

---

# KUsistant

*Release 0.1*

**Alkan Burak Ismail Mete**

**May 24, 2022**



**CONTENTS:**

- 1 KUsistant is a personal student assistant. It aims to help student meet their academic and personal needs. 1**
  - 1.1 KUsistant is being developed by 4 Koç University Computer Engineering students: Alkan, Burak, Ismail and Mete. You can read below our docs for different APIs and modules. . . . . 1
- 2 Indices and tables 5**
- Python Module Index 7**
- Index 9**



---

**KUSISTANT IS A PERSONAL STUDENT ASSISTANT. IT AIMS TO HELP STUDENT MEET THEIR ACADEMIC AND PERSONAL NEEDS.**

---

**1.1 KUsistant is being developed by 4 Koç University Computer Engineering students: Alkan, Burak, Ismail and Mete. You can read below our docs for different APIs and modules.**

### 1.1.1 Source codes

#### BlackBoardAPI module

```
class BlackBoardAPI.BlackBoardAPI(cookie: str)
```

Bases: object

```
getAllCourses()
```

It gets the all courses from Blackboard and returns the courses as list

**Returns** the courses list

**Return type** list

```
getCalendarEvents(from_date_ms, to_date_ms)
```

It gets the calendar events from Blackboard and returns the calendar events as JSON

**Parameters**

- **from\_date\_ms** (*str*) – start date of calendar date interval
- **to\_date\_ms** (*str*) – end date of calendar date interval

**Returns** the calendar events on the given interval

**Return type** JSON

```
getGrades(courseID)
```

It gets the grades of a certain course from Blackboard and returns the grades as list

**Parameters** **courseID** (*int*) – The course id for the wanted course grade

**Returns** the grade for given course id

**Return type** int

```
class BlackBoardAPI.Course(courseID, courseName, courseCode, courseSemester, courseSection)
```

Bases: object

```
class BlackBoardAPI.Grade(name, type, due, state, lastActivity, grade, maxGrade)  
    Bases: object  
    print()
```

### DialogFlow\_Bridge\_API module

```
DialogFlow_Bridge_API.KUSISCookie = ''
```

This is the connection between Dialogflow and KUsistant Flutter application

```
DialogFlow_Bridge_API.getCalendarEvents(data)
```

Processes JSON from our BlackBoard API to extract calendar events

**Parameters** *data* (*JSON*) – The answer from the BlackBoard API

**Returns** the processed JSON with the calendar events

**Return type** dict

```
DialogFlow_Bridge_API.getCourseGradeResult(courseId, bbCookie)
```

Processes JSON from our Blackboard API to extract course grades

**Parameters**

- **courseId** (*str*) – The course id for the wanted grades
- **bbCookie** (*str*) – The cookie for user's BlackBoard

**Returns** the processed JSON with the course grades

**Return type** dict

```
DialogFlow_Bridge_API.getGPA(data)
```

Processes JSON from our BlackBoard API to extract GPA

**Parameters** *data* (*JSON*) – The answer from the BlackBoard API

**Returns** the processed JSON with the GPA value

**Return type** dict

```
DialogFlow_Bridge_API.getLetterGrade(data, courseCode)
```

Processes JSON from our KUSIS API to extract letter grade of given course

**Parameters**

- **data** (*JSON*) – The answer from the KUSIS API
- **courseCode** (*str*) – The course for wanted letter grade

**Returns** the processed JSON with the letter grade

**Return type** dict

```
DialogFlow_Bridge_API.getWeatherInfo(data)
```

Processes JSON returned from weather API

**Parameters** *data* (*JSON*) – The answer from the OpenWeather API

**Returns** the processed JSON with the max and min temperature values

**Return type** dict

`DialogFlow_Bridge_API.processRequest(req, userID)`

Posts request to the relevant API and processes the result.

