**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.

(XYZ is username)

“server error occurred” – when some server-side error occurs

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 “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.

(XYZ is username)

“server error occurred” – 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

“optionNumber” is not

given.

“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

not correct.

“incorrect optionNumber: XYZ” – when the parameter “optionNumber” is

not correct.

“further submission is not allowed” – invalid attempt to save answer

“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 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&examId=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"  } | | | |