

AisKosong Technical Documentation v 1.0

1st March 2016

by: Ng Ek Wyn
Shukri Hussain

Table of Content

Introduction	2
Authentication	2
Erros	3
API Call, Request and Response	4
1. Registration	5
2. Verification	6
3. Get List of Universities' IDs	7
4. Validation	9
5. Login	10
6. Getting list of deals	11
7. Getting individual deals	14
8. Redeem deal	15

Introduction

The AisKosong API is intended for mobile app developer to connect with AisKosong backend server for the purpose of storing and retrieving data.

The API:

- All API requests are sent to backend server via HTTP Post except a get request
- Uses predictable, resource-oriented URLs.
- Uses built-in HTTP capabilities for passing parameters and authentication.
- Responds with standard HTTP response code to indicate errors.
- Return JSON.

Base URL

All API URLs reference in this documentation start with the following base part:

<https://api.aiskosong.com/api?>

Authentication

In order to use the API, you need to obtain an API key from us. Please email us at api@aiskosong.com to obtain the key.

Each api request will have to embed an encryption key that is hashed from a private key and request parameter. The hashing is using SHA-1 and the following example is written using Java.

Please refer to the table in section API Call, Parameter and Response to get each api parameter and parameters' order. It is important to construct the parameter sequence as instructed and do include the parameter key even though the value is nil.

Example - Login

Request formed from the parameter:

```
private String request = "req=login&email=john@gmail.com&password=password123"
```

```
private static String generateHashString(String request)
{
    private static PRIVATE_KEY = "Ud1g\\ikWIKFxGZ9X8ngvqA=="

    private string reqWithKey = request + "key=" + PRIVATE_KEY
    try{
        MessageDigest digest = MessageDigest.getInstance("SHA-1");
        byte[] hashedBytes = digest.digest(reqWithKey.getBytes("UTF-8"));

        return convertByteArrayToHexString(hashedBytes)
    }
```

```

    }catch (Exception)
    {
        //Handle exception
    }
}

private static String convertByteArrayToHexString(byte[] arrayBytes) {
    StringBuffer stringBuffer = new StringBuffer();
    for (int i = 0; i < arrayBytes.length; i++) {
        stringBuffer.append(Integer.toString((arrayBytes[i] & 0xff) + 0x100, 16)
            .substring(1));
    }
    return stringBuffer.toString();
}

```

Errors

AisKosong API returns standard HTTP response code.

Code	Description
200	Everything worked as expected
400	Bad request. Often missing a required parameter
401	Encryption key not supplied or incorrect.
402	Parameter and token is valid but request failed.
404	Link not found.
460	User has not been verified.
461	User has not been validated.
462	Incorrect validation code.

API Call, Parameter and Response

AisKosong API response come back as JSON.

1. Registration	5
2. Verification	6
3. Get List of Universities' IDs	7
4. Validation	9
5. Login	10
6. Getting list of deals	11
7. Getting individual deals	14
8. Redeem deal	15

1. Registration

This is to send student details to be registered on the server. The response will give a user token and the activation flag for later use.

Request

Property	Value	Type(Max Length)	Required/Optional
reqtype	“studentreg”	String(10)	Required
firstname	Student’s first name	String(255)	Required
lastname	Student’s last name	String(255)	Required
email	Student’s personal email	String(50)	Required
gender	Student’s gender	Char(1)	Required. Value is either ‘m’ or ‘f’
password	Student’s password	String(30)	Required
retypepassword	Retyping password. Validation to be done on front end and backend	String(30)	Required

Request sample

<https://api.aiskosong.com/api?reqtype=studentreg&firstname=john&lastname=doe&email=john@gmail.com&gender=m&password=password123&retypepassword=password123>

3

Response

Upon registration, server will return appropriate code and message. User token will be stored into mobile application database and is required for future transaction with server.

Property	Type(Length)	Description
type	String(10)	Response type
code	Integer(3)	Response code
message	String(200)	Response message
token	String(50)	User session token

Response sample

```
{
  "response": {
    "type": "studentreg",
    "code": "200",
    "message": "Registration successful.",
    "token": "878KJHKJadsfk=="
  }
}
```

2. Verification

This is to send student institution details to the server. Upon success it will send an email contains a validation code to the student's email address and response accordingly. This is to be done after user has successfully registered using personal email.

