Pretty print ☑

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "Interviews & Group Discussions APIs",
    "description": "Welcome to the API documentation for Mercer | Mettl's Interviews and
Group Discussions. These APIs will allow you to seamlessly integrate your platforms and
tools with our system, enabling the scheduling, conducting, and managing of interviews and
group discussions. With these APIs, you can automate interview processes, fetch candidate
feedback & scores, and manage sessions efficiently.",
    "version": "241.1.6"
  },
  "servers": [
    {
      "url": "api.mettl.pro/interviews"
    }
  ],
  "tags": [
    {
      "name": "Interview APIs Controller",
      "description": "This controller serves multiple APIs allowing for different
operations related to interviews"
    },
    {
      "name": "Report APIs Controller",
      "description": "This controller serves multiple APIs allowing for different
operations related to reports"
    },
    {
      "name": "Job Profile APIs Controller",
      "description": "This controller serves multiple APIs allowing for different
operations related to job profiles"
    }
  ],
  "paths": {
    "/v1/report/action/{interviewUuid}/{candidateUuid}/{candidateStatus}": {
      "post": {
        "tags": [
          "Report APIs Controller"
        ],
        "summary": "This API can be used to change the shortlisting status of a candidate",
        "operationId": "takeAction",
        "parameters": [
          {
            "name": "interviewUuid",
            "in": "path",
            "description": "Unique identifier of the interview",
            "required": true,
            "schema": {
              "minLength": 1,
              "type": "string"
            },
            "example": "7ce6643b"
          },
          {
            "name": "candidateUuid",
            "in": "path",
            "description": "Unique identifier of the candidate",
            "required": true,
            "schema": {
              "minLength": 1,
              "type": "string"
            },
            "example": "e09d5e41"
          },
          {
            "name": "candidateStatus",
            "in": "path",
```

Pretty print

```
            "schema": {
              "type": "string",
              "enum": [
                "SHORTLISTED",
                "REJECTED",
                "ON_HOLD",
                "NOT_EVALUATED"
              ]
            }
          }
        ],
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK"
          }
        }
      }
    },
      "post": {
        "tags": [
          "Interview APIs Controller"
        ],
        "summary": "This API can be used to update the name of an interview.",
        "operationId": "updateInterviewName",
        "parameters": [
          {
            "name": "interviewUuid",
            "in": "path",
            "description": "Unique identifier for the interview",
            "required": true,
            "schema": {
              "type": "string"
            },
            "example": "5080dc7f"
          }
        ],
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "maxLength": 100,
                "minLength": 0,
                "type": "string",
                "description": "Updated name of the interview"
              }
            }
          },
          "required": true
        },
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "type": "boolean"
                }
              }
            }
          }
        }
      }
    },
    "/v1/interview/schedule": {
      "post": {
        "tags": [
          "Interview APIs Controller"
        ],
        "summary": "This API can be used to create and schedule an interview",
        "operationId": "scheduleInterviewV2",
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ScheduleInterviewApiRequest"
              }
            }
          },
          "required": true
        },
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/ScheduleInterviewApiResponse"
                }
              }
            }
          }
        }
      }
    },
    "/v1/interview/reschedule": {
      "post": {
        "tags": [
          "Interview APIs Controller"
        ],
        "summary": "This API can be used to update the scheduled date, time and time zone
of an interview",
        "operationId": "rescheduleInterview",
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/RescheduleInterviewApiRequest"
              }
            }
          },
          "required": true
        },
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK"
          }
        }
      }
    },
    "/v1/interview/edit": {
      "post": {
        "tags": [
          "Interview APIs Controller"
        ],
        "summary": "This API can be used to edit an interview.",
        "operationId": "editInterviewV2",
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/EditInterviewApiRequest"
              }
            }
          },
          "required": true
        },
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/ScheduleInterviewApiResponse"
                }
              }
            }
          }
        }
      }
    },
    "/v1/report/summary/{interviewUuid}": {
      "get": {
        "tags": [
          "Report APIs Controller"
        ],
        "summary": "This API can be used to fetch the summary of an interview",
        "operationId": "getSummary",
        "parameters": [
          {
            "name": "interviewUuid",
            "in": "path",
            "description": "Unique identifier of the interview",
            "schema": {
              "type": "string"
            },
            "example": "7ce6643b"
          }
        ],
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewSummaryResponse"
                }
              }
            }
          }
        }
```

```
    },
    "/v1/report/recording/{interviewUuid}": {
      "get": {
        "tags": [
          "Report APIs Controller"
        ],
        "summary": "This API can be used to fetch the recordings of an interview",
        "operationId": "getInterviewRecordings",
        "parameters": [
          {
            "name": "interviewUuid",
            "in": "path",
            "description": "Unique identifier of the interview",
            "required": true,
            "schema": {
              "type": "string"
            },
            "example": "7ce6643b"
          }
        ],
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "application/json": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/MeetingSummaryResponse"
                  }
                }
              }
            }
          }
        }
      }
    },
    "/v1/report/logs/{interviewUuid}": {
      "get": {
        "tags": [
          "Report APIs Controller"
        ],
        "summary": "This API can be used to fetch the logs of an interview",
        "operationId": "getInterviewLogs",
        "parameters": [
          {
            "name": "interviewUuid",
            "in": "path",
            "description": "Unique identifier of the interview",
            "required": true,
            "schema": {
              "type": "string"
            },
            "example": "7ce6643b"
          }
        ],
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "application/json": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/InterviewActivityLogSummaryResponse"
                  }
                }
              }
            }
          }
        }
      }
    },
    "/v1/report/feedback/{interviewUuid}": {
