API Documentation

Content

Login:

Call file: api/login.php
Request type: - http
Method: POST
Input:

      key: -   Key provided by the API developers
      username: -   User name of teacher/ student or COE
      password: -   Password of the given username.
      category: -   0/1/2 (0 for COE, 1 for teacher, 2 for student)

E.g.:

      key=WnXsEnsDk&username=GCECTB-R17-3018&password=1234&category=2

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:

      Type: json file

- **If successful:**

    First parameter "status" value of which will be "OK"
    Second parameter "comment" value of which will be "login successful"
    E.g.:
```
{
        status: "OK",
        comment: "login successful"
}
```

- **If fail:**

    First parameter "status" value of which will be "FAIL"
    Second parameter "comment" value of which can be either of:
    "incorrect key: XYZ" – when the key is not correct.
    "incorrect username: XYZ" – when the username is not correct.
    "incorrect password" – when the password is not correct.
    "incorrect category: XYZ" – when the category is anything other than 0 or 1 or 2.
    "undefined parameter: key" – when the parameter "key" is not given.
    "undefined parameter: username" – when the parameter "username" is not given.
    "undefined parameter: password" – when the parameter "password" is not given.
    "undefined parameter: category" – when the parameter "category" is not given.
    "server error occurred" – when some server-side error occurs

E.g.1:
```
{
    status: "FAIL",
    comment: "incorrect username: XYZ"
}
```
E.g.2:

```
{
    status: "FAIL",
    comment: "undefined parameter: key"
}
```

Individual Student's info:

Call file: api/student_info.php
Request type: - http
Method: GET
Input:

        key: -   Key provided by the API developers
        username: -   User name of student.
        E.g.:
        api/student_info.php?key=WnXsEnsDk&username=GCECTB-R17-3018

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:

        Type: json file

- **If successful:**

    First parameter "status" value of which will be "OK"
    Second parameter "result" value of which will be the information of the
        student whose username is provided. The result will contain (username, name,
        department, joining year, pass out year, registration no, email, contact no)
    E.g.:

```
{
        status: "OK",
        result: {
                username: "GCECTB-R17-3018",
                name: "RASHED MEHDI",
                department: "CSE",
                joining year: 2017,
                pass out year: 2021,
                registration no: "",
                email: "",
                contact no: ""
        }
}
```

- **If fail:**

    First parameter "status" value of which will be "FAIL"
    Second parameter "comment" value of which can be either of:
    "incorrect key: XYZ" – when the key is not correct.
    "incorrect username: XYZ" – when the username is not correct.
    "undefined parameter: key" – when the parameter "key" is not given.
    "undefined parameter: username" – when the parameter "username" is not given.

E.g.1:

Request:

api/student_info.php?key=WnXsEnsDk&username=GCECTB-R17-3018

Output:

```
{
    status: "FAIL",
    comment: "incorrect key: WnXsEnsDk"
}
```

E.g.2:

Request:

api/student_info.php?key=WnXsEnsDk&uname=GCECTB-R17-3018

```
{
    status: "FAIL",
    comment: "undefined parameter: username"
}
```

Group Student info:

Call file: api/student_group_info.php
Request type: - http
Method: GET
Input:

key: -	Key provided by the API developers
stream: -	can be either of CSE/IT/CT or 1 for all the department
batch_passout_year: -	passing out year of the batch (e.g. 2021) or 1 for all the passing out year
joining_year: -	joining year (e.g. 2017) or 1 for all the joining year

NOTE: *joining_year* includes all the student that has joined in the year given, this will not include a year lag student. *batch_passout_year* on the other hand will include the year lag student and will display all the students who are expected to pass out in the given year.

E.g.1:
api/student_group_info.php?key=WnXsEnsDk&stream=1&batch_passout_year=1&joining_year=2017

This will give all the student data who have joined in the year 2017.

E.g.2:
api/student_group_info.php?key=WnXsEnsDk&stream=CSE&batch_passout_year=2021&joining_year=1

This will give all the student data who will pass out in the year 2021 and the stream is CSE.

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:
   Type: json file

   - **If successful:**

      First parameter "status" value of which will be "OK"
      Second parameter "result" value of which will be an array of information of the students who met the given criteria. The array will contain (username, name, department, joining year, pass_out_year, registration no, email, contact no)

      E.g.1:
      Request:

api/student_group_info.php?key=WnXsEnsDk&stream=1&batch_passout_year=1&joining_year=2017

Output:

```
{
        status: "OK",
        result: [
                {
                        username: "GCECTB-R17-3018",
                        name: "RASHED MEHDI",
                        department: "CSE",
                        joining year: 2017,
                        pass out year: 2021,
                        registration no: "",
                        email: "",
                        contact no: ""
                },
                {
                        username: "GCECTB-R17-1019",
                        name: "RISHAV BANERJEE",
                        department: "CT",
                        joining year: 2017,
                        pass out year: 2021,
                        registration no: "",
                        email: "",
                        contact no: ""
                },
                {
                        username: "GCECTB-R17-3030",
                        name: "SOUMYA MUKHERJEE",
                        department: "CSE",
                        joining year: 2017,
                        pass out year: 2021,
                        registration no: "",
                        email: "",
                        contact no: ""
                },
                {
                        username: "GCECTB-R17-2024",
                        name: "SARANYA NAHA ROY",
                        department: "IT",
                        joining year: 2017,
                        pass out year: 2022,
                        registration no: "",
                        email: "",
                        contact no: ""
                }
        ]
}
```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:
  "incorrect key: XYZ" – when the key is not correct.
  "incorrect stream: XYZ" – when the stream is other than CSE/IT/CT/1
  "incorrect joining_year: XYZ" – when the joining year is not 1 or in the format 20XX.
  "incorrect batch_passout_year XYZ" - when the pass out year is not 1 or in the format 20XX.
  "undefined parameter: key" – when the parameter "key" is not given.
  "undefined parameter: stream" – when the parameter stream is not defined.
  "undefined parameter: joining_year" – when the parameter joining_year is not defined.
  "undefined parameter: batch_passout_year" - when the parameter batch_passout_year is not defined.

  E.g.1:
  Request:
  api/student_group_info.php?key=WnXsEnsDk&stm=CSE&batch_passout_year=1&joining_year=2017
  Output:
  ```
  {
          status: "FAIL",
          comment: "undefined parameter: stream"
  }
  ```

  E.g.2:
  Request:
  api/student_group_info.php?key=WnXsEnsDk&stream=CSE&batch_passout_year=1&joining_year=3017

  Output:
  ```
  {
          status: "FAIL",
          comment: "incorrect joining year: 3017"
  }
  ```

Individual Teacher's info:

Call file: api/teacher_info.php
Request type: - http
Method: GET
Input:

      key: -   Key provided by the API developers
      username: -   User name of teacher.

E.g.:

      api/teacher_info.php?key=WnXsEnsDk&username=GCECT/F/0001

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:
      Type: json file

- **If successful:**

   First parameter "status" value of which will be "OK"
   Second parameter "result" value of which will be the information of the
         teacher whose username is provided. The result will contain (username,
         honorary title, first name, last name, department, designation, address, email,
         contact no)
   E.g.:

```
{
     status: "OK",
     result: {
          username: "GCECTF/0001",
          honorary title: "Mr"
          first name: "SAGAR",
          last name: "MAJUMDAR"
          department: "CSE",
          designation: "Assistant professor",
          address: "",
          email: "",
          contact no: ""
     }
}
```

- **If fail:**

   First parameter "status" value of which will be "FAIL"
   Second parameter "comment" value of which can be either of:
   "incorrect key: XYZ" – when the key is not correct.
   "incorrect username: XYZ" – when the username is not correct.
   "undefined parameter: key" – when the parameter "key" is not given.

"undefined parameter: username" – when the parameter "username" is not given.

E.g.1:
Request:
api/teacher_info.php?key=WnXsEnsDk&username=GCECTF/0001
Output:
```
{
     status: "FAIL",
     comment: "incorrect key: WnXsEnsDk"
}
```

E.g.2:
Request:
api/student_info.php?key=WnXsEnsDk&username=GCECTB-R17-3018
```
{
     status: "FAIL",
     comment: "incorrect username: GCECTB-R17-3018"
}
```

Group Teacher info:

Call file: api/teacher_group_info.php
Request type: - http
Method: GET
Input:

key: -	Key provided by the API developers
stream: -	can be either of CSE/IT/CT or 1 for all the department

E.g.:

api/teacher_group_info.php?key=WnXsEnsDk&stream=CSE

This will give all the teacher's data who are in CSE department.