Request

Property	Value	Type(Length)	Required/Optional
reqtype	"studentver"	String(10)	Required
institutionid	Institution id	Integer(11)	Required
institutionemail	Institution email	String(255)	Required
token	User's token	String(50)	Required

Request sample

<https://api.aiskosong.com/api?reqtype=studentver&institutionid=00002&institutionemail=jamesbrown@taylors.edu.my&token=878KJHKJadsfk==>

Response

Upon verification, server will return appropriate code and message.

Property	Type(Length)	Description
type	String(10)	Response type
code	Integer(3)	Response code
message	String(200)	Response message
token	String(50)	User session token

Response sample

```
{
  "response": {
    "type": "studentver",
    "code": "200",
    "message": "Verification successful.",
    "token": "878KJHKJadsfk=="
  }
}
```

3. Get List of University IDs

This is to retrieve the list of university IDs during student verification stage.

Request

Property	Value	Type(Max Length)	Required/Optional
reqtype	"getuniversityids"	String(16)	Required
token	User's token	String(50)	Required

Request sample

<https://api.aiskosong.com/api?reqtype=getuniveristyids>

Response

Return the list of university ids and name

Property	Type(Length)	Description
type	String(16)	Response type
code	Integer(3)	Response code
universities	Array	Array of universities
id	Integer(10)	University's id
name	String(200)	University's name

Response sample

```
{
  "response": {
    "type": "getuniversityids",
    "code": "200",
    "Universities":
      [{
        "Id": 00001,
        "name": "Taylor's University"
      },
      {
        "Id": 00002,
        "name": "Monash University"
      },
      {
        "Id": 00003,
        "name": "KDU University"
      }
    ]
  }
}
```

4. Validation

This is to send validation code to be check in the server. Validation code are sent to the user's institution email address. If it is a success, user are flagged as active and logged in and Ais Kosong ID is created and to be passed as the response.

Request

Property	Value	Type(Max Length)	Required/Optional
reqtype	"studentval"	String(10)	Required
token	User's token	String(50)	Required
validationcode	Validation code	String(8)	Required

Request sample

<https://api.aiskosong.com/api?reqtype=studentval&validationcode=Tzh2345tGa&token=878KJHKJadsfk==>

Response

Upon verification, server will return appropriate code and message.

Property	Type(Length)	Description
code	Integer(10)	Response code
message	String(200)	Response message
token	String(50)	User session token
aiskosong_id	String(20)	New generated aiskosong id

Response sample

{

```

"response": {
  "type": "studentval",
  "code": "200",
  "message": "Validation successful.",
  "token": "878KJHKJadsfk==",
  "aiskosong_id": "AK2346"
}
}

```

Note

If the validation code entered by student is incorrect, error code 462 will be returned.

Example of response with error code 462

```

{
  "response": {
    "type": "studentval",
    "code": "462",
    "Message": "Incorrect validation code"
  }
}

```

5. Login

Allow the student to login into aiskosong app.

Request

Property	Value	Type(Max Length)	Required/Optional
reqtype	"login"	String(5)	Required
email	Student's email	String(50)	Required
password	Account's password	String(30)	Required

Request sample

<https://api.aiskosong.com/api?reqtype=login&email=john@gmail.com&password=password123>

Response

Upon successful login, server will return a token and the aiskosong id. The token and the aiskosong id will be stored in the app.

Property	Type(Length)	Description
type	String(5)	Response type
code	Integer(3)	Response code
message	String(200)	Response message
aiskosong_id	String(50)	aiskosong id
token	String(50)	User session token

Response sample

```
{
  "response": {
    "type": "login",
    "code": "200",
    "message": "Login successful.",
    "aiskosong_id" : "AK2346",
    "token": "878KJHKJadsfk=="
  }
}
```

6. Getting List of Deals

Getting list of deals based on location coordinate.