      "get": {
        "tags": [
          "Report APIs Controller"
        ],
        "summary": "This API can be used to fetch the detailed feedback and scores for an
interview",
        "operationId": "getInterviewFeedback",
        "parameters": [
          {
            "name": "interviewUuid",
            "in": "path",
            "description": "Unique identifier of the interview",
            "required": true,
            "schema": {
              "type": "string"
            },
            "example": "7ce6643b"
          }
        ],
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewFeedback"
                }
              }
            }
          }
        }
      }
    },
    "/v1/jobProfile/{uuid}": {
      "get": {
        "tags": [
          "Job Profile APIs Controller"
        ],
        "summary": "This API can be used to fetch all details of a job profile",
        "operationId": "getJobProfileDetail",
        "parameters": [
          {
            "name": "uuid",
            "in": "path",
            "description": "Unique id of job profile",
            "required": true,
            "schema": {
              "type": "string"
            },
            "example": "7ce6643b-e287-44c3-93ca-5d31ce8dc56d"
          }
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/JobProfileDto"
                }
              }
            }
          }
        }
      }
    },
    "/v1/jobProfile/all": {
      "get": {
        "tags": [
          "Job Profile APIs Controller"
        "summary": "This API can be used to fetch job profiles in your account in a
paginated form with limited data. To fetch complete details of a job profile please use
getJobProfileDetail method.",
        "operationId": "getAllJobProfiles",
        "parameters": [
          {
            "name": "pageNumber",
            "in": "query",
            "description": "Page Number denotes which next data to be fetched. For example
if page size is 15 and page no 3 the it means 31 to 45 data will be fetched.",
            "required": false,
            "schema": {
              "type": "string"
            },
            "example": 1
          },
          {
            "name": "pageSize",
            "in": "query",
            "description": "No of records to be fetched in a request",
            "required": false,
            "schema": {
              "type": "string"
            },
            "example": 20
          },
          {
            "name": "candidateDetails",
            "in": "query",
            "description": "Candidate email or candidate name to search for",
            "required": false,
            "schema": {
              "type": "string"
            },
            "example": "john.doe@example.com"
          },
          {
            "name": "jobProfileName",
            "in": "query",
            "description": "Job profile name that you want to search for",
            "required": false,
            "schema": {
              "type": "string"
            },
            "example": "Software Developer"
          }
        ],
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/JobProfilePage"
                }
              }
            }
          }
        }
      }
    },
    "/v1/interview": {
      "get": {
        "tags": [
          "Interview APIs Controller"
        ],
        "summary": "This API can be used to fetch all interviews in your account in a
paginated form",
        "operationId": "getAllInterviews",
        "parameters": [
          {
            "name": "candidateEmailId",
            "in": "query",
            "description": "Candidate email to search for",
            "required": false,
            "schema": {
              "type": "string"
            },
            "example": "john.doe@example.com"
          },
          {
            "name": "pageNumber",
            "in": "query",
            "description": "Starts from 0",
            "required": false,
            "schema": {
              "type": "integer",
            },
            "example": 0
          },
          {
            "name": "pageSize",
            "in": "query",
            "description": "No of records to be fetched in a request",
            "required": false,
            "schema": {
              "type": "integer",
              "format": "int32"
            },
            "example": 15
          },
          {
            "name": "interviewType",
            "in": "query",
            "description": "Type of interview",
            "required": false,
            "schema": {
              "type": "string",
              "default": "INTERVIEW",
              "enum": [
                "INTERVIEW",
                "GROUP_DISCUSSION",
                "ROLE_PLAY",
                "ORIENTATION"
              ]
            },
            "example": "INTERVIEW"
          }
        ],
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewPage"
                }
              }
            }
          }
        }
      }
    },
    "/v1/interview/timezones": {
      "get": {
        "tags": [
          "Interview APIs Controller"
        ],
        "summary": "This API can be used to get the list of the supported time zones",
        "description": "Example - \"Asia/Kolkata\": \"GMT+05:30 Asia/Kolkata\",
\"America/Toronto\": \"GMT-04:00 America/Toronto\", \"Europe/Moscow\": \"GMT+03:00
Europe/Moscow\", \"Pacific/Funafuti\": \"GMT+12:00 Pacific/Funafuti\", \"UTC\": \"GMT+00:00
UTC\", \"WET\": \"GMT+01:00 WET\"",
        "operationId": "getAllTimezones",
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
```

Pretty print

```
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "application/json": {
                "schema": {
                  "type": "object",
                  "additionalProperties": {
                    "type": "string"
                  }
                }
              }
            }
          }
        }
      }
    },
    "/v1/interview/detail/{interviewUuid}": {
      "get": {
        "tags": [
          "Interview APIs Controller"
        ],
        "summary": "This API can be used to fetch all details of an interview",
        "operationId": "getInterviewDetail",
        "parameters": [
          {
            "name": "interviewUuid",
            "in": "path",
            "description": "Unique identifier for the interview",
            "required": true,
            "schema": {
              "minLength": 1,
              "type": "string"
            },
            "example": "5080dc7f"
          }
        ],
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
```

Pretty print

```
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewResponse"
                }
              }
            }
          }
        }
      }
    },
    "/v1/interview/cancel/{interviewUuid}": {
      "delete": {
        "tags": [
          "Interview APIs Controller"
        ],
        "summary": "This API can be used to cancel an interview",
        "operationId": "cancelInterview",
        "parameters": [
          {
            "name": "interviewUuid",
            "in": "path",
            "description": "Unique identifier for the interview",
            "required": true,
            "schema": {
              "type": "string"
            },
            "example": "5080dc7f"
          {
            "name": "isSendEmail",
            "in": "query",
            "description": "Should send email",
            "required": true,
            "schema": {
              "type": "boolean"