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:
Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "result" value of which will be an array of information of the teachers who met the given criteria. The array will contain (username, honorary title, first name, last name, department, designation, email, contact no)

E.g.1:
Request:
api/teacher_group_info.php?key=WnXsEnsDk&stream=CSE

Output:
```
{
        status: "OK",
        result: [
                {
                        username: "GCECTF/0001",
                        honorary title: "Mr"
                        first name: "SAGAR",
                        last name: "MAJUMDAR"
                        department: "CSE",
                        designation: "Assistant professor",
                        address: ""
                        email: "",
                        contact no: ""
                },
```

```
              {
                      username: "GCECTF/0002",
                      honorary title: "Mr"
                      first name: "PARTHA",
                      last name: "HALDAR"
                      department: "CSE",
                      designation: "Assistant professor",
                      address: ""
                      email: "",
                      contact no: ""
              },
              {
                      username: "GCECTF/0001",
                      honorary title: "Mr"
                      first name: "KINGSHUK",
                      last name: "CHATERJEE"
                      department: "CSE",
                      designation: "Assistant professor",
                      address: ""
                      email: "",
                      contact no: ""
              },
              {
                      username: "GCECTF/0001",
                      honorary title: "Mr"
                      first name: "ALOK",
                      last name: "MAJUMDAR"
                      department: "CSE",
                      designation: "Assistant professor",
                      address: ""
                      email: "",
                      contact no: ""
              }
          ]
  }
```

- **If fail:**

   First parameter "status" value of which will be "FAIL"
   Second parameter "comment" value of which can be either of:
   "incorrect key: XYZ" – when the key is not correct.
   "incorrect stream: XYZ" – when the stream is other than CSE/IT/CT/1
   "undefined parameter: key" – when the parameter "key" is not given.
   "undefined parameter: stream" – when the parameter stream is not defined.

   E.g.1:
   Request:
   api/teacher_group_info.php?key=WnXsEnsDk&stm=CSE
   Output:

```
{
    status: "FAIL",
    comment: "undefined parameter: stream"
}
```

E.g.2:
Request:
api/teacher_group_info.php?key=WnXsEnsDk&stream=CSTE

Output:
```
{
    status: "FAIL",
    comment: "incorrect stream: CSTE"
}
```

Add a Student:

Call file: api/add_student.php
Request type: - http
Method: POST
Input:

      key: -   Key provided by the API developers
      username: -   User name of student.
      password: -   Password of the new student's account.

Output:

    Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "student added successfully"

  E.g.:
  ```
  {
       status: "OK",
       comment: "student added successfully"
  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:
  "incorrect key: XYZ" – when the key is not correct.
  "username: XYZ already exists" – when the username already exists.
  "undefined parameter: key" – when the parameter "key" is not given.
  "undefined parameter: username" – when the parameter "username" is not given.
  "undefined parameter: password" – when the parameter "password" is not given.
  "server error occurred" – when some server error occurs.

  E.g.1:
  Output:
  ```
  {
       status: "FAIL",
       comment: "username: GCECTB-R17-3018 already exists"
  }
  ```

  E.g.2:
  ```
  {
       status: "FAIL",
       comment: "undefined parameter: password"
  }
  ```

Modify a Student's data:

Call file: api/modify_student.php
Request type: - http
Method: POST
Input:

key: -	Key provided by the API developers
username: -	User name of student.
name: -	Name of the student.
department: -	department of the student which should be either CSE or IT or CT (cannot be null)
registration_no: -	registration no of the student or null if not provided
joining_year: -	joining year of the student (cannot be null)
email: -	email of the student or null if not provided
contact_no: -	contact no of the student or null if not provided

Output:
Type: json file

- **If successful:**

    First parameter "status" value of which will be "OK"
    Second parameter "comment" value of which will be "record successfully updated"

    E.g.:
    ```
    {
          status: "OK",
          comment: "record successfully updated"
    }
    ```

- **If fail:**

    First parameter "status" value of which will be "FAIL"
    Second parameter "comment" value of which can be either of:
    "incorrect key: XYZ" –   when the key is not correct.
    "incorrect username: XYZ" –   when the username does not exist.
    "undefined parameter: key" –   when the parameter "key" is not given.
    "undefined parameter: username" –   when the parameter "username" is not given.
    "undefined parameter: name" –   when the parameter "name" is not given.
    "undefined parameter: department" – when the parameter "department" is not given.
    "incorrect department: XYZ" –   when the parameter "department" is other than CSE or IT or CT.
    "undefined parameter: registration_no" – when the parameter "registration_no" is not given.

"undefined parameter: joining_year" – when the parameter "joining_year" is not given.

"incorrect joining_year: XYZ" – when the parameter "joining_year" is not in the format 20XX.

"undefined parameter: joining_year" – when the parameter "joining_year" is not given.

"undefined parameter: email" – when the parameter "email" is not given.

"undefined parameter: contact_no" – when the parameter "contact_no" is not given.

"server error occurred" – when some server error occurs.

E.g.1:
```
{
     status: "FAIL",
     comment: "username: GCECTB-R17-3018 does not exist"
}
```

E.g.2:
```
{
     status: "FAIL",
     comment: "undefined parameter: joining_year"
}
```

Delete a student:

Call file: api/delete_student.php
Request type: - http
Method: GET
Input:

      key: -   Key provided by the API developers
      username: -   User name of student.

Output:

    Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "student deleted"

  E.g.:
  ```
  {
        status: "OK",
        comment: "student deleted"
  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:
  "incorrect key: XYZ" –   when the key is not correct.
  "incorrect username: XYZ" –   when the username does not exist.
  "undefined parameter: key" –   when the parameter "key" is not given.
  "undefined parameter: username" – when the parameter "username" is not given.

  E.g.1:
  Output:
  ```
  {
        status: "FAIL",
        comment: "username: GCECTB-R17-3018 does not exist"
  }
  ```

  E.g.2:
  ```
  {
        status: "FAIL",
        comment: "undefined parameter: key"
  }
  ```

Year lag a student:

Call file: api/year_lag_student.php
Request type: - http
Method: GET
Input:

key: -   Key provided by the API developers
username: - User name of student.

Output:

  Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "student deleted"

  E.g.:
  ```
  {
       status: "OK",
       comment: "record updated successfully"
  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:
  "incorrect key: XYZ" –   when the key is not correct.
  "incorrect username: XYZ" –   when the username does not exist.
  "undefined parameter: key" –   when the parameter "key" is not given.
  "undefined parameter: username" – when the parameter "username" is not given.

  E.g.1:
  Output:
  ```
  {
       status: "FAIL",
       comment: "username: GCECTB-R17-3018 does not exist"
  }
  ```

  E.g.2:
  ```
  {
       status: "FAIL",
       comment: "undefined parameter: key"
  }
  ```

Add a Teacher:

Call file: api/add_teacher.php
Request type: - http
Method: POST
Input:

key: -	Key provided by the API developers
username: -	User name of teacher.
password: -	Password of the new teacher's account.
Stream: -	Stream of the teacher.

Output:

Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "teacher added successfully"

  E.g.:
  ```
  {
      status: "OK",
      comment: "teacher added successfully"
  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:
  "incorrect key: XYZ" – when the key is not correct.
  "username: XYZ already exists" – when the username already exists.
  "undefined parameter: key" – when the parameter "key" is not given.
  "undefined parameter: username" – when the parameter "username" is not given.
  "undefined parameter: password" – when the parameter "password" is not given.
  "server error occurred" – when some server error occurs.

  E.g.1:
  Output:
  ```
  {
      status: "FAIL",
      comment: "username: GCECTF/0001 already exists"
  }
  ```

  E.g.2:
  ```
  {
      status: "FAIL",
      comment: "undefined parameter: password"
  }
  ```

}

Modify a Teacher's data:

Call file: api/modify_teacher.php
Request type: - http
Method: POST
Input:

key: -	Key provided by the API developers
username: -	User name of teacher.
name: -	Name of teacher.
department: -	department of the teacher which should be either CSE or IT or CT or BSEH (cannot be null)
designation: -	designation of the teacher.
address: -	address of the teacher or null if not provided
email: -	email of the student or null if not provided
contact_no: -	contact no of the student or null if not provided

Output:

Type: json file

- **If successful:**

    First parameter "status" value of which will be "OK"
    Second parameter "comment" value of which will be "record successfully updated"

    E.g.:
    ```
    {
        status: "OK",
        comment: "record successfully updated"
    }
    ```

- **If fail:**

    First parameter "status" value of which will be "FAIL"
    Second parameter "comment" value of which can be either of:

"incorrect key: XYZ" –	when the key is not correct.
"incorrect username: XYZ" –	when the username does not exist.
"undefined parameter: key" –	when the parameter "key" is not given.
"undefined parameter: username" –	when the parameter "username" is not given.
"undefined parameter: name" –	when the parameter "fname" is not given.
"undefined parameter: department" –	when the parameter "department" is not given.
"undefined parameter: designation" –	when the parameter "designation" is not given.
"incorrect department: XYZ" –	when the parameter "department" is other than CSE or IT or CT or BSEH.
"undefined parameter: address" –	when the parameter "address" is not given.

