swagger

http://0.0.0.0:8000/api-docs/ap

api_key

Explore

Django REST Swagger

SAFE - Secure, Authenticated and Fast Examination

An application to conduct quizzes on mobile platform securely

Contact the developer

- ## account
  - Show/Hide
  - List Operations
  - Expand Operations
  - Raw
  - - ### post /account/passcode_reset/
      - APIView to reset passcode of account and send email

      Implementation Notes

      APIView to reset passcode of account and send email

      Response Class

      - Model
      - Model Schema

      object

      Response Content Type application/json ▼

      Try it out!

- ## api
  - Show/Hide
  - List Operations
  - Expand Operations
  - Raw
  - - ### post /api/account/login/
      - APIView for passcode based and google login

      Implementation Notes

      APIView for passcode based and google login

      Response Class

      - Model
      - Model Schema

      object

      Response Content Type application/json ▼

      Try it out!

- - **post /api/account/logout/**

    - Delete token if present and destroy session if present

    Implementation Notes

    Logout API view

    User should be logged in first to use this view

    Delete token if present and destroy session if present

    Args: request: request from client

    Response Class

    - Model
    - Model Schema

    object

    Response Content Type application/json ▾

    Try it out!

- - **post /api/account/change-password/**

    - user will send new password and that password will be set as new password

    Implementation Notes

    This view will be called to change password of user.

    user will send new password and that password will be set as new password

    Args: request: DRF request object

    Response Class

    - Model
    - Model Schema

    object

    Response Content Type application/json ▾

    Parameters

Parameter	Value	Description	Parameter Type	Data Type
newPassword	(required) Parameter content type: application/json ▾	**JSON object will contain {'newPassword':'xx'}**	body	json

    Error Status Codes

HTTP Status Code	Reason
200	password changed successfully
400	Authentication failed

    Try it out!

- - **post /api/account/google-login/**

    - Login user using google profile and return Token if authenticated else return Error

    Implementation Notes

    APIView for google login

    Login user using google profile and return Token if authenticated else return Error

    Args: request: request from client

    Response Class

    - Model
    - Model Schema

    GoogleLoginPostResponse {
    google_token (string): Token id generated by google oauthv2 api for authentication
    }

    Response Content Type application/json ▾

    Error Status Codes

HTTP Status Code	Reason
200	Authentication successful
400	Authentication failed

    Try it out!

- - **get /api/account/profile-update/**

    - APIView for get and set student details like rollno, name

    Implementation Notes

    APIView for get and set student details like rollno, name

    Response Class

    - Model
    - Model Schema

    object

    Response Content Type application/json ▾
    Try it out!

- - **post /api/account/profile-update/**

    - APIView for get and set student details like rollno, name

    Implementation Notes

    APIView for get and set student details like rollno, name

    Response Class

    - Model
    - Model Schema

object

Response Content Type application/json ▼

Try it out!

- ○
  - **get /api/quiz/{key}/info/**
    - Args:

Implementation Notes

API for getting the info of the quiz to be used by the app

Args: request: Client Request Key: instance_id Response: Json of quiz info

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)		path	string

Try it out!

- ○
  - **get /api/quiz/{key}/download/**
    - Return quiz information including questions to student with quiz

Implementation Notes

API for downloading quiz for student app. This includes settings required only for student

Return quiz information including questions to student with quiz configuration.

Args: request: Client Request key: quiz random key

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)	**Safe UUID of running quiz**	path	string

Error Status Codes

HTTP Status Code	Reason
404	Running quiz instance with key not found

Try it out!

- - **get /api/quiz/{key}/authenticate/**

    -

Implementation Notes

API for authenticate a user to start a quiz

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)	**Safe UUID of running quiz**	form	int

Error Status Codes

HTTP Status Code	Reason
404	For given quiz id there is no any quiz
403	Not authenticated for running quiz with given key

Try it out!

- - **post /api/quiz/{key}/submit/**

    - Save responses for user as final responses and logout user if applicable

Implementation Notes

View for question response submission from student.

Periodic submissions will happen through websocket and save_response method is for enabling that. Post method should be used for final submission.

Save responses for user as final responses and logout user if applicable.

Logout Condition: - Quiz has been finished at the time of submission - Multiple submissions are not allowed

Args: request: Request instance from student key: Random key for current running quiz

Response Class

- Model
- Model Schema

QuestionSubmissionPostResponse {
detail (string),

submissions_remaining (boolean, optional): If success is True, it returns the number of extra submissions remaining. It will be >= 0.,
success (boolean)
}

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)	**Unique key sent in while quiz authentication for submit**	path	string
question_responses	Parameter content type: application/json ▾		body	QuestionResponseSerializer

Try it out!