            },
            "example": true
          }
        ],
        "responses": {
          "500": {
            "description": "Internal Server Error",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "400": {
            "description": "Bad Request",
            "content": {
              "*/*": {
                "schema": {
                  "$ref": "#/components/schemas/InterviewError"
                }
              }
            }
          },
          "200": {
            "description": "OK"
          }
        }
      }
    }
```

Pretty print

```
  "components": {
    "schemas": {
      "CandidateBasicDetails": {
        "type": "object",
        "properties": {
          "firstName": {
            "type": "string",
            "description": "First name of candidate",
            "example": "John"
          },
          "lastName": {
            "type": "string",
            "description": "Last name of candidate",
            "example": "Doe"
          },
          "email": {
            "type": "string",
            "description": "Email of candidate",
            "example": "john.doe@example.com"
          },
          "participantUuid": {
            "type": "string",
            "description": "Unique id of candidate",
            "example": "7ce6643b-e287-44c3-93ca-5d31ce8dc56d"
          },
          "participantStatus": {
            "type": "string",
            "description": "Current status of candidate",
            "example": "INVITED",
            "enum": [
              "INVITED",
              "WAITING",
              "JOINED_INTERVIEW",
              "LEFT_INTERVIEW",
              "FEEDBACK_SUBMITTED",
              "AWAITING_FEEDBACK",
              "NOT_ACTIVE",
              "DENIED",
              "ALLOWED",
              "DISCONNECTED",
              "LEFT_LOBBY"
            ]
          },
          "resume": {
            "$ref": "#/components/schemas/FileDetailsResponse"
          }
        },
        "description": "List of candidates in the interview"
      },
      "CandidateCompetencyEvaluationSummary": {
        "type": "object",
        "properties": {
          "competencyName": {
            "type": "string",
            "description": "Name of the competency",
            "example": "Problem Solving"
          },
          "averageScore": {
            "type": "number",
            "description": "Average score obtained by candidate",
            "format": "float",
            "example": 12.3
          },
          "maxScore": {
            "type": "number",
            "description": "Max score of the competency",
            "format": "float",
            "example": 20
          "scoringType": {
            "type": "string",
            "description": "Scoring type of competency",
            "example": "STAR",
            "enum": [
              "STAR",
              "SCORE"
            ]
          },
          "interviewerCompetencyFeedbackDtoList": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/InterviewerCompetencyFeedbackResponse"
            }
          },
          "notAssessed": {
            "type": "boolean",
            "description": "If the competency is not assessed by all interviewers",
            "example": false
          },
          "candidateSubCompetencyEvaluationSummaryDtoList": {
            "type": "array",
            "description": "Evaluation summary for sub competencies if there are any",
            "items": {
              "$ref": "#/components/schemas/CandidateCompetencyEvaluationSummary"
            }
          }
        }
      },
      "CandidateEvaluationSummaryResponse": {
        "type": "object",
        "properties": {
          "interviewerUuid": {
            "type": "string",
            "description": "Unique id of interview",
            "example": "7e30d29a"
          },
          "interviewerName": {
            "type": "string",
            "description": "Name of interviewer",
            "example": "Interviewer 1"
          },
          "interviewerStatus": {
            "type": "string",
            "description": "Interviewer status in interview",
            "example": "FEEDBACK_SUBMITTED",
            "enum": [
              "INVITED",
              "WAITING",
              "JOINED_INTERVIEW",
              "LEFT_INTERVIEW",
              "FEEDBACK_SUBMITTED",
              "AWAITING_FEEDBACK",
              "NOT_ACTIVE",
              "DENIED",
              "ALLOWED",
              "DISCONNECTED",
              "LEFT_LOBBY"
            ]
          },
          "interviewerJoinedDuration": {
            "type": "integer",
            "description": "Joined duration interviewer in minutes",
            "format": "int64",
            "example": 30
          },
          "evaluationResult": {
            "type": "string",
            "example": "RECOMMENDED",
            "enum": [
              "RECOMMENDED",
              "NOT_SURE",
              "NOT_RECOMMENDED",
              "NOT_EVALUATED"
            ]
          },
          "feedback": {
            "type": "string",
            "description": "Feedback by interviewer",
            "example": "Feedback by interviewer"
          },
          "skipEvaluation": {
            "type": "boolean",
            "description": "If the evaluation is skipped by interviewer",
            "example": false
          },
          "skipEvaluationReason": {
            "type": "string",
            "description": "Reason for skipping evaluation",
            "example": "Reason for skipping evaluation"
          }
        }
      },
      "CandidateFeedbackAndResultSummary": {
        "type": "object",
        "properties": {
          "candidate": {
            "$ref": "#/components/schemas/CandidateBasicDetails"
          },
          "feedback": {
            "$ref": "#/components/schemas/FeedbackSummaryResponse"
          },
          "candidateResultSummaryResponse": {
            "$ref": "#/components/schemas/CandidateResultSummaryResponse"
          }
        }
      },
      "CandidateInfo": {
        "required": [
          "email",
          "firstName"
        ],
        "type": "object",
        "properties": {
          "email": {
            "type": "string",
            "description": "Email of the user.\nMax length: 200 characters",
            "example": "John.doe@example.com"
          },
          "firstName": {
            "type": "string",
            "description": "First name of the user.\nMax length: 100 characters",
            "example": "John"
          },
          "lastName": {
            "type": "string",
            "description": "Last name of the user.\nMax length: 100 characters",
            "example": "Doe"
          }
        },
        "description": "List of candidates for the interview"
      },
      "CandidateResponse": {
        "type": "object",
        "properties": {
          "uuid": {
            "description": "Unique id of the participant",
            "example": "d8ccc0a"
          },
          "status": {
            "type": "string",
            "description": "Current status of the participant",
            "example": "INVITED",
            "enum": [
              "INVITED",
              "WAITING",
              "JOINED_INTERVIEW",
              "LEFT_INTERVIEW",
              "FEEDBACK_SUBMITTED",
              "AWAITING_FEEDBACK",
              "NOT_ACTIVE",
              "DENIED",
              "ALLOWED",
              "DISCONNECTED",
              "LEFT_LOBBY"
            ]