"undefined parameter: email" –   when the parameter "email" is not given.
"undefined parameter: contact_no" – when the parameter "contact_no" is not given.
"server error occurred" –   when some server error occurs.

E.g.1:
```
{
     status: "FAIL",
     comment: "username: GCECTF/0001 does not exist"
}
```

E.g.2:
```
{
     status: "FAIL",
     comment: "undefined parameter: email"
}
```

Delete a Teacher:

Call file: api/delete_teacher.php
Request type: - http

Method: GET

Input:

      key: -   Key provided by the API developers

      username: -   User name of teacher.

Output:

      Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "comment" value of which will be "teacher deleted"

E.g.:
```
{
        status: "OK",
        comment: "teacher deleted"
}
```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:
"incorrect key: XYZ" –   when the key is not correct.
"incorrect username: XYZ" –   when the username does not exist.
"undefined parameter: key" –   when the parameter "key" is not given.
"undefined parameter: username" – when the parameter "username" is not given.

E.g.1:
Output:
```
{
        status: "FAIL",
        comment: "username: GCECTF/0001 does not exist"
}
```

E.g.2:
```
{
        status: "FAIL",
        comment: "undefined parameter: key"
}
```

Change Password:

Call file: api/change_password.php
Request type: - http

Method: POST
Input:

    key: -   Key provided by the API developers
    username: -   Username.
    old_password – Old password
    new_password – New password
    category -   should be either 0 or 1 or 2 (0 for COE, 1 for teacher, 2 for student)

Output:

    Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "password updated successfully"

  E.g.:
  ```
  {
        status: "OK",
        comment: "password updated successfully"
  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:

  "undefined parameter: key" –   when the parameter "key" is not given.
  "incorrect key: XYZ" –   when the key is not correct.
  "undefined parameter: username" –   when the parameter "username" is not given.
  "incorrect username: XYZ" –   when the username does not exist.
  "undefined parameter: old_password" – when the parameter "old_password" is not
                                         given.
  "incorrect old password" –   when old password is not correct.
  "undefined parameter: new_password" – when the parameter "new_password" is not
                                           given.
  "undefined parameter: category" –   when the parameter "category" is not given.
  "incorrect category: XYZ" –   when category is other than 0 or 1 or 2.

  E.g.1:
  Output:
  ```
  {
        status: "FAIL",
  ```

```
          comment: "incorrect username: GCECTF/0001"
}


E.g.2:
{
          status: "FAIL",
          comment: "incorrect category: 4"
}
```

Reset Password:

Call file: api/reset_password.php
Request type: - http
Method: POST
Input:

key: -   Key provided by the API developers
username: -   Username.
new_password – New password

category -   should be either 0 or 1 or 2 (0 for COE, 1 for teacher, 2 for student)

Output:
Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "password updated successfully"

  E.g.:
  ```
  {
        status: "OK",
        comment: "password updated successfully"
  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:
  "undefined parameter: key" –   when the parameter "key" is not given.
  "incorrect key: XYZ" –   when the key is not correct.
  "undefined parameter: username" –   when the parameter "username" is not given.
  "incorrect username: XYZ" –   when the username does not exist.
  "undefined parameter: new_password" – when the parameter "new_password" is not
                                              given.
  "undefined parameter: category" –   when the parameter "category" is not given.
  "incorrect category: XYZ" –   when category is other than 0 or 1 or 2.

  E.g.1:
  Output:
  ```
  {
        status: "FAIL",
        comment: "incorrect username: GCECTF/0001"
  }
  ```

  E.g.2:
  ```
  {
        status: "FAIL",
        comment: "incorrect category: 4"
  }
  ```

Add a New Subject Code:

Call file: api/add_subcode.php

Request type: - http
Method: POST
Input:

key: -	Key provided by the API developers.
username: -	Username of COE.
password: -	password of COE.
subject_code: -	Subject code to add.
paper_name: -	Name of the paper.

Output:

Type: json file

- **If successful:**

    First parameter "status" value of which will be "OK"
    Second parameter "comment" value of which will be "subject code added successfully"
    E.g.:
    ```
    {
        status: "OK",
        comment: "subject code added successfully"
    }
    ```

- **If fail:**

    First parameter "status" value of which will be "FAIL"
    Second parameter "comment" value of which can be either of:

"incorrect key: XYZ" –	when the key is not correct.
"undefined parameter: key" –	when the parameter "key" is not given.
"undefined parameter: username"-	when the parameter "username" is not given
"undefined parameter: password"-	when the parameter "password" is not given.
"undefined parameter: subject_code" –	when the parameter "subject_code" is not given.
"undefined parameter: paper_name" –	when the parameter "paper_name" is not given.
"incorrect username: XYZ"-	when the username is incorrect.
"incorrect password"-	when the password is incorrect.
"subject code already exists" –	when given subject code already exists.
"server error occurred" –	when some server-side error occurs

E.g.1:
```
{
    status: "FAIL",
    comment: "incorrect key: XYZ"
}
```
E.g.2:

```
{
    status: "FAIL",
    comment: "subject code: CS502 already exists"
}
```

Modify Paper name of a Subject Code:

Call file: api/mod_subcode.php
Request type: - http
Method: POST
Input:

key: -	Key provided by the API developers.
username: -	Username of COE.
password: -	password of COE.
subject_code: -	Subject code to add.
new_paper_name: -	Modified name of the paper.

Output:

    Type: json file

- **If successful:**

    First parameter "status" value of which will be "OK"
    Second parameter "comment" value of which will be "paper name modified"
    E.g.:
    ```
    {
        status: "OK",
        comment: "paper name modified"
    }
    ```

- **If fail:**

    First parameter "status" value of which will be "FAIL"
    Second parameter "comment" value of which can be either of:

"incorrect key: XYZ" –	when the key is not correct.
"undefined parameter: key" –	when the parameter "key" is not given.
"undefined parameter: subject_code" –	when the parameter "subject_code" is not given.
"undefined parameter: new_paper_name" –	when the parameter "new_paper_name" is not given.
"undefined parameter: username"-	when the parameter "username" is not given
"undefined parameter: password"-	when the parameter "password" is not given.
"incorrect username: XYZ"-	when the username is incorrect.
"incorrect password"-	when the password is incorrect.
"incorrect subject code: XXX" –	when given subject code does not exists.
"server error occurred" –	when some server-side error occurs

    E.g.1:
    ```
    {
        status: "FAIL",
    ```

```
        comment: "incorrect key: XYZ"
    }
E.g.2:

    {
        status: "FAIL",
        comment: "incorrect subject code: CS502f"
    }
```

Delete a Subject Code:

Call file: api/delete_subcode.php
Request type: - http
Method: POST
Input:

key: -	Key provided by the API developers.
username: -	Username of COE.
password: -	password of COE.
subject_code: -	Subject code to delete.

Output:

Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "subject code deleted"
  E.g.:
  ```
  {
        status: "OK",
        comment: "subject code deleted"
  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:

"incorrect key: XYZ" –	when the key is not correct.
"undefined parameter: key" –	when the parameter "key" is not given.
"undefined parameter: subject_code" –	when the parameter "subject_code" is not given.
"undefined parameter: new_paper_name" –	when the parameter "new_paper_name" is not given.
"undefined parameter: username"-	when the parameter "username" is not given
"undefined parameter: password"-	when the parameter "password" is not given.
"incorrect username: XYZ"-	when the username is incorrect.
"incorrect password"-	when the password is incorrect.
"incorrect subject code: XXX" –	when given subject code does not exists.
"server error occurred" –	when some server-side error occurs

  E.g.1:
  ```
  {
  ```

```
        status: "FAIL",
        comment: "incorrect key: XYZ"
}
```
E.g.2:

```
{
        status: "FAIL",
        comment: "incorrect subject code: CS502f"
}
```

Add a New Chapter under a Teacher:

Call file: api/add_chap.php
Request type: - http
Method: POST
Input:

key: -	Key provided by the API developers.
username: -	Username of Teacher.
password: -	password of Teacher.
sub_code: -	Subject code under which the chapter will be added.
chapter_name: -	Name of the chapter.

Output:

> **Type:** json file

- **If successful:**

    First parameter "status" value of which will be "OK"
    Second parameter "comment" value of which will be "chapter added successfully"
    E.g.:
    ```
    {
        status: "OK",
        comment: "chapter added successfully"
    }
    ```

- **If fail:**

    First parameter "status" value of which will be "FAIL"
    Second parameter "comment" value of which can be either of:

"incorrect key: XYZ" –	when the key is not correct.
"undefined parameter: key" –	when the parameter "key" is not given.
"undefined parameter: username"-	when the parameter "username" is not given
"undefined parameter: password"-	when the parameter "password" is not given.
"undefined parameter: sub_code" –	when the parameter "subject_code" is not given.
"undefined parameter: chapter_name" –	when the parameter "chapter_name" is not given.
"incorrect username: XYZ"-	when the username is incorrect.
"incorrect password"-	when the password is incorrect.
"incorrect sub_code: XYZ" –	when given subject code is incorrect.
"subject code: XYZ is not present the teacher" –	when given subject code is not present under the given teacher. First add the subject code under the teacher and then try.
"chapter name: XYZ already exists" –	when given chapter name already exists under the teacher.

"server error occurred" –   when some server-side error occurs

E.g.1:
```
{
    status: "FAIL",
    comment: "incorrect key: XYZ"
}
```
E.g.2:

```
{
    status: "FAIL",
    comment: "chapter name: Test chapter #2 already exists"
}
```

Modify a Chapter's name under a Teacher:

Call file: api/mod_chap.php

Request type: - http
Method: POST
Input:

key: -	Key provided by the API developers.
username: -	Username of Teacher.
password: -	password of Teacher.
sub_code: -	Subject code under which the chapter will be modified.
chapter_id: -	The chapter's id whose name is to be modified.
new_chapter_name: -	Name of the chapter.

Output:

Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "comment" value of which will be "chapter modified successfully"
E.g.:
```
{
    status: "OK",
    comment: "chapter modified successfully"
}
```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:

"incorrect key: XYZ" –	when the key is not correct.
"undefined parameter: key" –	when the parameter "key" is not given.
"undefined parameter: username"-	when the parameter "username" is not given
"undefined parameter: password"-	when the parameter "password" is not given.
"undefined parameter: sub_code" –	when the parameter "sub_code" is not given.
"undefined parameter: new_chapter_name" –	when the parameter "new_paper_name" is not given.
"incorrect username: XYZ"-	when the username is incorrect.
"incorrect password"-	when the password is incorrect.
"incorrect sub_code: XYZ" –	when given subject code is incorrect.
"subject code: XYZ is not present the teacher" –	when given subject code is not present under the given teacher. First add the subject code under the teacher and then try.
"incorrect chapter_id: XYZ" –	when given "chapter_id" is incorrect.
"server error occurred" –	when some server-side error occurs.

E.g.1:
```
{
    status: "FAIL",
    comment: "incorrect key: XYZ"
}
```
E.g.2:

```
{
    status: "FAIL",
    comment: "incorrect chapter_id: 205"
}
```

Delete a Chapter under a Teacher:

Call file: api/delete_chap.php
Request type: - http

Method: POST

Input:

key: -	Key provided by the API developers.
username: -	Username of Teacher.
password: -	password of Teacher.
sub_code: -	Subject code under which the chapter will be deleted.
chapter_id: -	The chapter's id which is to be deleted.

Output:

Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "comment" value of which will be "chapter modified successfully"
E.g.:
```
{
        status: "OK",
        comment: "chapter deleted"
}
```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:

"incorrect key: XYZ" –	when the key is not correct.
"undefined parameter: key" –	when the parameter "key" is not given.
"undefined parameter: username"-	when the parameter "username" is not given
"undefined parameter: password"-	when the parameter "password" is not given.
"undefined parameter: sub_code" –	when the parameter "sub_code" is not given.
"incorrect username: XYZ"-	when the username is incorrect.
"incorrect password"-	when the password is incorrect.
"incorrect sub_code: XYZ" –	when given subject code is incorrect.
"subject code: XYZ is not present the teacher" –	when given subject code is not present under the given teacher.
"incorrect chapter_id: XYZ" –	when given "chapter_id" is incorrect.
"server error occurred" –	when some server-side error occurs.

E.g.1:
```
{
        status: "FAIL",
        comment: "incorrect key: XYZ"
}
```

E.g.2:

```
{
    status: "FAIL",
    comment: "incorrect chapter_id: 205"
}
```

Add a New Question in the question bank:

Call file: api/add_ques.php
Request type: - http

Method: POST
Input:

key: -	Key provided by the API developers.
username: -	Username of Teacher.
password: -	password of Teacher.
sub_code: -	Subject code which contains the chapter under which the question will be added.
chapter_id: -	Chapter's Id under which the question will be added.
st: -	Question statement.
ac: -	Correct Answer.
wa1: -	Incorrect Answer 1.
wa2: -	Incorrect Answer 2.
wa3: -	Incorrect Answer 3.

Output:

Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "question added successfully"
  Third parameter "id" will be the id of the question which is added.

  E.g.:
  ```
  {
        status: "OK",
        comment: "question added successfully",
        id: 8
  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:

"incorrect key: XYZ" –	when the key is not correct.
"undefined parameter: key" –	when the parameter "key" is not given.
"undefined parameter: username"-	when the parameter "username" is not given
"undefined parameter: password"-	when the parameter "password" is not given.
"undefined parameter: sub_code" –	when the parameter "subject_code" is not given.
"undefined parameter: chapter_id" –	when the parameter "chapter_id" is not given.
"undefined parameter: st" –	when the parameter "st" is not given.

"undefined parameter: ac" –   when the parameter "ac" is not given.
"undefined parameter: wa1" –   when the parameter "wa1" is not given.
"undefined parameter: wa2" –   when the parameter "wa2" is not given.
"undefined parameter: wa3" –   when the parameter "wa3" is not given.
"incorrect username: XYZ"-   when the username is incorrect.
"incorrect password"-   when the password is incorrect.
"incorrect sub_code: XYZ" –   when given subject code is incorrect.
"subject code: XYZ is not present the teacher" –   when given subject code is
                                        not present under the given teacher. First
                                        add the subject code under the teacher and then
                                        try.
"incorrect chapter_id: XYZ" –when given chapter_id id incorrect.
"server error occurred" –   when some server-side error occurs

E.g.1:
```
{
     status: "FAIL",
     comment: "incorrect key: XYZ"
}
```
E.g.2:

```
{
     status: "FAIL",
     comment: "invalid chapter_id: 80"
}
```

Modify a Question in the question bank:

Call file: api/mod_ques.php
Request type: - http
Method: POST

Input:

key: -	Key provided by the API developers.
username: -	Username of Teacher.
password: -	password of Teacher.
sub_code: -	Subject code which contains the chapter under which the question will be added.
chapter_id: -	Chapter's Id under which the question will be added.
ques_id: -	Question Id of the question to be modified.
st: -	Modified question statement.
ac: -	Modified correct Answer.
wa1: -	Modified incorrect Answer 1.
wa2: -	Modified incorrect Answer 2.
wa3: -	Modified incorrect Answer 3.

Output:

Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "comment" value of which will be "question added successfully"
Third parameter "id" will be the id of the question which is added.

E.g.:
```
{
    status: "OK",
    comment: "question modified successfully",
    id: 8
}
```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:

"incorrect key: XYZ" –   when the key is not correct.
"undefined parameter: key" –   when the parameter "key" is not given.
"undefined parameter: username"-   when the parameter "username" is not given
"undefined parameter: password"-   when the parameter "password" is not given.
"undefined parameter: sub_code" – when the parameter "subject_code" is not
                                      given.
"undefined parameter: chapter_id" – when the parameter "chapter_id" is not
                                      given.
"undefined parameter: ques_id" –   when the parameter "ques_id" is not

given.

"incorrect ques_id: XYZ" –   when the parameter "ques_id" is incorrect.
"undefined parameter: st" –   when the parameter "st" is not given.
"undefined parameter: ac" –   when the parameter "ac" is not given.
"undefined parameter: wa1" –   when the parameter "wa1" is not given.
"undefined parameter: wa2" –   when the parameter "wa2" is not given.
"undefined parameter: wa3" –   when the parameter "wa3" is not given.
"incorrect username: XYZ"-   when the username is incorrect.
"incorrect password"-   when the password is incorrect.
"incorrect sub_code: XYZ" –   when given subject code is incorrect.
"subject code: XYZ is not present the teacher" –   when given subject code is not present under the given teacher. First add the subject code under the teacher and then try.
"incorrect chapter_id: XYZ" –when given chapter_id id incorrect.
"server error occurred" –   when some server-side error occurs

E.g.1:

```
{
    status: "FAIL",
    comment: "incorrect key: XYZ"
}
```

E.g.2:

```
{
    status: "FAIL",
    comment: "invalid chapter_id: 80"
}
```

Delete a question:

Call file: api/delete_question.php
Request type: - http
Method: POST
Input:

key: -   Key provided by the API developers
questionId: -   Id of the question to be deleted.
chapterId: -   Id of the chapter.
code: -   subject code.
username: -   username of the teacher.

password: -   password of the teacher.