- ○
  - **[get /api/quiz/{key}/fetch/]()**
    - [Return quiz information including questions (if quiz has been started) to student with quiz]()

Implementation Notes

API for fetching quiz for student app. This includes settings required only for student

Return quiz information including questions (if quiz has been started) to student with quiz configuration.

Args: request: Client Request key: Running quiz random key

Response Class

- [Model]()
- [Model Schema]()

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)	**Safe UUID of running quiz**	path	string

Error Status Codes

HTTP Status Code	Reason
404	Running quiz instance with key not found or it is already finished
403	Not authenticated for running quiz with given key

Try it out!

- ○
  - **[get /api/quiz/{key}/auth/]()**

- Method to return 16 character random string containing passcode as well

Implementation Notes

View for authenticating a user to Running instance of Quiz

Method to return 16 character random string containing passcode as well.

App can pass an additional get parameter length to ask more than 16 keys.

Args: request: Http Request object key: safe_uuid for running quiz

Response Class

- Model
- Model Schema

RunningQuizAuthenticationGetResponse {
detail (string): Detailed message of response.,
keyboard (string): If success is True, and passcode is required, then a string of unicodes will be returned. Otherwise None will be returned,
success (boolean): If request is successful, i.e. all parameters are valid.,
keystate (string): Unique key to submit response.
}

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)	**Safe UUID of running quiz**	path	string

Error Status Codes

HTTP Status Code	Reason
404	Running quiz instance with key not found or it is already finished

Try it out!

- **post /api/quiz/{key}/auth/**

  - Check if user has posted correct credentials to connect to a running quiz instance

Implementation Notes

View for authenticating a user to Running instance of Quiz

Check if user has posted correct credentials to connect to a running quiz instance.

Verification is based on quiz_id, safe_uuid and passcode (if required). After successful verification, user will be marked authenticated for this running_quiz

Args: request: Request object key: safe_uuid of running quiz

Response Class

- Model
- Model Schema

RunningQuizAuthenticationPostResponse {
detail (string): Detailed message of response.,

success (boolean): If request is successful, i.e. all parameters are valid and authentication is successful.
}

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)	**Safe UUID of running quiz**	path	string
passcode		Unicode string corresponding to picture passcode if passcode is required. Otherwise Optional.	form	string

Error Status Codes

HTTP Status Code	Reason
404	Running quiz instance with key not found or it is already finished
400	Passcode is required and invalid or no passcode is given in request data

Try it out!

- - ## get /api/quiz/submissions/

    - Get method to fetch summary of past submissions

Implementation Notes

view for fetching past summary in short and concise manner.

It includes basic information about quiz, running quiz and marks only

Get method to fetch summary of past submissions

It returns latest 20 submissions in one request. Takes optional query param `page_number` which should be greater than 0.

Args: request: DRF request object

Response Class

- Model
- Model Schema

ShortSubmissionHistoryGetResponse {
results (array[QuizResponse]): list of objects containing submission data,
count (integer): Total number of submissions (This will determine number of pages),
page_size (integer): Current page size at server.,
prev (string): URL for prev 10 results (may be null),
next (string): URL for next 10 results (may be null)
}

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
page		Page number of result set (group of 20)	query	integer

Error Status Codes

HTTP Status Code	Reason
429	Too many requests (Crossed throttle limit)

Try it out!

- - ## get /api/quiz/submissions/{submission_id}/detail/

    - Return detailed result of submission for a specific running instance

Implementation Notes

Get Detailed response of a particular submission by safe_uuid

Return detailed result of submission for a specific running instance

Args: request: DRF Request object submission_id: Id of submission

```
YAMLError:
 mapping values are not allowed here
  in "", line 31, column 18:
                   1: {
                     ^
```

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
submission_id	(required)		path	string

Try it out!

- - ## get /api/quiz/submissions/{submission_id}/detailforceshow/

    - Return detailed result of submission for a specific running instance

Implementation Notes

This is a class for force showing answers, marks, responses, in the instructor interface

Return detailed result of submission for a specific running instance

Args: request: DRF Request object submission_id: Id of submission

```
YAMLError:
 mapping values are not allowed here
  in "", line 31, column 18:
```

```
1: {
 ^
```

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
submission_id	(required)		path	string

Try it out!