          },
          "user": {
            "$ref": "#/components/schemas/UserResponse"
          },
          "invited": {
            "type": "boolean",
            "description": "If the participant is invited for the interview",
            "example": true
          },
          "joinedOnce": {
            "type": "boolean",
            "description": "If the participant has joined once in the interview",
            "example": true
          },
          "joinedDuration": {
            "type": "integer",
            "description": "Duration of participant in the interview",
            "format": "int64",
            "example": 17465881
          },
          "resumeUrl": {
            "type": "string",
            "description": "Url of the uploaded candidate resume",
            "example": "https://resume-url.com"
          },
          "resumeFileName": {
            "type": "string",
            "description": "Name of the uploaded candidate resume",
            "example": "john_resume.pdf"
          },
          "candidateStatus": {
            "type": "string",
            "description": "Current status of the candidate",
            "example": "SHORTLISTED",
            "enum": [
              "SHORTLISTED",
              "REJECTED",
              "ON_HOLD",
              "NOT_EVALUATED"
            ]
          },
          "averageMarksObtained": {
            "type": "number",
            "description": "Average marks obtained by candidate",
            "format": "float",
            "example": 78.3
          },
          "maxMarks": {
            "description": "Max marks",
            "format": "float",
            "example": 100
          },
          "notAssessed": {
            "type": "boolean",
            "description": "Flag telling if the candidate is not assessed",
            "example": true
          }
        }
      },
      "CandidateResultSummaryResponse": {
        "type": "object",
        "properties": {
          "totalAverageScore": {
            "type": "number",
            "description": "Total average score obtained by Candidate",
            "format": "float",
            "example": 76.3
          },
          "totalMaxScore": {
            "type": "number",
            "description": "Total max score",
            "format": "float",
            "example": 100
          },
          "notAssessed": {
            "type": "boolean",
            "description": "Whether evaluations are not assessed by interviewers",
            "example": false
          },
          "competencySummaryList": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/CandidateCompetencyEvaluationSummary"
            }
          }
        },
        "description": "If there is any job profile in interview, then this will provide
overall score and summary of evaluations provided by all interviewers for competencies"
      },
      "CompetencyDto": {
        "type": "object",
        "properties": {
          "uuid": {
            "type": "string",
            "description": "Unique id of a job profile",
            "example": "b57371ca-d4fb-414e-bdf3-ad337531e5ae"
          },
          "name": {
            "type": "string",
            "description": "Name of the competency",
            "example": "Tools & Technologies"
          },
          "description": {
            "type": "string",
            "description": "Description of the competency",
            "example": "Sufficient understanding of how different Cloud Service providers
work is very essential. Cloud Service providers offer end-to-end services like compute,
storage, databases, ML and Migration, which is why almost everything that is related to
cloud computing is catered by them making this understanding a very vital skill."
          },
          "scoringType": {
            "type": "string",
            "description": "Scoring type of the competency",
            "example": "STAR",
            "enum": [
              "STAR",
              "STAR",
              "SCORE"
            ]
          },
          "marks": {
            "type": "number",
            "description": "Marks for the competency",
            "format": "float",
            "example": 20
          },
          "subCompetencies": {
            "type": "array",
            "description": "Sub competencies added in this competency",
            "items": {
              "$ref": "#/components/schemas/CompetencyDto"
            }
          },
          "calculationMethod": {
            "type": "string",
            "description": "If subCompetencies available total score for this competency will be calculated from subCompetencies as SUM or AVERAGE. If subCompetencies are not available this field will be ignored.",
            "example": "SUM",
            "enum": [
              "SUM",
              "AVERAGE",
              "SUM",
              "AVERAGE"
            ]
          },
          "type": {
            "type": "string",
            "description": "If the competency is created by client itself or admin",
            "example": "ADMIN",
            "enum": [
              "ADMIN",
              "SELF",
              "ADMIN",
              "SELF"
            ]
          }
        },
        "description": "List of competencies in this job profile"
      },
      "EditInterviewApiRequest": {
        "required": [
          "candidates",
          "interviewUuid",
          "interviewers"
        ],
        "type": "object",
        "properties": {
          "interviewUuid": {
            "type": "string",
            "description": "Unique identifier for the interview you want to edit.",
            "example": "4de94450"
          },
          "name": {
            "type": "string",
            "description": "Name of the interview. Only applicable when your interview type is 'GROUP_DISCUSSION'",
            "example": "Interview with john"
          },
          "interviewers": {
            "type": "array",
            "description": "List of interviewers for the interview",
            "items": {
              "$ref": "#/components/schemas/InterviewerInfo"
```

Pretty print

```
          },
          "candidates": {
            "type": "array",
            "description": "List of candidates for the interview",
            "items": {
              "$ref": "#/components/schemas/CandidateInfo"
            }
          },
          "jobProfileUuid": {
            "type": "string",
            "description": "Unique id of job profile",
            "example": "7ce6643b-e287-44c3-93ca-5d31ce8dc56d"
          },
          "settings": {
            "$ref": "#/components/schemas/InterviewSettingRequest"
          }
        }
      },
      "FeedbackSummaryResponse": {
        "type": "object",
        "properties": {
          "candidateStatus": {
            "type": "string",
            "description": "Status of candidate",
            "example": "SHORTLISTED",
            "enum": [
              "SHORTLISTED",
              "REJECTED",
              "ON_HOLD",
              "NOT_EVALUATED",
              "SHORTLISTED",
              "REJECTED",
              "ON_HOLD",
              "NOT_EVALUATED"
            ]
          },
          "evaluationResultVsCount": {
            "type": "object",
            "additionalProperties": {
              "type": "integer",
              "description": "Recommendation by interviewer counts",
              "format": "int64"
            },
            "description": "Recommendation by interviewer counts",