E.g.:

key=WnXsEnsDk&questionId=7894&chapterId=5240&code=sAd&username=
"Dinesh Kartik"&password=125

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:

Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "question is deleted"
  E.g.:
  ```
  {
        status: "OK",
        comment: "question is deleted"
  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:
  "undefined parameter: key" – when the parameter "key" is not given.
  "undefined parameter: questionId" – when the parameter "questionId" is not
  given.
  "undefined parameter: chapterId" – when the parameter "chapterId" is not
  given.
  "undefined parameter: code" – when the parameter "code" is not
  given.
  "undefined parameter: username" – when the parameter "username" is not
  given.
  "undefined parameter: password" – when the parameter "password" is not
  given.
  "incorrect key: XYZ" – when the parameter "key" is not correct.
  "incorrect questionId: XYZ" – when the parameter "questionId" is not correct.
  "incorrect chapterId: XYZ" – when the parameter "chapterId" is not correct.
  "incorrect code: XYZ" – when the parameter "code" is not correct.
  "teacher XYZ has not undertaken subject ABC" – when the pair {"username",
  "code"}
  is invalid

  "incorrect username/password: XYZ" – when the parameter "username" or
  "password" is not correct.

<center>(XYZ is username)</center>

<center>"server error occurred" – when some server-side error occurs</center>

E.g.2:

```
{
      status: "FAIL",
      comment: "undefined parameter: key"
}
```

Display all available subject codes:

Call file: api/all_subjects.php
Request type: - http
Method: GET
Input:
      key: -   Key provided by the API developers
E.g.:
      key=WnXsEnsDk

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:
      Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "result" value of which will be an array of information   of the
subjects. The array will contain (id, code, name)
E.g.:
```
{
      status: "OK",
      result: [
            {
                  code: "FB780",
                  name: "Mario"
            },
            {
                  code: "AM780",
                  name: "Remember Frankenstein"
            },
```

```
                    {
                        code: "SHFFL780",
                        name: "Shuffling & Playing"
                    },
                ]


        }
```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:
"incorrect key: XYZ" – when the key is not correct.
"undefined parameter: key" – when the parameter "key" is not given.
"server error occurred" – when some server-side error occurs

E.g.1:

```
{
        status: "FAIL",
        comment: "undefined parameter: key"
}
```

Display all chapters under a teacher for a specific subject:

Call file: api/all_chapters.php
Request type: - http
Method: GET
Input:

key: -	Key provided by the API developers
username: -	username of the teacher.
code: -	code of the subject. code = "ALL" to display chapters from all the subjects.

E.g.:
key=WnXsEnsDk?username=W3School&code=ALL

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:

    Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "result" value of which will be an array of information   of the
chapters. The array will contain (id, name, numberOfQuestions,
                             code)

E.g.:

```
{
    status: "OK",
    result: [
        {
            id: "10",
            name: "Einstein Has a Cat",
            numberOfquestions: "5",
            code: "45B"
        },
        {
            id: "32",
            name: "House of a Mouse",
            numberOfQuestions: "10",
            code: "L4L",

        },
        {
            id: "52",
            name: "Parrots Can Talk",
            numberOfQuestions: "2",
            code: "SUBGP32"

        },
        {
            id: "101",
            name: "Have a good day",
            numberOfQuestions: "2",
            code: "N0GP"
        }
    ]


}
```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:
"incorrect key: XYZ" – when the key is not correct.

"undefined parameter: key" – when the parameter "key" is not given.
"undefined parameter: username" – when the parameter "username" is not
given.
"undefined parameter: code" – when the parameter "code" is not
given.
"incorrect username: XYZ" – when the parameter "username" is not correct.
"incorrect code: XYZ" – when the parameter "code" is not correct.
"teacher ABC has not undertaken subject XYZ" – when the pair
{"username", "code"} is invalid
"server error occurred" – when some server-side error occurs

E.g.1:

```
{
        status: "FAIL",
        comment: "undefined parameter: key"
}
```

Display Questions:

Call file: api/questions.php
Request type: - http
Method: POST
Input:

key: -	Key provided by the API developers.
questionId: -	Id of the question or ALL to display all questions.
chapterId: -	Id of the chapter or ALL to display questions from all chapters.
code: -	subject code or ALL to display questions from all subjects.
username: -	username of the teacher.
password: -	password of the teacher.

E.g.:

key=WnXsEnsDk&questionId=7894&chapterId=ALL&code=sAd&username=
"Dinesh Kartik"&password=125

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:
Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "result" value of which will be information of the question. The
array will contain (id, problemStatement, correctOption,
incorrectOption1, incorrectOption2,
incorrectOption3, chapterId,chapterName,code,paperName)

E.g.:

```
{
     status: "OK",
     result: [
          {
               id: 178,
               problemStatement: "html problem",
               correctOption: "html option",
               incorrectOption1: "html option",
               incorrectOption2: "html option",
               incorrectOption3: "html option",
               chapterId: 201,
               chapterName: "banannaBerry",
               code: "LM10",
               paperName: "Mirror Cube",

          },
          {
               id: 118,
               problemStatement: "html problem",
               correctOption: "html option",
               incorrectOption1: "html option",
               incorrectOption2: "html option",
               incorrectOption3: "html option",
               chapterId: 401,
               chapterName: "server loading",
               code: "MU250",
               paperName: "Manchester Rocks",

          },
          {
               id: 528,
               problemStatement: "html problem",
               correctOption: "html option",
               incorrectOption1: "html option",
               incorrectOption2: "html option",
               incorrectOption3: "html option",
               chapterId: 230,
               chapterName: "Extraction",
               code: "GOA250",
               paperName: "Journey with Zombie",
```

```
                    }

            ]
    }
```

- **If fail:**

  First parameter "<mark>status</mark>" value of which will be "<mark>FAIL</mark>"
  Second parameter "<mark>comment</mark>" value of which can be either of:
  "<mark>undefined parameter: key</mark>" – when the parameter "key" is not given.
  "<mark>undefined parameter: questionId</mark>" – when the parameter "questionId" is not
                                                        given.
  "<mark>undefined parameter: chapterId</mark>" – when the parameter "chapterId" is not
                                                        given.
  "<mark>undefined parameter: code</mark>" – when the parameter "code" is not
                                                        given.
  "<mark>undefined parameter: username</mark>" – when the parameter "username" is not
                                                        given.
  "<mark>undefined parameter: password</mark>" – when the parameter "password" is not
                                                        given.
  "<mark>incorrect key: XYZ</mark>" – when the parameter "key" is not correct.
  "<mark>incorrect questionId: XYZ</mark>" – when the parameter "questionId" is not correct.
  "<mark>incorrect chapterId: XYZ</mark>" – when the parameter "chapterId" is not correct.
  "<mark>incorrect code: XYZ</mark>" – when the parameter "code" is not correct.
  "<mark>teacher XYZ has not undertaken subject ABC</mark>" – when the pair {"username",
                                                        "code"}
                                                        is invalid

  "<mark>incorrect username/password: XYZ</mark>" – when the parameter "username" or
                                                        "password" is not correct.
                                                        (XYZ is username)
  "<mark>server error occurred</mark>" – when some server-side error occurs

  E.g.2:

```
{
      status: "FAIL",
      comment: "undefined parameter: key"
}
```

Teacher add subject code under him:

Call file: api/take_subjects.php
Request type: - http
Content-Type: - application/json
Method: POST
Input:

key: -	Key provided by the API developers
username: -	username of the teacher.
password: -	password of the teacher.
codes: -	Array of subject codes.

Output:

Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "comment" value of which will be "subjects have been added to the teacher"

E.g.:
```
{
    status: "OK",
    comment: "subjects have been added to the teacher",