- post /api/quiz/submissions/{submission_id}/toggle-envelope-status/
  - APIview for toggling envelope status of a submission

Implementation Notes

APIview for toggling envelope status of a submission

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
submission_id	(required)		path	string

Try it out!

- post /api/quiz/submissions/{submission_id}/check-toggle-envelope/
  -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
submission_id	(required)		path	string

Try it out!

- - **get /api/quiz/uploadImage/**
    - View for uploading image responses from student

Implementation Notes

View for uploading image responses from student.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Try it out!

- - **post /api/quiz/uploadImage/**
    - View for uploading image responses from student

Implementation Notes

View for uploading image responses from student.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Try it out!

- - **get /api/quiz/{key}/downloadable-quizzes/**
    - Args:

Implementation Notes

API for getting the list of all downloadble zipped quizzes

Args: request: Client Request Key: course ID (course code in string) Response: List of offline quizzes in this format: [{"id":"safe_uuid","description":"description", "duration":"duration"},{},...]

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)		path	string

Try it out!

- - **[get /api/quiz/{quiz_id}/instance-list/]()**
    -

Implementation Notes

API for fetching all instances of a given quiz id

Response Class

- [Model]()
- [Model Schema]()

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)	**Safe UUID of running quiz**	path	string

Error Status Codes

HTTP Status Code	Reason
404	For given quiz id there is no any quiz
403	Not authenticated for running quiz with given key

Try it out!

- - **[get /api/quiz/course-submission-history/]()**
    - [Get method to fetch the course list for which the student have given at least one test]()

Implementation Notes

view for fetching past course summary in short and concise manner.

It returns latest 20 submissions in one request. Takes optional query param `page_number` which should be greater than 0.

Get method to fetch the course list for which the student have given at least one test.

Args: request: DRF request object

Response Class

- [Model]()
- [Model Schema]()

CourseListSubmissionHistoryGetResponse {
results (array): list of course,

count (integer): Total number of submissions (This will determine number of pages),
page_size (integer): Current page size at server.,
prev (string): URL for prev 10 results (may be null),
next (string): URL for next 10 results (may be null)
}

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
page		Page number of result set (group of 20)	query	integer

Error Status Codes

HTTP Status Code	Reason
429	Too many requests (Crossed throttle limit)

Try it out!

- ### get /api/quiz/course-submission-history/{course_id}/

  - Get method to fetch summary of past submissions for a given course

Implementation Notes

view for fetching quiz list for given course in short and concise manner.

It includes basic information about quiz, running quiz and marks only

Get method to fetch summary of past submissions for a given course

It returns latest 20 submissions in one request. Takes optional query param `page_number` which should be greater than 0.

Args: request: DRF request object course_id: course_id

Response Class

- Model
- Model Schema

QuizListSubmissionHistoryGetResponse {
results (array[QuizResponse]): list of objects containing submission data,
count (integer): Total number of submissions (This will determine number of pages),
page_size (integer): Current page size at server.,
prev (string): URL for prev 10 results (may be null),
next (string): URL for next 10 results (may be null)
}

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
course_id	(required)	**course-id for which quiz list has to be fetched**	path	integer
page		Page number of result set (group of 20)	query	integer

Error Status Codes

HTTP Status Code	Reason
429	Too many requests (Crossed throttle limit)

Try it out!

- - ### post /api/quiz/{quiz_id}/re-evaluate/

    - Post method which takes quiz id as paramter, re-evaluates the quiz

Implementation Notes

APIview for re-evaluating quizzes of a course

Post method which takes quiz id as paramter, re-evaluates the quiz Args: request: DRF request object quiz_id: quiz_id

response: status : success or failed

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- - ### get /api/quiz/{key}/fetch-remaining-time/

    -

Implementation Notes

API for fetching remaining time on a running quiz

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)	**Safe UUID of running quiz**	form	int

Error Status Codes

HTTP Status Code	Reason
404	For given quiz id there is no any quiz
403	Not authenticated for running quiz with given key

Try it out!

- ## post /api/quiz/{key}/partial_submission/
  - View for saving partial submissions for the quiz

Implementation Notes

View for saving partial submissions for the quiz

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)		path	string

Try it out!

- ## get /api/quiz/qqc/{key}/
  -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)		path	string

Try it out!

- **post /api/quiz/qqc/{key}/**
  -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)		path	string

Try it out!