            "example": " {\n            \"RECOMMENDED\": 1,\n
\"NOT_RECOMMENDED\": 0,\n          \"NOT_SURE\": 1,\n            \"NOT_EVALUATED\": 1,\n
}"
          },
          "candidateEvaluationSummaryDtoList": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/CandidateEvaluationSummaryResponse"
            }
          }
        },
        "description": "This will provide summary of overall evaluation provided by all
interviewers"
      },
      "FileDetailsResponse": {
        "type": "object",
        "properties": {
          "uuid": {
            "type": "string",
            "description": "Unique id of file",
            "example": "11239494-b5f0-440c-ab97-d4748ae7fd9b"
          },
          "name": {
            "type": "string",
```

Pretty print

```
          "example": "candidate-resume.pdf"
          },
          "url": {
            "type": "string",
            "description": "Url of the file",
            "example": "https://file-url.com"
          }
        }
      },
      "InterviewActivityLogSummaryResponse": {
        "type": "object",
        "properties": {
          "createdOn": {
            "type": "integer",
            "description": "Time when this log was created",
            "format": "int64",
            "example": 12312378909
          },
          "eventType": {
            "type": "string",
            "description": "Event type of log",
            "example": "INTERVIEW_STATUS_EVENT",
            "enum": [
              "PARTICIPANT_STATUS_EVENT",
              "QUESTION_STATUS_EVENT",
              "INTERVIEW_LOCK_EVENT",
              "INTERVIEW_STATUS_EVENT",
              "PARTICIPANT_DEVIATION_EVENT",
              "PARTICIPANT_REMINDER_EVENT",
              "QUESTION_LOCK_EVENT",
              "QUESTION_TITLE_EVENT",
              "QUESTION_DESCRIPTION_EVENT",
              "INTERVIEWSETTINGS_EVENT"
            ]
          },
          "currentActivityType": {
            "type": "string",
            "description": "Activity performed for the event type.",
            "example": "INTERVIEW_STATUS_EVENT",
            "enum": [
              "PARTICIPANT_INVITED",
              "PARTICIPANT_WAITING",
              "PARTICIPANT_ALLOWED",
              "PARTICIPANT_DENIED",
              "PARTICIPANT_JOINED",
              "PARTICIPANT_LEFT",
              "PARTICIPANT_DISCONNECTED",
              "PARTICIPANT_LEFT_LOBBY",
              "PARTICIPANT_AWAITING_FEEDBACK",
              "PARTICIPANT_FEEDBACK_SUBMITTED",
              "PARTICIPANT_RECEIVED_FEEDBACK",
              "PARTICIPANT_REMINDER_SENT",
              "QUESTION_ADDED",
              "QUESTION_DELETED",
              "QUESTION_TEXT_UPDATED",
              "QUESTION_LOCKED",
              "QUESTION_UNLOCKED",
              "QUESTION_RENAMED",
              "INTERVIEW_LOCKED",
              "INTERVIEW_UNLOCKED",
              "INTERVIEW_UNSCHEDULED",
              "INTERVIEW_SCHEDULED",
              "INTERVIEW_RESCHEDULED",
              "INTERVIEW_IN_PROGRESS",
              "INTERVIEW_AWAITING_FEEDBACK",
              "INTERVIEW_EVALUATED",
              "INTERVIEW_DEACTIVATED",
              "INTERVIEW_INTERRUPTED",
              "BROWSER_TAB_SWITCH_BLUR",
              "BROWSER_TAB_SWITCH_FOCUS",
              "INTERVIEW_SETTING_COPY_PASTE_TOGGLE"
            ]
          },
          "eventData": {
            "type": "object",
            "additionalProperties": true,
            "description": "EventData will provide details about the events of occurred
during this activity log."
          }
        }
      },
      "InterviewerCompetencyFeedbackResponse": {
        "type": "object",
        "properties": {
          "interviewerName": {
            "type": "string",
            "description": "Interviewer name",
            "example": "Interviewer 1"
          },
          "interviewerStatus": {
            "type": "string",
            "description": "Current interview status of the interviewer",
            "example": "FEEDBACK_SUBMITTED",
            "enum": [
              "INVITED",
              "WAITING",
              "JOINED_INTERVIEW",
              "LEFT_INTERVIEW",
              "FEEDBACK_SUBMITTED",
              "AWAITING_FEEDBACK",
              "NOT_ACTIVE",
              "DENIED",
              "ALLOWED",
              "DISCONNECTED",
              "LEFT_LOBBY"
            ]
          },
          "score": {
            "type": "number",
            "description": "Score provided by interviewer for competency",
            "format": "float",
            "example": 14.4
          },
          "comment": {
            "type": "string",
            "description": "Comment provided by interviewer for competency",
            "example": "A comment for candidate for competency"
          },
          "skippedEvaluation": {
            "type": "boolean",
            "description": "If the evaluation is skipped by interviewer",
            "example": false
          },
          "skippedEvaluationReason": {
            "type": "string",
            "description": "Reason for skipping evaluation",
            "example": "Reason for skipping evaluation"
          },
          "notAssessed": {
            "type": "boolean",
            "description": "If the interviewer not assessed the competency",
            "example": false
          }
        }
      },
      "InterviewerInfo": {
```

Pretty print

```
          "email",
          "firstName"
        ],
        "type": "object",
        "properties": {
          "firstName": {
            "type": "string",
            "description": "First name of the user.\nMax length: 100 characters",
            "example": "John"
          },
          "lastName": {
            "type": "string",
            "description": "Last name of the user.\nMax length: 100 characters",
            "example": "Doe"
          },
          "email": {
            "type": "string",
            "description": "Email of the user.\nMax length: 200 characters",
            "example": "John.doe@example.com"
          }
        },
        "description": "List of interviewers for the interview"
      },
      "InterviewError": {
        "type": "object",
        "properties": {
          "status": {
            "type": "integer",
            "format": "int32"
          },
          "message": {
            "type": "string"
          },
          "i18nKey": {
            "type": "string"
          }
        }
      },
      "InterviewFeedback": {
        "type": "object",
        "properties": {
          "interviewSummary": {
            "$ref": "#/components/schemas/InterviewSummary"
          },
          "candidateFeedbackAndCompetencyDetailResponseList": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/CandidateFeedbackAndResultSummary"
            }
          }
        }
      },
      "InterviewPage": {
        "type": "object",
        "properties": {
          "number": {
            "type": "integer",
            "description": "Data of this page number will be provided in case of
pagination.",
            "format": "int32",
            "example": 1
          },
          "size": {
            "type": "integer",
            "description": "Number of records available in request",
            "format": "int32",
            "example": 20
          },
            "type": "boolean",
            "description": "If there is more data available in next pages",
            "example": true,
            "enum": [true, false]
          },
          "totalPages": {