}
```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:
"incorrect key: XYZ" – when the key is not correct.
"undefined parameter: key" – when the parameter "key" is not given.
"undefined parameter: username" – when the parameter "username" is not given.
"undefined parameter: password" – when the parameter "password" is not given.

"undefined parameter: codes" – when the parameter "codes" is not given.
"incorrect username/password: XYZ" – when the parameter "username" or "password" is not correct. XYZ is username.
"incorrect code: XYZ" – when one of the codes in array is not correct.
"empty array of codes" – when the array "codes" is empty.

"server error occurred" – when some server-side error occurs

E.g.1:

```
{
        status: "FAIL",
        comment: "undefined parameter: key"
}
```

Teacher delete subject code under him:

Call file: api/remove_subject.php
Request type: - http
Method: POST
Input:
key: -   Key provided by the API developers.
username: -   username of the teacher.
password: -   password of the teacher.
code: -   code of the subject to be removed.

E.g.:
        key=WnXsEnsDk?username= "Lewandwoski"&password= "Long Range Shot"&code=
"practice".

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:
        Type: json file

- **If successful:**

        First parameter "status" value of which will be "OK"
        Second parameter "comment" value of which will be "subject has been
                                                        removed from the teacher"
        E.g.:
```
{
        status: "OK",
        comment: "subject has been removed from the
                  teacher",

}
```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:

"undefined parameter: key" – when the parameter "key" is not given.
"undefined parameter: code" – when the parameter "code" is not
given.
"undefined parameter: username" – when the parameter "username" is not
given.
"undefined parameter: password" – when the parameter "password" is not
given.
"incorrect key: XYZ" – when the parameter "key" is not correct.
"incorrect code: XYZ" – when the parameter "code" is not correct.
"teacher XYZ has not undertaken subject ABC" – when the pair {"username",
"code"}
is invalid

"incorrect username/password: XYZ" – when the parameter "username" or
"password" is not correct.
(XYZ is username)
"server error occurred" – when some server-side error occurs

E.g.1:

```
{
        status: "FAIL",
        comment: "undefined parameter: key"
}
```

Display all subject codes under a particular teacher:

Call file: api/teacher_codes.php
Request type: - http
Method: GET
Input:
    key: -   Key provided by the API developers
    username: -   username of the teacher.

E.g.:

key=WnXsEnsDk?username=25

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:

Type: json file

- **If successful:**

    First parameter "status" value of which will be "OK"
    Second parameter "result" value of which will be an array of information   of the
    subject-codes. The array will contain (id, code, name)
    E.g.:

    ```
    {
        status: "OK",
        result: [
                {
                    id: "10",
                    code: "NF7",
                    name: "Netflix Rocks",

                },
                {
                    id: "10",
                    code: "CHOW2020",
                    name: "Love Chinese, Eat Chinese",

                }

        ]


    }
    ```

- **If fail:**

    First parameter "status" value of which will be "FAIL"
    Second parameter "comment" value of which can be either of:
    "incorrect key: XYZ" – when the key is not correct.
    "undefined parameter: key" – when the parameter "key" is not given.
    "undefined parameter: username" – when the parameter "teacherId" is not
                                              given.
    "incorrect username: XYZ" – when the parameter "teacherId" is not correct.
    "server error occurred" – when some server-side error occurs

E.g.1:

```
{
        status: "FAIL",
        comment: "undefined parameter: key"
}
```

add questions to an exam:

Call file: api/add_exam_questions.php
Request type: - http
Content-Type: - application/json
Method: POST
Input:

key: -	Key provided by the API developers.
username: -	username of the teacher.
password: -	password of the teacher.
examId: -	id of the exam.
code: -	subject code.
questions: -	array of questions. Each question is an array where first element is the chapterId and second element is the questionId.

E.g.:
     key=WnXsEnsDk?username= "Lewandwoski"&password= "Long Range Shot"&examId=4&code= "practice"&questions=[[4,6],[5,9],[8,10]].

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:
    Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "questions are added
                           to the exam"

  E.g.:
  ```
  {
          status: "OK",
          comment: "questions are added to the exam",

  }
  ```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:

"undefined parameter: key" – when the parameter "key" is not given.
 "undefined parameter: code" – when the parameter "code" is not
given.
"undefined parameter: username" – when the parameter "username" is not
given.
"undefined parameter: password" – when the parameter "password" is not
given.
"undefined parameter: examId" – when the parameter "examId" is not
given.
"undefined parameter: questions" – when the parameter "questions" is not
given.
"incorrect key: XYZ" – when the parameter "key" is not correct.
 "incorrect code: XYZ" – when the parameter "code" is not correct.
"incorrect examId: XYZ" – when the parameter "examId" is not correct.
"teacher XYZ has not undertaken subject ABC" – when the pair {"username",
"code"}
is invalid
"incorrect chapterId: XYZ" – when one of chapterId s in parameter
"questions" is not correct.
"invalid pair (chapter,question)" – when one of the pairs in parameter
"questions" is not correct.
"incorrect username/password: XYZ" – when the parameter "username" or
"password" is not correct.
(XYZ is username)
 "server error occurred" – when some server-side error occurs

E.g.1:

```
{
      status: "FAIL",
      comment: "undefined parameter: key"
}
```

save attempts in an exam:

Call file: api/attempt.php
Request type: - http

Method: POST

Input:

      key: -   Key provided by the API developers.

      username: -   username of the student

      password: -   password of the student

      examId: -   id of the exam.

      questionNumber: -   question number

      optionNumber: -   attempted option. 0 for no attempt. Note that attempted option must be one of the values:{0,1,2,3,4}

E.g.:

      key=WnXsEnsDk?username= "Lewandwoski"&password= "Long Range Shot"&examId=4&questionNumber=2&optionNumber=3

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:

      Type: json file

- **If successful:**

      First parameter "status" value of which will be "OK"

      Second parameter "comment" value of which will be "attempt is saved"

      E.g.:

```
{
        status: "OK",
        comment: "attempt is saved",

}
```

- **If fail:**

      First parameter "status" value of which will be "FAIL"

      Second parameter "comment" value of which can be either of:

      "undefined parameter: key" – when the parameter "key" is not given.

      "undefined parameter: username" – when the parameter "username" is not given.

      "undefined parameter: password" – when the parameter "password" is not given.

      "undefined parameter: examId" – when the parameter "examId" is not given.

      "undefined parameter: questionNumber" – when the parameter "questionNumber" is not given.

      "undefined parameter: optionNumber" – when the parameter

<div align="center">"optionNumber" is not

given.</div>

"incorrect key: XYZ" – when the parameter "key" is not correct.
"incorrect examId: XYZ" – when the parameter "examId" is not correct.
"incorrect questionNumber: XYZ" – when the parameter "questionNumber" is
<div align="center">not correct.</div>
"incorrect optionNumber: XYZ" – when the parameter "optionNumber" is
<div align="center">not correct.</div>
"further submission is not allowed" – invalid attempt to save answer
"incorrect username/password: XYZ" – when the parameter "username" or
<div align="right">"password" is not correct.</div>
<div align="right">(XYZ is username)</div>
"server error occurred" – when some server-side error occurs

E.g.1:

```
{
      status: "FAIL",
      comment: "undefined parameter: key"
}
```

COE deletes an exam:

Call file: api/coe_delete_exam.php
Request type: - http
Method: POST
Input:

key: -	Key provided by the API developers.
username: -	username of the COE
password: -	password of the COE
examId: -	id of the exam.

E.g.:

key=WnXsEnsDk?username= "Lewandwoski"&password= "Long Range Shot"&examId=4

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:
Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "exam has been deleted"
  E.g.:

  ```
  {
        status: "OK",
        comment: "exam has been deleted",

  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:

  "undefined parameter: key" – when the parameter "key" is not given.
   "undefined parameter: username" – when the parameter "username" is not
  given.
  "undefined parameter: password" – when the parameter "password" is not
  given.
  "undefined parameter: examId" – when the parameter "examId" is not
  given.
  "incorrect key: XYZ" – when the parameter "key" is not correct.
   "incorrect examId: XYZ" – when the parameter "examId" is not correct.
  "incorrect username/password: XYZ" – when the parameter "username" or
  "password" is not correct.
  (XYZ is username)
   "server error occurred" – when some server-side error occurs

  E.g.1:

  ```
  {
        status: "FAIL",
        comment: "undefined parameter: key"
  }
  ```

COE ends an exam:

Call file: api/coe_end_exam.php
Request type: - http
Method: POST

Input:

key: - Key provided by the API developers.
username: - username of the COE
password: - password of the COE
examId: - id of the exam or "ALL" to close all the exams.