- **get /api/course/{course_id}/**
  - Returns list of quizzes with basic info thereof

Implementation Notes

Return details(all quizzes) for a given course id

Returns list of quizzes with basic info thereof

Args: request: Client Request id: Course id

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
course_id	(required)	**Course id(primary key)**	path	string

Error Status Codes

HTTP Status Code	Reason
403	Not authenticated for getting quizzes information for the given course

Try it out!

- **get /api/course/{key}/fetch/**
  - Return list of running quizzes

Implementation Notes

API for fetching running quizzes of a course for student app. This includes settings required only for student

Return list of running quizzes.

Args: request: Client Request key: Course code

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
key	(required)	**Course code**	path	string

Error Status Codes

HTTP Status Code	Reason
404	Course with given code doesn't exist
403	Not authenticated for getting running quizzes information for the given course

Try it out!

- **post /api/course/register/**

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Try it out!

- **get /api/course/**
  - Returns list of courses for respective instructors

Implementation Notes

API View for getting list of available courses of the instructor (non archived)

Returns list of courses for respective instructors

Args: request: Client Request

Response Class

- Model

- Model Schema

object

Response Content Type application/json ▼

Try it out!

- - **post /api/log/write/**
    - Write logs of student activity

Implementation Notes

API for writing logs of student activity

Write logs of student activity.

Args: request: Client Request

Response Class

- Model
- Model Schema

WriteLogPostResponse {
detail (string),
success (boolean)
}

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
log_activity	Parameter content type: application/json ▼		body	JsonResponse

Try it out!

- - **get /api/organization/list/**
    - Return a JSON array of { id, name } dict

Implementation Notes

API for getting list of organizations

Return a JSON array of { id, name } dict

Args: request: drf request object

Response Class

- Model
- Model Schema

GetOrganizationsListGetResponse {
results (undefined, optional): List of dicts with keys "id", "name"
}

Response Content Type application/json ▾
Try it out!

- - **post /api/acra/log/**
    - Post method to add crash reports to log reports

Implementation Notes

Add crash reports to db

Post method to add crash reports to log reports

Args: request: DRF request object

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾
Try it out!

- - **post /api/attendance/**
    - API view to mark the attendance of a student

Implementation Notes

API view to mark the attendance of a student

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾
Try it out!

- - **post /api/attendance/attendance_list/**
    - API view to get attendance history

Implementation Notes

API view to get attendance history

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾
Try it out!

- ## consolidate_attendance
  - Show/Hide
  - List Operations
  - Expand Operations
  - Raw
  - - ### post /consolidate_attendance/
      - API interface to generate attendance and store for all the students in a course

      Implementation Notes

      API interface to generate attendance and store for all the students in a course

      Response Class

      - Model
      - Model Schema

      object

      Response Content Type application/json ▾

      Try it out!

- ## consolidate_report
  - Show/Hide
  - List Operations
  - Expand Operations
  - Raw
  - - ### post /consolidate_report/
      -

      Response Class

      - Model
      - Model Schema

      object

      Response Content Type application/json ▾

      Try it out!

- ## course
  - Show/Hide
  - List Operations
  - Expand Operations
  - Raw
  - - ### post /course/enter_classroom_quiz/
      - View for enter class room quiz

      Implementation Notes

      View for enter class room quiz

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Try it out!

- - **post /course/attendance_details/**
    - API interface to get list of attendances details of a single student along with images

Implementation Notes

API interface to get list of attendances details of a single student along with images

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Try it out!

- - **post /course/attendance_for_date/**
    -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Try it out!

- - **post /course/flag_attendance/**
    - API interface to manually mark the attendance of a student

Implementation Notes

API interface to manually mark the attendance of a student

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Try it out!

- - **post /course/add_attendance/**

- API interface to add a new attendance entry for a student

Implementation Notes

API interface to add a new attendance entry for a student

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼
Try it out!

- post /course/register-students/

  - To handle uploaded csv data, csv file contains 3 fields,

Implementation Notes

To give interface to instructors to upload CSV files and when CSV file is uploaded then register the students from that file.

To handle uploaded csv data, csv file contains 3 fields, 1st field is roll_no 2nd field is username (username is unique field, in case of ldap login username is the rollno) 3rd field is email

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼
Try it out!

- post /course/deregister-student/

  -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼
Try it out!

- post /course/edit-student/

  -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼
Try it out!

- - ### post /course/add-ta/
    - To give interface to instructors to add Ta's to the course

Implementation Notes

To give interface to instructors to add Ta's to the course

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼
Try it out!