            "type": "integer",
            "description": "Number of pages that can be fetched for complete data",
            "format": "int32",
            "example": 50
          },
          "totalRecords": {
            "type": "integer",
            "description": "Number of total records available that can be fetched in
paginated way",
            "format": "int64",
            "example": 300
          },
          "interviewList": {
            "type": "array",
            "description": "List of interviews",
            "items": {
              "$ref": "#/components/schemas/InterviewResponse"
            }
          }
        }
      },
      "InterviewParticipantResponse": {
        "type": "object",
        "properties": {
          "uuid": {
            "type": "string",
            "description": "Unique id of the participant",
            "example": "d8ccc0a"
          },
          "status": {
            "type": "string",
            "description": "Current status of the participant",
            "example": "INVITED",
            "enum": [
              "INVITED",
              "WAITING",
              "JOINED_INTERVIEW",
              "LEFT_INTERVIEW",
              "FEEDBACK_SUBMITTED",
              "AWAITING_FEEDBACK",
              "NOT_ACTIVE",
              "DENIED",
              "ALLOWED",
              "DISCONNECTED",
              "LEFT_LOBBY"
            ]
          },
          "user": {
            "$ref": "#/components/schemas/UserResponse"
          },
          "invited": {
            "type": "boolean",
            "description": "If the participant is invited for the interview",
            "example": true
          },
          "joinedOnce": {
            "type": "boolean",
            "description": "If the participant has joined once in the interview",
            "example": true
          },
          "joinedDuration": {
            "type": "integer",
            "format": "int64",
            "example": 17465881
          }
        }
      },
      "InterviewResponse": {
        "required": [
          "duration",
          "timezone"
        ],
        "type": "object",
        "properties": {
          "uuid": {
            "type": "string",
            "description": "A unique identifier of the interview",
            "example": "4de94450"
          },
          "candidateLink": {
            "type": "string",
            "description": "Interview link for the candidate.",
            "example": "https://interviews.mettl.com/candidate/d8ccc0a"
          },
          "interviewerLink": {
            "type": "string",
            "description": "Interview link for the interviewer.",
            "example": "https://interviews.mettl.com/interviewer/d8ccc0a"
          },
          "pin": {
            "type": "string",
            "description": "PIN to access interview for interviewers",
            "example": "HTG7SF"
          },
          "createdOn": {
            "type": "integer",
            "description": "Timestamp when interview was created",
            "format": "int64",
            "example": 17465881
          },
          "linkExpiryDate": {
            "type": "integer",
            "description": "Timestamp when the interview link will expire",
            "format": "int64",
            "example": 17465881
          },
          "settings": {
            "$ref": "#/components/schemas/InterviewSettingsResponse"
          },
          "name": {
            "type": "string",
            "description": "Name of the interview.",
            "example": "Interview with John"
          },
          "mode": {
            "type": "string",
            "description": "Mode of interview",
            "example": "ONLINE",
            "enum": [
              "ONLINE",
              "OFFLINE"
            ]
          },
          "type": {
            "type": "string",
            "description": "Type of interview",
            "example": "INTERVIEW",
            "enum": [
              "INTERVIEW",
              "GROUP_DISCUSSION",
            ]
          },
          "startDate": {
            "type": "integer",
            "description": "Start date(epoch time) of interview",
            "format": "int64",
            "example": 1726403385000
          },
          "duration": {
            "type": "integer",
            "description": "Duration for interview in minutes.",
            "format": "int64",
            "example": 120
          },
          "timezone": {
            "type": "string",
            "description": "Timezone for interview.",
            "example": "Asia/Kolkata"
          },
          "interviewStatus": {
            "type": "string",
            "description": "Current status of the interview",
            "example": "SCHEDULED",
            "enum": [
              "UNSCHEDULED",
              "SCHEDULED",
              "IN_PROGRESS",
              "AWAITING_FEEDBACK",
              "EVALUATED",
              "DEACTIVATED",
              "INTERRUPTED",
              "CANCELLED"
            ]
          },
          "webhookUrl": {
            "type": "string",
            "description": "A webhook url where interview events are published",
            "example": "https://any-web-hook-url.com"
          },
          "linkExpiryDurationInHr": {
            "type": "integer",
            "description": "Duration for which interview link will be active",
            "format": "int32",
            "example": 12
          },
          "location": {
            "type": "string",
            "description": "Location of interview. Only required when mode is 'OFFLINE'",
            "example": "221B Baker Street"
          },
          "jobProfileUuid": {
            "type": "string",
            "description": "Unique id of job profile",
            "example": "7ce6643b-e287-44c3-93ca-5d31ce8dc56d"
          },
          "interviewers": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/InterviewParticipantResponse"
            }
          },
          "candidates": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/CandidateResponse"
            }
          }
        },
      },
      "InterviewSettingRequest": {
        "type": "object",
        "properties": {
          "allowedQuestionTypes": {
            "uniqueItems": true,
            "type": "array",
            "description": "Allowed question types for interview",
            "example": "CODING, WHITEBOARD, DIAGRAM, METTL_QUESTION",
            "items": {
              "type": "string",
              "description": "Allowed question types for interview",
              "example": "CODING, WHITEBOARD, DIAGRAM, METTL_QUESTION",
              "enum": [
                "CODING",
                "WHITEBOARD",
                "DIAGRAM",
                "METTL_QUESTION"
              ]
            },
            "enum": [
              "CODING",
              "WHITEBOARD",
              "DIAGRAM",
              "METTL_QUESTION"
            ]
          },
          "enableCopyPaste": {
            "type": "boolean",
            "description": "Enable copy/paste option during the interview",
            "example": true
          },
          "enableRecording": {
            "type": "boolean",
            "description": "Enable/disable recording of the interview",
            "example": true
          },
          "openInVideoMode": {
            "type": "boolean",
            "description": "Start the interview in video mode",
            "example": true
          },
          "showOverallRecommendation": {