**Parameters**

- **req** (*str*) – The request for the API
- **userID** (*int*) – The userID mapped to the cookies

**Returns** the processed JSON from the given request

**Return type** JSON

`DialogFlow_Bridge_API.setBetweenDates()`

Returns a time range between now and next 14 days in milliseconds

**Returns** the int pair values of now and next 14 days in milliseconds

**Return type** tuple

`DialogFlow_Bridge_API.test()`

`DialogFlow_Bridge_API.webhook()`

## KusisAPI module

**class** `KusisAPI.AcademicRecord`(*code, name, semester, grade, credits*)

Bases: object

**class** `KusisAPI.KusisAPI`(*cookie*)

Bases: object

**getGPA()**

It gets the GPA from KUSIS and returns it

**Returns** the dictionary element mapped to GPA

**Return type** dict

**getLetterGrades()**

It gets the letter grades from KUSIS and returns the grades

**Returns** the letter grade records

**Return type** list

## QuestionAnsweringModel module

`QuestionAnsweringModel.predict_answer(question, context)`

Predicts an answer for the given question from the context

**Parameters**

- **question** (*str*) – The question asking to the model
- **context** (*str*) – The context of the asked question

**Returns** the predicted answer

**Return type** str

### main module

#### `main.getAllCourses()`

It gets the all courses from Blackboard and returns the courses

**Returns** the courses list

**Return type** JSON

#### `main.getCalendarEvent()`

It gets the calendar events from Blackboard and returns the calendar events

**Returns** the calendar events on the given interval

**Return type** JSON

#### `main.getCourseID()`

It gets the course ID of a certain course from Blackboard and returns it

**Returns** the dictionary element mapped to course id

**Return type** dict

#### `main.getGPA()`

It gets the GPA from KUSIS and returns it

**Returns** the GPA

**Return type** JSON

#### `main.getGrade(course_id)`

It gets the grades of a certain course from Blackboard and returns the grades

**Parameters** `courseID (int)` – The course id for the wanted course grade

**Returns** the grade for given course id

**Return type** JSON

#### `main.getLetterGrades()`

It gets the letter grades of a certain course from KUSIS and returns the grades

**Returns** the grade for given course id

**Return type** JSON

#### `main.hello()`



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### b

BlackBoardAPI, 1

### d

DialogFlow\_Bridge\_API, 2

### k

KusisAPI, 3

### m

main, 4

### q

QuestionAnsweringModel, 3



## INDEX

### A

AcademicRecord (*class in KuisAPI*), 3

### B

BlackBoardAPI  
    module, 1

BlackBoardAPI (*class in BlackBoardAPI*), 1

### C

Course (*class in BlackBoardAPI*), 1

### D

DialogFlow\_Bridge\_API  
    module, 2

### G

getAllCourses() (*BlackBoardAPI.BlackBoardAPI*  
    *method*), 1

getAllCourses() (*in module main*), 4

getCalendarEvent() (*in module main*), 4

getCalendarEvents() (*BlackBoard-*  
    *API.BlackBoardAPI method*), 1

getCalendarEvents() (*in module Di-*  
    *alogFlow\_Bridge\_API*), 2

getCourseGradeResult() (*in module Di-*  
    *alogFlow\_Bridge\_API*), 2

getCourseID() (*in module main*), 4

getGPA() (*in module DialogFlow\_Bridge\_API*), 2

getGPA() (*in module main*), 4

getGPA() (*KuisAPI.KuisAPI method*), 3

getGrade() (*in module main*), 4

getGrades() (*BlackBoardAPI.BlackBoardAPI method*),  
    1

getLetterGrade() (*in module Di-*  
    *alogFlow\_Bridge\_API*), 2

getLetterGrades() (*in module main*), 4

getLetterGrades() (*KuisAPI.KuisAPI method*), 3

getWeatherInfo() (*in module Di-*  
    *alogFlow\_Bridge\_API*), 2

Grade (*class in BlackBoardAPI*), 1

### H

hello() (*in module main*), 4

### K

KuisAPI  
    module, 3

KuisAPI (*class in KuisAPI*), 3

KUSISCookie (*in module DialogFlow\_Bridge\_API*), 2

### M

main  
    module, 4

module  
    BlackBoardAPI, 1  
    DialogFlow\_Bridge\_API, 2  
    KuisAPI, 3  
    main, 4  
    QuestionAnsweringModel, 3

### P

predict\_answer() (*in module QuestionAnswering-*  
    *Model*), 3

print() (*BlackBoardAPI.Grade method*), 2

processRequest() (*in module Di-*  
    *alogFlow\_Bridge\_API*), 2

### Q

QuestionAnsweringModel  
    module, 3

### S

setBetweenDates() (*in module Di-*  
    *alogFlow\_Bridge\_API*), 3

### T

test() (*in module DialogFlow\_Bridge\_API*), 3

### W

webhook() (*in module DialogFlow\_Bridge\_API*), 3