- - ### post /course/add-taName/
    - View to attach TA name to the account

Implementation Notes

View to attach TA name to the account

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼
Try it out!

- - ### post /course/remove-ta/
    - To give interface to instructors to remove Ta's to the course

Implementation Notes

To give interface to instructors to remove Ta's to the course

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼
Try it out!

- - ### post /course/email-ta/
    - To give interface to instructors to email Ta's to the course

Implementation Notes

To give interface to instructors to email Ta's to the course

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Try it out!

- - **post /course/add-sch/**

    - To give interface to instructors to add schedule for the course

Implementation Notes

To give interface to instructors to add schedule for the course

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Try it out!

- - **post /course/remove-sch/**

    - To give interface to instructors to remove schedule for the course

Implementation Notes

To give interface to instructors to remove schedule for the course

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Try it out!

- - **post /course/add_extraClass/**

    - To give interface to instructors to add schedule for the course

Implementation Notes

To give interface to instructors to add schedule for the course

Response Class

- Model

  - Model Schema

object

Response Content Type application/json ▼

Try it out!

- - **post /course/remove-extraClass/**
    - To give interface to instructors to remove Ta's to the course

Implementation Notes

To give interface to instructors to remove Ta's to the course

TO-DO: permission and access control

Response Class

  - Model
  - Model Schema

object

Response Content Type application/json ▼

Try it out!

- - **post /course/update-wifi-info/**
    - API interface to update WIFI info of a course

Implementation Notes

API interface to update WIFI info of a course

Response Class

  - Model
  - Model Schema

object

Response Content Type application/json ▼

Try it out!

- ## organisation

  - Show/Hide
  - List Operations
  - Expand Operations
  - Raw

- - **post /organisation/add-instructor/**
    -

Response Class

  - Model
  - Model Schema

object

Response Content Type application/json ▼

Try it out!

- - **post /organisation/remove-instructor/**
    -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Try it out!

- **quiz**
  - Show/Hide
  - List Operations
  - Expand Operations
  - Raw

- - **post /quiz/{quiz_id}/status/**
    - make the quiz zip downloadable for a given quiz id

Implementation Notes

View to make quiz zip downloadable so that quiz can be attempted offline

make the quiz zip downloadable for a given quiz id. Args: request: The request object quiz_id: ID of the quiz which would be zip downloadable

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- - **get /quiz/{instance_id}/manualGrading/{question_id}/detail/**
    -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
instance_id	(required)		path	string
question_id	(required)		path	string

Try it out!

- - [**post /quiz/manualGrading/rubric/delete/**]()

    -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Try it out!

- - [**post /quiz/manualGrading/rubric/save/**]()

    -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Try it out!

- - [**post /quiz/manualGrading/rubric/saveImage/**]()

    -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Try it out!

- - [**post /quiz/manualGrading/rubric/submit/**]()

    -

Response Class

- Model

- Model Schema

object

Response Content Type application/json ▾

Try it out!

- ### post /quiz/manualGrading/rubric/update/type/

  -

  Response Class

  - Model
  - Model Schema

  object

  Response Content Type application/json ▾

  Try it out!

- ### post /quiz/manualGrading/rubric/update/

  -

  Response Class

  - Model
  - Model Schema

  object

  Response Content Type application/json ▾

  Try it out!

- ### post /quiz/new/{course_id}/

  - View for a new quiz

  Implementation Notes

  View for a new quiz. Creates a new quiz in database and then redirects to the corresponding edit quiz page

  Response Class

  - Model
  - Model Schema

  object

  Response Content Type application/json ▾

  Parameters

Parameter	Value	Description	Parameter Type	Data Type
course_id	(required)		path	string

  Try it out!

- ### post /quiz/upload-questions/

- Function to handle the uploaded quiz

Implementation Notes

Function to handle the uploaded quiz :param: rest framework Request by user(containing-- quiz_id and quiz_file) :return: success if file doesn't contain errors and is uploaded successfully, else list of errors as json response.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾
Try it out!

- - **post /quiz/delete-all/**

  -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾
Try it out!

- # quiz_manual_mark

  - Show/Hide
  - List Operations
  - Expand Operations
  - Raw

  - - **post /quiz_manual_mark/**

    -

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾
Try it out!

- # validate_tags

  - Show/Hide
  - List Operations
  - Expand Operations
  - Raw

  - - **post /validate_tags/**

- API interface to check quiz tags

Implementation Notes

API interface to check quiz tags

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Try it out!