            "type": "boolean",
            "description": "Show the overall recommendations",
            "example": true
          },
          "sendEmails": {
            "type": "boolean",
            "description": "Send emails of interview activity",
            "example": true
          },
          "enableInvite": {
            "type": "boolean",
            "description": "Enable inviting other participants in a interview",
            "example": true
          },
          "language": {
            "type": "string",
            "description": "Language of the interview",
            "example": "English",
            "enum": [
              "English",
              "German",
              "Italian",
              "French",
              "European Portuguese",
              "European Spanish",
              "Greek"
            ]
          },
          "enablePreviousSessionReports": {
            "type": "boolean",
            "description": "Show all previous session reports in a session",
            "example": true
          },
          "enableEvaluationPanel": {
            "type": "boolean",
            "description": "Show evaluation panel in the session",
            "example": true
          },
          "enableCandidateProfile": {
            "type": "boolean",
            "description": "Show candidate profile in the session",
            "example": true
          }
        },
        "description": "Interview settings you want to enable for this interview"
      },
      "InterviewSettingsResponse": {
        "type": "object",
        "properties": {
          "allowedQuestionTypes": {
            "uniqueItems": true,
            "type": "array",
            "description": "Allowed question types for interview",
            "example": [
              "CODING",
              "WHITEBOARD",
              "DIAGRAM",
              "METTL_QUESTION"
            ],
            "items": {
              "type": "string",
              "description": "Allowed question types for interview",
              "example": "[\"CODING\",\"WHITEBOARD\",\"DIAGRAM\",\"METTL_QUESTION\"]",
              "enum": [
                "CODING",
                "WHITEBOARD",
                "DIAGRAM",
                "METTL_QUESTION"
              ]
            },
            "enum": [
              "CODING",
              "WHITEBOARD",
              "DIAGRAM",
              "METTL_QUESTION"
            ]
          },
          "enableCopyPaste": {
            "type": "boolean",
            "description": "Enable copy/paste option during the interview",
            "example": true
          },
          "enableRecording": {
            "type": "boolean",
            "description": "Enable/disable recording of the interview",
            "example": true
          },
          "openInVideoMode": {
            "type": "boolean",
            "description": "Start the interview in video mode",
            "example": true
          },
          "showOverallRecommendation": {
```

Pretty print

```
            "description": "Show the overall recommendations",
            "example": true
          },
          "recordingDisabledFromClient": {
            "type": "boolean",
            "description": "This flag shows if recording is disabled from client level",
            "example": true
          },
          "enableCopyPasteSettingModified": {
            "type": "boolean",
            "description": "This flag shows if the copy paste setting is modified at least once in the current interview",
            "example": true
          },
          "sendEmails": {
            "type": "boolean",
            "description": "Send emails of interview activity",
            "example": true
          },
          "enableInvite": {
            "type": "boolean",
            "description": "Enable candidate profile in the interview",
            "example": true
          },
          "language": {
            "type": "string",
            "description": "Language of the interview",
            "example": "English",
            "enum": [
              "English",
              "German",
              "Italian",
              "French",
              "European Portuguese",
              "European Spanish",
              "Polish",
              "Greek"
            ]
          }
        },
        "description": "Details of applicable interview settings."
      },
      "InterviewSummary": {
        "type": "object",
        "properties": {
          "duration": {
            "type": "integer",
            "description": "Duration of interview in millis",
            "format": "int64",
            "example": 868396
          },
          "startTime": {
            "type": "integer",
            "description": "Start time of interview in millis",
            "format": "int64",
            "example": 1715158648725
          },
          "endTime": {
            "type": "integer",
            "description": "End time of interview in millis",
            "format": "int64",
            "example": 1715158656027
          }
        },
        "description": "This field will provide interview summary"
      },
      "InterviewSummaryResponse": {
        "type": "object",
          "duration": {
            "type": "integer",
            "description": "Duration of interview in millis",
            "format": "int64",
            "example": 868396
          },
          "startTime": {
            "type": "integer",
            "description": "Start time of interview in millis",
            "format": "int64",
            "example": 1715158648725
          },
          "endTime": {
            "type": "integer",
            "description": "End time of interview in millis",
            "format": "int64",
            "example": 1715158656027
          },
          "interviewSettings": {
            "$ref": "#/components/schemas/InterviewSettingsResponse"
          },
          "name": {
            "type": "string",
            "description": "Name of interview",
            "example": "Interview with john"
          },
          "type": {
            "type": "string",
            "description": "Type of interview",
            "example": "INTERVIEW",
            "enum": [
              "INTERVIEW",
              "GROUP_DISCUSSION",
              "ROLE_PLAY"
            ]
          },
          "mode": {
            "type": "string",
            "description": "Mode of interview",
            "example": "ONLINE",
            "enum": [
              "ONLINE",
              "OFFLINE"
            ]
          },
          "candidates": {
            "type": "array",
            "description": "List of candidates in the interview",
            "items": {
              "$ref": "#/components/schemas/CandidateBasicDetails"
            }
          },
          "jobProfile": {
            "$ref": "#/components/schemas/JobProfileBasicDetailsResponse"
          },
          "location": {
            "type": "string",
            "description": "Location of interview. Only required when mode is 'OFFLINE'",
            "example": "221B Baker Street"
          },
          "meetingRecordingAvailable": {
            "type": "boolean",
            "description": "Is meeting recording available",
            "example": true,
            "enum": [true, false]
          }
        }
      },
        "type": "object",
        "properties": {
          "name": {
            "type": "string",
            "description": "Name of job profile",