E.g.:

key=WnXsEnsDk?username= "Lewandwoski"&password= "Long Range Shot"&examId=ALL

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:

Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "comment" value of which will be "exam has been ended"
E.g.:
```
{
        status: "OK",
        comment: "exam has been ended",

}
```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:

"undefined parameter: key" – when the parameter "key" is not given.
 "undefined parameter: username" – when the parameter "username" is not given.
"undefined parameter: password" – when the parameter "password" is not given.
"undefined parameter: examId" – when the parameter "examId" is not given.
"incorrect key: XYZ" – when the parameter "key" is not correct.
 "incorrect examId: XYZ" – when the parameter "examId" is not correct.
"incorrect username/password: XYZ" – when the parameter "username" or "password" is not correct.
(XYZ is username)
 "server error occurred" – when some server-side error occurs

E.g.1:

```
{
        status: "FAIL",
        comment: "undefined parameter: key"
}
```

COE starts an exam:

Call file: api/coe_start_exam.php
Request type: - http
Method: POST
Input:

key: -   Key provided by the API developers.
username: -   username of the COE
password: -   password of the COE
examId: -   id of the exam or "ALL" to start all the exams.

E.g.:

key=WnXsEnsDk?username= "Lewandwoski"&password= "Long Range
Shot"&examId=ALL

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:
    Type: json file

- **If successful:**

  First parameter "status" value of which will be "OK"
  Second parameter "comment" value of which will be "exam has been started"
  E.g.:
  ```
  {
        status: "OK",
        comment: "exam has been started",

  }
  ```

- **If fail:**

  First parameter "status" value of which will be "FAIL"
  Second parameter "comment" value of which can be either of:

"undefined parameter: key" – when the parameter "key" is not given.
 "undefined parameter: username" – when the parameter "username" is not given.
"undefined parameter: password" – when the parameter "password" is not given.
"undefined parameter: examId" – when the parameter "examId" is not given.
"incorrect key: XYZ" – when the parameter "key" is not correct.
 "incorrect examId: XYZ" – when the parameter "examId" is not correct.
"incorrect examId: XYZ" – when the parameter "examId" is not correct.
"invalid attempt to activate two exams on same subject" – can not activate two exams on same subject.
"incorrect username/password: XYZ" – when the parameter "username" or "password" is not correct.
(XYZ is username).
 "server error occurred" – when some server-side error occurs

E.g.1:

```
{
        status: "FAIL",
        comment: "undefined parameter: key"
}
```

teacher creates exam:

Call file: api/create_exam.php
Request type: - http
Method: POST
Input:

key: -   Key provided by the API developers.
username: -   username of the teacher.
password: -   password of the teacher.
description: - description of the exam.
batchPassoutYear: -   batch passout year of intended students.
stream: -
1: "CSE"
2: "IT"
3: "CSE + IT"
4: "CT"

5: "CSE" + "CT"
6: "IT" + "CT"
7: "CSE" + "CT" + "IT"
 use these codes for the stream
code: - subject code.

E.g.:

key=WnXsEnsDk?username= "Lewandwoski"&password= "Long Range Shot"&description=nice&batchPassoutYear=2019&stream=5&code=CS502

(NOTE: here "WnXsEnsDk" is not the actual key, it is just shown for clarity of the example)

Output:

Type: json file

- **If successful:**

First parameter "status" value of which will be "OK"
Second parameter "result" value of which will be an associative array containing the exam id

E.g.:
```
{
        status: "OK",
        result: {id: 250}

}
```

- **If fail:**

First parameter "status" value of which will be "FAIL"
Second parameter "comment" value of which can be either of:

"undefined parameter: ABC" – when the parameter "ABC" is not given.
 "incorrect key: XYZ" – when the parameter "key" is not correct.
"incorrect code: XYZ" – when the parameter "code" is not correct.
"incorrect username/password: XYZ" – when the parameter "username" or
                                                          "password" is not correct.
                                                          (XYZ is username)
"teacher XYZ has not undertaken subject ABC" – when the pair {"username",
                                                                          "code"}
                                                                          is invalid
"incorrect batchPassoutYear: XYZ" – when the parameter
                                                "batchPassoutYear" is not correct.
"incorrect stream: XYZ" – when the parameter "stream" is not correct.
"server error occurred" – when some server-side error occurs
E.g.1:

```
{
     status: "FAIL",
     comment: "undefined parameter: key"
}
```

delete questions from the exam	
Call File	delete_exam_questions.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
username	Username of the teacher
Password	Password of the teacher
questionId	id of the question
examId	id of the exam
code	subject code
Example	
key=WnXsEnsDk?username=Lewandwoski&password=Long Range Shot&questionId=250&examId=20&code=DOG40	
Output if success	
status	OK
comment	question is deleted from the exam
Example	
{ status: "OK", comment: "question is deleted from the exam" }	
Output if failure	
status	FAIL
comment	comments are listed below
Comments	
undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect

teacher XYZ has not undertaken subject ABC	When the pair {username, code} is invalid
server error occurred	When some server-side error occurs.
Example	

```
{
status: "FAIL",
comment: "undefined parameter: key"
}
```

delete exam	
Call File	delete_exam.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
username	Username of the teacher
Password	Password of the teacher
examId	id of the exam
code	subject code
Example	

```
key=WnXsEnsDk?username=Lewandwoski&password=Long
Range Shot&examId=20&code=DOG40
```

Output if success	
status	OK
comment	exam has been deleted
Example	

```
{
status: "OK",
comment: "exam has been deleted"
}
```

Output if failure	
status	FAIL
comment	comments are listed below
Comments	

undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect
teacher XYZ has not undertaken subject ABC	When the pair {username, code} is invalid
server error occurred	When some server-side error occurs
Example	
{ status: "FAIL", comment: "undefined parameter: key" }	

view questions of an exam	
Call File	exam_questions.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
username	Username of the user
Password	Password of the user
category	category of the user 0: COE 1: Teacher
examId	id of the exam
Example	
key=WnXsEnsDk?username=Lewandwoski&password=Long Range Shot&category=0&examId=20	
Output if success	
status	OK
Result	Array of associative arrays. Each associative arrays have fields: id: id of the question problemStatement: correctOption:

	incorrectOption1:
	incorrectOption2:
	incorrectOption3:
	incorrectOption4:
	chapterId:
	chapterName:
	code:
	paerName:

Example

```
{
status: "OK",
result:
[
    {
        id:240
        problemStatement:html
        correctOption:html
        incorrectOption1:html
        incorrectOption2:html
        incorrectOption3:html
        chapterId:20
        chapterName:"or and"
        code:"ERD75"
        paperName:"javascript"
    },
    {
        id:40
        problemStatement:html
        correctOption:html
        incorrectOption1:html
        incorrectOption2:html
        incorrectOption3:html
        chapterId:10
        chapterName:"eagle"
        code:ERD75
        paperName:"javascript"
    }

]
```

Output if failure	
status	FAIL
comment	comments are listed below

Comments

undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect
server error occurred.	When some server-side error occurs
Example	

{
status: "FAIL",
comment: "undefined parameter: key"
}	

view students of an exam	
Call File	exam_students.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
username	Username of the COE
Password	Password of the COE
examId	id of the exam
Example	
key=WnXsEnsDk?username=Lewandwoski&password=Long Range Shot&examId=20	
Output if success	
Status	OK
Result	Array of associative arrays. Each associative arrays have fields: examId: username: name: stream: joiningYear: batchPassoutYear:

	email:
	registration:
	contactNumber:

Example

```
{
status: "OK",
result:
[
    {
        id:240
        username: "Saranya"
        name: "Saranya"
        stream: "CSE"
        joiningYear: 2017
        batchPassoutYear: 2021
        email: #@.com
        registration: "1500"
        contactNumber:"548028"
    },
    {
        id:820
        username: "Ty"
        name: "Ty Koon"
        stream: "IT"
        joiningYear: 2017
        batchPassoutYear: 2021
        email: $%@.com
        registration: "2100"
        contactNumber:"100"
    },
]
```

Output if failure

Status	FAIL
comment	comments are listed below

Comments

undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect

server error occurred.	When some server-side error occurs.
Example	

{
status: "FAIL",
comment: "undefined parameter: key"
}	

view an exam	
Call File	exam.php
Request type	http
Method	GET
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
username	username of the creator or "ALL"
examStatus	Status of the exam or "ALL" 0: created 1: live 2: submitted 3: ended (results have not been updated) 4: ended (results have been updated)
examId	id of the exam or "ALL"
code	Subject code or "ALL"
batchPassoutYear	Passout year of intended students or "ALL"
stream	stream code: or "ALL" 1: "CSE" 2: "IT" 3: "CSE" +"IT" 4: "CT" 5: "CSE+CT" 6: "IT+CT" 7: "CT"
Example	
key=WnXsEnsDk?username=Lewandwoski&examStatus=1&examId=ALL&code=ALL&batchPassoutYear=ALL&stream=6	
Output if success	
Status	OK
Result	Array of associative arrays.