- ## web_api

  - Show/Hide
  - List Operations
  - Expand Operations
  - Raw

  - ### get /web_api/instructor/get_course/{course_id}/

    - The course detail template of the dashboard page

Implementation Notes

The course detail template of the dashboard page. Permissions are used for checking if the user has access for viewing and editing stuff of a particular course.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
course_id	(required)		path	string

Try it out!

  - ### get /web_api/instructor/get_archived_courses/

    - API View for getting archived courses for the user

Implementation Notes

API View for getting archived courses for the user

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼
Try it out!

- - **get /web_api/instructor/get_running_instances/**

    - API View for getting list of Running instances for the instructor

Implementation Notes

API View for getting list of Running instances for the instructor

Response Class

- Model
- Model Schema

RunningQuizSerializer {
id (integer),
safe_uuid (string),
quiz (QuizSerializer),
start_time (string),
end_time (string)
}
QuizSerializer {
id (integer),
name (string): Short name for the quiz,
description (string): Description for the quiz,
course (CourseSerializer)
}
CourseSerializer {
id (integer),
name (string),
code (string),
image (string),
url (string)
}

Response Content Type application/json ▼
Try it out!

- - **get /web_api/instructor/get_recent_quiz/**

    - API View for getting list of recently modified Quiz for the given instructor

Implementation Notes

API View for getting list of recently modified Quiz for the given instructor

Response Class

- Model
- Model Schema

QuizSerializer {
id (integer),
name (string): Short name for the quiz,
description (string): Description for the quiz,
course (CourseSerializer)
}
CourseSerializer {
id (integer),

name (string),
code (string),
image (string),
url (string)
}

Response Content Type application/json ▼
Try it out!

- ○
  - post /web_api/instructor/create_course/

    - Args:

    Implementation Notes

    API for creating a course.

    Args: request: The request object

    Returns: Returns serialized course created. ResponseForbidden if it is forbidden. Or JSON of errors returned by the form

    Response Class

    - Model
    - Model Schema

    object

    Response Content Type application/json ▼
    Try it out!

- ○
  - get /web_api/course/{course_id}/get_past_instances/

    - API View for getting list of Running instances for the course

    Implementation Notes

    API View for getting list of Running instances for the course

    Response Class

    - Model
    - Model Schema

    RunningQuizSerializer {
    id (integer),
    safe_uuid (string),
    quiz (QuizSerializer),
    start_time (string),
    end_time (string)
    }
    QuizSerializer {
    id (integer),
    name (string): Short name for the quiz,
    description (string): Description for the quiz,
    course (CourseSerializer)
    }
    CourseSerializer {
    id (integer),
    name (string),
    code (string),
    image (string),

url (string)
}

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
course_id	(required)		path	string

Try it out!

- - [post](#) [/web_api/course/{id}/edit/](#)
    - [Args:](#)

Implementation Notes

API for editing a course.

Args: request: The request object

Returns: Returns serialized course. ResponseForbidden if it is forbidden. Or JSON of errors returned by the form

Response Class

- [Model](#)
- [Model Schema](#)

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)		path	string

Try it out!

- - [get](#) [/web_api/course_list/](#)
    - [API View for getting list of available courses of the instructor (non archived)](#)

Implementation Notes

API View for getting list of available courses of the instructor (non archived)

Response Class

- [Model](#)
- [Model Schema](#)

CourseSerializer {
id (integer),
name (string),
code (string),
image (string),
url (string)
}

Response Content Type application/json ▼

Try it out!

- ○
  - **post /web_api/course/{id}/change_reg_code/**
    - Args:

Implementation Notes

API for regenerating registration code for the course.

Args: request: The request object

Returns: Returns new registration code.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)		path	string

Try it out!

- ○
  - **post /web_api/quiz/delete/{quiz_id}/**
    - Method to delete quiz from database

Implementation Notes

View to delete a quiz. Delete quiz and corresponding data

Method to delete quiz from database

If configuration of this quiz is not referenced with any other quiz, delete configuration as well

Args: request: DRF Request instance quiz_id: id of the quiz needs to be deleted

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- - **get /web_api/quiz/clone/{quiz_id}/{course_id}/**
    - Method to get status of cloning from task id

Implementation Notes

View to clone a quiz

Method to get status of cloning from task id

Args: request: DRF request object *args: Additional Args* *kwargs: Additional keyword arguments

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string
course_id	(required)		path	string

Try it out!