Request

Property	Value	Type(Max Length)	Required/Optional
reqtype	"getdealslist"	String(12)	Required
dealttype	Deal's type	String(50)	Required
longitude	User's longitude	Float	Optional

	location		
latitude	User's latitude location	Float	Optional
token	User's token	String(50)	Required

Request sample

<https://api.aiskosong.com/api?reqtype=getdealslist&dealttype=lunch&longitude=101.518461&latitude=3.073281&token=878KJHKJadsfk==>

Response

Upon successful request, server will return a list of deals near to the user's location.
Sort by the nearest location

Property	Type	Description
type	String	Response type
code	Integer	Response code
message	String	Response message
token	String	User session token
deal_id	Integer	Deal's id
deals	List of object (deal)	List of deals
deal	Object	Deal
deal_name	String	Deal's name
shop_name	String	Deal's shop name
location_name	String	Deal's location name
distance	Double	Distance between given user's location and the shop location in kilometres (km)

price_rating	integer	Current rating price. Out of 3.
shop_icon_link	String	Url of shop icon

Response sample

```
{
  "response": {
    "type": "List of deals",
    "code": "200",
    "message": "List found.",
    "token": "878KJHKJadsfk==",
    "deals": {
      "deal": [
        {
          "deal_id": "252",
          "deal_name": "20% off total bill",
          "shop_name": "Bert's",
          "location_name": "Shah Alam",
          "distance": "1",
          "rating": "2.5",
          "shop_icon_link": "http://aiskosong.com/img/img1.jpg"
        },
        {
          "deal_id": "253",
          "deal_name": "10% off everything",
          "shop_name": "Tappers Cafe",
          "location_name": "Oasis Square",
          "distance": "3",
          "rating": "2.5",
          "shop_icon_link": "http://aiskosong.com/img/img2.jpg"
        },
        {
          "deal_id": "256",
          "deal_name": "10% off",
          "shop_name": "Talent Lounge",
          "location_name": "Damansara Perdana",
          "distance": "3",
          "rating": "2.5",
          "shop_icon_link": "http://aiskosong.com/img/img3.jpg"
        }
      ]
    }
  }
}
```

7. Getting Individual Deals

Getting individual deals based on selection.

Request

Property	Value	Type(Max Length)	Required/Optional
reqtype	"getdeal"	String(7)	Required
dealttype	Deal's type	String(50)	Required
dealid	Deal's id	Integer(10)	Required
token	User's token	String(50)	Required

Request sample

<https://api.aiskosong.com/api?reqtype=getdeals&dealttype=lunch&dealid=2&token=878KJHKJadsfk==>

Response

Upon successful request, server will return a list of deals near to the user's location.
Sort by the nearest location

Property	Type	Description
type	String	Response type
code	Integer	Response code
message	String	Response message
token	String	User session token
main_description	String	Response deal description
extra_description	Object	consist of description
description	List of strings	List of description

shop_icon_link	String	Url of shop icon
----------------	--------	------------------

Response sample

```
{
  "response": {
    "type": "Deal",
    "code": "200",
    "message": "Deal found.",
    "token": "878KJHKJadsfk==",
    "main_description": "Calling all students! Score 10% off when you eat at Talent Lounge. Just show us your student ID at the counter and get 10% off from your La Carte order",
    "extra_description": {
      "description": [
        "Discount can be redeemed multiple times.",
        "Applicable with other ongoing promotions and discounts",
        "Display valid student ID to redeem",
        "Available all day, everyday."
      ],
      "shop_icon_link": "http://aiskosong.com/img/img3.jpg"
    }
  }
}
```

8. Redeem Deal

This is to allow student to get a redemption code for online purchase.

Request

Property	Value	Type(Max Length)	Required/Optional
reqtype	"getdealcode"	String(11)	Required
token	User's token	String(50)	Required
dealid	Deal's id	String(10)	Required

Request sample

<https://api.aiskosong.com/api?reqtype=getdealcode&token=Tzh2345tGa&dealid=1001>

Response

Upon verification, server will return the redemption code

Property	Type(Length)	Description
type	String(11)	Response type
code	Integer(3)	Response code
message	String(200)	
redemptioncode	String(10)	Redemption code

Response sample

```
{
  "response": {
    "type": "getdealcode",
    "code": "200",
    "redemptioncode": "BONUS123"
  }
}
```

Note

Depending on the student status, if the student has not yet verified, the return code will be 460. If the student has not yet validated, the return code will be 461. Developer is required to handle this two error codes specifically so that student is been informed of their status and be reminded to verified/validate the account.

Example of response with error code 461

```
{
  "response": {
    "type": "getdealcode",
    "code": "600",
    "Message": "User has not been verified",
    "redemptioncode": ""
  }
}
```