200 Response

```
{
  "code": 0,
  "data": {},
  "msg": "string"
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

HTTP Status Code 200

Name	Type	Required	Restrictions	Title	description
» code	integer	true	none		none
» data	object	true	none		none
» msg	string	true	none		none

POST Predict the texts

POST /map/predict

Predict the texts

Body Parameters

```
{
  "texts": [
    "string"
  ]
}
```

Params

Name	Location	Type	Required	Description
body	body	object	no	none
» texts	body	[string]	yes	none

Response Examples

200 Response

```
{
  "code": 0,
  "msg": "string",
  "data": {
    "name": "string",
    "predictions": {}
  }
}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

HTTP Status Code **200**

Name	Type	Required	Restrictions	Title	description
» code	integer	true	none		none
» msg	string	true	none		none
» data	object	true	none		none
» » name	string	true	none		none
» » predictions	object	true	none		none

POST Retrain the model

POST /map/retrain

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

POST Reset the model to original

POST /map/reset

Response Examples

200 Response

```
{}
```

Responses

HTTP Status Code	Meaning	Description	Data schema
200	OK	success	Inline

Responses Data Schema

Data Schema