- - **post /web_api/quiz/clone/{quiz_id}/{course_id}/**
    - Method to schedule cloning task of a quiz after permission check for user

Implementation Notes

View to clone a quiz

Method to schedule cloning task of a quiz after permission check for user

Args: request: DRF Request object quiz_id: id of quiz to be cloned

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string
course_id	(required)		path	string

Try it out!

- - **get /web_api/configuration/{quiz_id}/**

- Get the quiz configuration for a quiz object

Implementation Notes

View for setting a quiz's configuration.

Get the quiz configuration for a quiz object.

Args: request: The request object quiz_id: Quiz for which configuration should be sent

Returns: JSON object with the details of the quiz configuration.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- - **post /web_api/configuration/{quiz_id}/**

- API for setting the quiz configuration

Implementation Notes

View for setting a quiz's configuration.

API for setting the quiz configuration.

Args: request: The request object quiz_id: The quiz id for which the configuration should be changed

Returns: JSON object with the details of the quiz configuration after the updates.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- - **post /web_api/question_module/**

- Create a new question module

Implementation Notes

API's to add a new active question module or editing existing question module

Only active question modules (that have sub questions under it) are created and edited with this API

Create a new question module

Response Class

- Model
- Model Schema

QuestionModuleSerializer {
id (integer),
module_number (string),
description (string),
quiz (string),
randomize_questions (boolean),
is_active (boolean)
}

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
module_number			form	string
description			form	string
quiz	(required)		form	string
randomize_questions	▼		form	boolean
is_active	▼		form	boolean

Try it out!

- ## put /web_api/question_module/

  - Update question module

Implementation Notes

API's to add a new active question module or editing existing question module

Only active question modules (that have sub questions under it) are created and edited with this API

Update question module

Response Class

- Model
- Model Schema

QuestionModuleSerializer {
id (integer),
module_number (string),
description (string),
quiz (string),
randomize_questions (boolean),
is_active (boolean)
}

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
module_number			form	string
description			form	string
quiz	(required)		form	string
randomize_questions	▼		form	boolean
is_active	▼		form	boolean

Try it out!

- - **[delete /web_api/question_module/]**
    - Delete question module

Implementation Notes

API's to add a new active question module or editing existing question module

Only active question modules (that have sub questions under it) are created and edited with this API

Delete question module

Response Class

- Model
- Model Schema

QuestionModuleSerializer {
id (integer),
module_number (string),
description (string),
quiz (string),
randomize_questions (boolean),
is_active (boolean)
}

Response Content Type application/json ▼

Try it out!

- - **[post /web_api/question_module/{pk}/]**
    - Create a new question module

Implementation Notes

API's to add a new active question module or editing existing question module

Only active question modules (that have sub questions under it) are created and edited with this API

Create a new question module

Response Class

- Model
- Model Schema

QuestionModuleSerializer {
id (integer),
module_number (string),

description (string),
quiz (string),
randomize_questions (boolean),
is_active (boolean)
}

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
pk	(required)		path	string
module_number			form	string
description			form	string
quiz	(required)		form	string
randomize_questions	▼		form	boolean
is_active	▼		form	boolean

Try it out!

- **put /web_api/question_module/{pk}/**
  - Update question module

Implementation Notes

API's to add a new active question module or editing existing question module

Only active question modules (that have sub questions under it) are created and edited with this API

Update question module

Response Class

- Model
- Model Schema

QuestionModuleSerializer {
id (integer),
module_number (string),
description (string),
quiz (string),
randomize_questions (boolean),
is_active (boolean)
}

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
pk	(required)		path	string
module_number			form	string
description			form	string
quiz	(required)		form	string
randomize_questions	▼		form	boolean
is_active	▼		form	boolean

Try it out!

- - **delete /web_api/question_module/{pk}/**

    - Delete question module

Implementation Notes

API's to add a new active question module or editing existing question module

Only active question modules (that have sub questions under it) are created and edited with this API

Delete question module

Response Class

- Model
- Model Schema

QuestionModuleSerializer {
id (integer),
module_number (string),
description (string),
quiz (string),
randomize_questions (boolean),
is_active (boolean)
}

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
pk	(required)		path	string

Try it out!

- - **post /web_api/fib_check_answers/**

    - API View for parsing the correct answers in FIB questions

Implementation Notes

API View for parsing the correct answers in FIB questions.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼
Try it out!

- - **post /web_api/question/{quiz_id}/**

    - Create a module, add the question to it and then return empty string as a sign of success

Implementation Notes

API View for adding new questions.