	Each associative array has fields: id: id of the exam created at: (timestamp) started at: ("not started" if exam has not started yet) description: batchPassoutYear: stream: array of streams examStatus: code: created by:

Example

```
{
status: "OK",
result:
[
    {
        id:240
        created at: 48764310
        started at: not started
        description: "Test Exam"
        batchPassoutYear: 2021
        stream: ["CSE","CT"]
        examStatus: "submitted"
        code: "CSGO"
        created by: "You"
    },
    {
        id:24
        created at: 487643
        started at: 87210
        description: "Test Exam2"
        batchPassoutYear: 2021
        stream: ["CSE","CT"]
        examStatus: "live"
        code: "OP850"
        created by: "Me"
    },

]
```

Output if failure	
Status	FAIL
comment	comments are listed below

Comments	
undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect
server error occurred.	When some server-side error occurs.
Example	

`{`
`status: "FAIL",`
`comment: "undefined parameter: key"`
`}`	

Generate the result	
Call File	generate_result.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
Username	username of the COE
password	password of the COE
Example	
key=WnXsEnsDk?username=Lewandwoski&password=LongRangeShot	
Output if success	
Status	OK
comment	results have been generated
Example	

`{`
`status: "OK",`
`comment: "results have been generated"`
`}`	

Output if failure	
Status	FAIL
comment	comments are listed below

Comments	
undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect
server error occurred.	When some server-side error occurs.

Example	

{
status: "FAIL",
comment: "undefined parameter: key"
}	

View marksheet of a student	
Call File	marksheet.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded

Input	
Key	Developers key
Username	username of the COE
password	password of the COE
examId	id of the exam
studentUsername	username of the student

Example	
key=WnXsEnsDk?username=Lewandwoski&password=LongRangeShot&examId=789&studentUsername="ui"	

Output if success	
Status	OK
Result	Array of associative arrays. Each associative array has fields: question: option1: option2: option3: option4: markedOption: correctOption: status: AC for correct

	WA for incorrect
marks	Total marks obtained by the student

Example	

```
{
status: "OK",
result:
[
    {
        question: moon is bright
        option1: knights
        option2: this is option2
        option3: this is correct
        option4: option3 is wrong
        markedOption: option3
        correctOption: option4
        status: WA
    },
    {
        question: Soumya is a Dev
        option1: I am correct
        option2: I am incorrect
        option3: you are incorrect
        option4: click me
        markedOption: option1
        correctOption: option1
        status: AC
    },
],
marks: 1
}
```

Output if failure	
Status	FAIL
comment	comments are listed below

Comments	
undefined parameter XYZ	when one of the parameters are undefined
server error occurred.	When some server-side error occurs.

Example	

```
{
status: "FAIL",
comment: "undefined parameter: key"
}
```

View question paper of an exam	
Call File	question_paper.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
Username	username of the student
password	Password of the student
examId	Id of the exam
Example	
key=WnXsEnsDk?username=Lewandwoski&password=LongRangeShot&examId=3	
Output if success	
Status	OK
Result	Array of associative arrays. Each associative array has fields: question: option1: option2: option3: option4: markedOption:
Example	

```
{
status: "OK",
result:
[
    {
        question: moon is bright
        option1: knights
        option2: this is option2
        option3: this is correct
        option4: option3 is wrong
        markedOption: option3
    },
    {
        question: Soumya is a Dev
        option1: I am correct
        option2: I am incorrect
        option3: you are incorrect
        option4: click me
```

``
        markedOption: option1
    },
]
}```	

Output if failure	
Status	FAIL
comment	comments are listed below
Comments	
undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect
further submission is not allowed	When student is not allowed to submit
server error occurred.	When some server-side error occurs.
Example	

``
{
status: "FAIL",
comment: "undefined parameter: key"
}```	

Restart exam of a student	
Call File	restart_exam.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
Username	username of the COE
password	password of the COE
examId	Id of the exam
studentUsername	Username of the student
Example	
key=WnXsEnsDk?username=Lewandwoski&password=LongRangeShot&examId=789&studentUsername="BB"	
Output if success	
Status	OK

comment	exam has been restarted for the student

Example	

{
status: "OK",
comment: "exam has been restarted for the student"
}	

Output if failure	
Status	FAIL
comment	comments are listed below

Comments	
undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect
Student has never started the exam.	Advise him to start before restart
server error occurred.	When some server-side error occurs.

Example	

{
status: "FAIL",
comment: "undefined parameter: key"
}	

View marks of the students	
Call File	student_marks.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
Username	username of the COE
password	Password of the COE
examId	Id of the exam
Stream	Stream or "ALL" CSE: IT: CT:

Example	
key=WnXsEnsDk?username=Lewandwoski&password=LongRangeShot&exa mId=3&stream="CT"	
Output if success	
Status	OK
Result	Array of associative arrays. Each associative array has fields: username: name: stream: joiningYear: batchPassoutYear: email: registration: contactNumber:
Example	

```
{
status: "OK",
result:
[
    {
        username: "do or die"
        name: "noble prize"
        stream: "CSE"
        joiningYear: 2017
        batchPassoutYear: 2021
        email: er@gmail.com
        registration: 78950
        contactNumber: 882892004
    },
    {
        username: "hi"
        name: "hello"
        stream: "IT"
        joiningYear: 2015
        batchPassoutYear: 2020
        email: xxx@domain.net
        registration: 8520
        contactNumber: 20000000
    },
]
}
```

Output if failure	
Status	FAIL

comment	comments are listed below
Comments	
undefined parameter XYZ	when one of the parameters are undefined
server error occurred.	When some server-side error occurs.
Example	

{
status: "FAIL",
comment: "undefined parameter: key"
}	

Student participates in exam	
Call File	student_participate.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
Username	username of the student
password	password of the student
examId	Id of the exam
Example	
key=WnXsEnsDk?username=Lewandwoski&password=LongRangeShot&examId=789	
Output if success	
Status	OK
comment	student has participated in the exam
Example	

{
status: "OK",
comment: "student has participated in the exam"
}	

Output if failure	
Status	FAIL
comment	comments are listed below
Comments	
undefined parameter XYZ	when one of the parameters are undefined

incorrect paremeter_name XYZ	When one of the parameters are incorrect
further submission is not allowed	When student is not allowed to participate in the exam
server error occurred.	When some server-side error occurs.

Example

{
status: "FAIL",
comment: "undefined parameter: key"
}

Student submits an exam	
Call File	submit_exam.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
Username	username of the student
password	password of the student
examId	Id of the exam
Example	
key=WnXsEnsDk?username=Lewandwoski&password=LongRangeShot&examId=789	
Output if success	
Status	OK
comment	submission done
Example	

{
status: "OK",
comment: "submission done"
}	

Output if failure	
Status	FAIL
comment	comments are listed below
Comments	

undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect
further submission is not allowed	When student is not allowed to participate in the exam
server error occurred.	When some server-side error occurs.
Example	
{ status: "FAIL", comment: "undefined parameter: key" }	

Teacher submits exam	
Call File	teacher_submit_exam.php
Request type	http
Method	POST
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
username	Username of the teacher
Password	Password of the teacher
examId	id of the exam
code	subject code
Example	
key=WnXsEnsDk?username=Lewandwoski&password=Long Range Shot&examId=20&code=DOG40	
Output if success	
status	OK
comment	exam has been submitted
Example	
{ status: "OK", comment: "exam has been submitted" }	
Output if failure	
status	FAIL

comment	comments are listed below
Comments	
undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect
teacher XYZ has not undertaken subject ABC	When the pair {username, code} is invalid
server error occurred	When some server-side error occurs
Example	

```
{
status: "FAIL",
comment: "undefined parameter: key"
}
```

Check whether an exam is live or not	
Call File	is_live_exam.php
Request type	http
Method	GET
Content Type	x-www-form-urlencoded
Input	
Key	Developers key
examId	id of the exam
Example	
key=WnXsEnsDk?username=Lewandwoski&examId=20	
Output if success	
status	OK
comment	exam is running **OR** exam has ended
Example	

```
{
status: "OK",
comment: "exam has ended"
}
```

Output if failure	
status	FAIL
comment	comments are listed below
Comments	
undefined parameter XYZ	when one of the parameters are undefined
incorrect paremeter_name XYZ	When one of the parameters are incorrect
server error occurred	When some server-side error occurs
Example	

```
{
status: "FAIL",
comment: "undefined parameter: key"
}
```