Create a module, add the question to it and then return empty string as a sign of success. If there are errors, we are returning a 400 bad request. Args: request: The Request object quiz_id: ID of the quiz in which the current question must be inserted.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- **get /web_api/question_list/{quiz_id}/**
  - Act as if we are sending a post request

Implementation Notes

Return all the questions as a json format for displaying.

Act as if we are sending a post request.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- **post /web_api/question_list/{quiz_id}/**
  - Return the list of questions in a particular quiz

Implementation Notes

Return all the questions as a json format for displaying.

Return the list of questions in a particular quiz.

Args: request: The HTTP request object quiz_id: Quiz ID for which questions are required.

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- **post /web_api/delete/question/{question_id}/**

  - Delete a question

Implementation Notes

API for deleting a question.

Delete a question.

Args: request: The request object with the user data question_id: The question which should be deleted

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
question_id	(required)		path	string

Try it out!

- **post /web_api/question_module/{module_id}/reorder/**

  - API for Reordering Question Module

Implementation Notes

API for Reordering Question Module

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
module_id	(required)		path	string

Try it out!

- - **post /web_api/question/{id}/reorder/**

    - API for Reordering Question inside Question Module

Implementation Notes

API for Reordering Question inside Question Module

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)		path	string

Try it out!

- - **post /web_api/question/{question_id}/add_image/**

    - Post method of upload image to a question with id as question_id

Implementation Notes

APIView for uploading images to a question

Request should be sent as multi-part form data as files cannot be uploaded through AJAX. Hence MultiPartParser is used instead of default JSONParser

Post method of upload image to a question with id as question_id

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
question_id	(required)		path	string

Try it out!

- - **post /web_api/quiz/instances/qstart/{quiz_id}/**

- Start a quiz instantly given a quiz id

Implementation Notes

View for starting a quiz instantly

Start a quiz instantly given a quiz id.

Args: request: The request object quiz_id: ID of the quiz for which a quiz should be started

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- ### post /web_api/quiz/instances/schedule/{quiz_id}/

  - Schedule a quiz for given quiz id

Implementation Notes

View for scheduling Quiz

Schedule a quiz for given quiz id

Args: request: DRF Request quiz_id: ID of the quiz for which quiz should be scheduled

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- ### post /web_api/quiz/instance/{instance_id}/delete/

  - Post view to delete an instance from UI and database

Implementation Notes

View to delete an instance. It contains post method only

Post view to delete an instance from UI and database

Args: request: Django request object instance_id: Id of instance to be deleted

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
instance_id	(required)		path	string

Try it out!

- - **post /web_api/quiz/{quiz_id}/create-offline-instance/**

- make the quiz zip downloadable for a given quiz id

Implementation Notes

View to make quiz zip downloadable so that quiz can be attempted offline

make the quiz zip downloadable for a given quiz id. Args: request: The request object quiz_id: ID of the quiz which would be zip downloadable

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- - **post /web_api/quiz/instance/{instance_id}/start/**

- Args:

Implementation Notes

View to start offline instance

Args: request: The request object instance_id: Id of the instance which will get started now

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
instance_id	(required)		path	string

Try it out!

- - [post /web_api/quiz/instance/{instance_id}/stop_submissions/]()
    - [Post view to stop or start submissions of an instance]()

Implementation Notes

View to delete an instance. It contains post method only

Post view to stop or start submissions of an instance

Args: request: Django request object instance_id: Id of instance to allow(not) submissions

Response Class

- [Model]()
- [Model Schema]()

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
instance_id	(required)		path	string

Try it out!

- - [post /web_api/quiz/{quiz_id}/publish-quiz/]()
    - [Args:]()

Implementation Notes

View to publish a quiz, creates a downloadable zip of quiz for app to download

Args: request: The request object quiz_id: ID of the quiz which would be published

Response Class

- [Model]()
- [Model Schema]()

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quiz_id	(required)		path	string

Try it out!

- - **post /web_api/quiz/live/summary/{instance_id}/**

    - Retrieve quiz summary details

Implementation Notes

Retrieve different details of the running quiz such as number of people logged in, remaining time, logs, etc to be displayed on the quiz live dashboard and is fetched periodically.

Retrieve quiz summary details

Args: request: The drf request object instance_id: The RunningQuiz Instance for which details are required

Response Class

- Model
- Model Schema

object

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
instance_id	(required)		path	string

Try it out!

[BASE URL: **http://0.0.0.0:8000/api-docs/api-docs**]