            "example": "Java Developer"
          },
          "jobDescription": {
            "$ref": "#/components/schemas/FileDetailsResponse"
          }
        },
        "description": "Details of job profile associated with this interview"
      },
      "JobProfileData": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string",
            "description": "Name of job profile",
            "example": "Software Test Engineer"
          },
          "uuid": {
            "type": "string",
            "description": "Unique id of a job profile",
            "example": "7ce6643b-e287-44c3-93ca-5d31ce8dc56d"
          },
          "type": {
            "type": "string",
            "description": "Type of job profile",
            "example": "ADMIN",
            "enum": [
              "ADMIN",
              "SELF",
              "ADMIN",
              "SELF"
            ]
          },
          "createdOn": {
            "type": "integer",
            "description": "Timestamp (in millis) when this job profile was created",
            "format": "int64",
            "example": 12312378909
          },
          "modifiedOn": {
            "type": "integer",
            "description": "Timestamp (in millis) when this job profile was last modified",
            "format": "int64",
            "example": 12312378909
          },
          "isEditable": {
            "type": "boolean",
            "description": "Is this job profile editable",
            "example": true,
            "enum": [true, false]
          },
          "isFeedbackSubmittedForCandidate": {
            "type": "boolean",
            "description": "Is any feedback is submitted for the candidates which are
present in the session linked to this job profile",
            "example": true,
            "enum": [true, false]
          },
          "typeVsCountMap": {
            "type": "object",
            "additionalProperties": {
              "type": "integer",
              "format": "int64"
            }
        },
        "description": "List of job profiles"
      },
      "JobProfileDto": {
        "type": "object",
        "properties": {
          "uuid": {
            "type": "string",
            "description": "Unique id of a job profile",
            "example": "7ce6643b-e287-44c3-93ca-5d31ce8dc56d"
          },
          "name": {
            "type": "string",
            "description": "Name of job profile",
            "example": "Software Test Engineer"
          },
          "descriptionFileUrl": {
            "type": "string",
            "description": "Url for the description file uploaded for job profile",
            "example": "https://file-url.com"
          },
          "descriptionFileName": {
            "type": "string",
            "description": "Name of the uploaded description file for job profile",
            "example": "Job-Profile-Description.pdf"
          },
          "descriptionFileUuid": {
            "type": "string",
            "description": "Uuid belongs to the description file of job profile",
            "example": "7ce6643b-e287-44c3-93ca-5d31ce8dc56d"
          },
          "competencies": {
            "type": "array",
            "description": "List of competencies in this job profile",
            "items": {
              "$ref": "#/components/schemas/CompetencyDto"
            }
          },
          "type": {
            "type": "string",
            "description": "Type of job profile",
            "example": "ADMIN",
            "enum": [
              "ADMIN",
              "SELF",
              "ADMIN",
              "SELF"
            ]
          },
          "maxMarks": {
            "type": "number",
            "description": "Maximum marks for job profile",
            "format": "float",
            "example": 100
          },
          "guide": {
            "type": "string",
            "description": "Interviewer guide for job profile",
            "example": "guide"
          },
          "allowNotAssessed": {
            "type": "boolean",
            "description": "If interviewer is allowed to not assess a candidate",
            "example": true
          },
          "mettlAssessments": {
            "type": "string",
            "description": "Mettl assessment name linked in this job profile"
        }
      },
      "JobProfilePage": {
        "type": "object",
        "properties": {
          "number": {
            "type": "integer",
            "description": "Data of this page number will be provided in case of pagination.",
            "format": "int32",
            "example": 1
          },
          "size": {
            "type": "integer",
            "description": "Number of records available in request",
            "format": "int32",
            "example": 20
          },
          "hasMorePage": {
            "type": "boolean",
            "description": "If there is more data available in next pages",
            "example": true,
            "enum": [true, false]
          },
          "totalPages": {
            "type": "integer",
            "description": "Number of pages that can be fetched for complete data",
            "format": "int32",
            "example": 50
          },
          "totalRecords": {
            "type": "integer",
            "description": "Number of total records available that can be fetched in paginated way",
            "format": "int64",
            "example": 300
          },
          "jobProfileDataList": {
            "type": "array",
            "description": "List of job profiles",
            "items": {
              "$ref": "#/components/schemas/JobProfileData"
            }
          }
        }
      },
      "MeetingSummaryResponse": {
        "type": "object",
        "properties": {
          "playUrl": {
            "type": "string",
            "description": "Play url of the recorded meeting",
            "example": "https://video_play_url.com"
          },
          "downloadUrl": {
            "type": "string",
            "description": "Download url of the recorded meeting",
            "example": "https://video_download_url.com"
          },
          "startTime": {
            "type": "string",
            "description": "Start time of the recorded meeting",
            "example": "1715158648725"
          },
          "endTime": {
            "type": "string",
            "description": "End time of the recorded meeting",
            "example": "1715158648725"
          "size": {
            "type": "integer",
            "description": "Size of the recorded meeting",
            "format": "int64",
            "example": 1715158648725
          },
          "status": {
            "type": "string",
            "description": "status of the recorded meeting",
            "example": "AVAILABLE",
            "enum": [
              "STARTED",
              "STOPPED",
              "AVAILABLE"
            ]
          }
        }
      },
      "RescheduleInterviewApiRequest": {
        "required": [
          "duration",
          "interviewUuid",
          "sendEmail",
          "startDate",
          "timezone"
        ],
        "type": "object",
        "properties": {
          "interviewUuid": {
            "type": "string",
            "description": "Unique identifier for the interview",
            "example": "7ce6643b-e287-44c3-93ca-5d31ce8dc56d"
          },
          "duration": {
            "type": "integer",
            "description": "Duration for interview in minutes. Max duration: 180",
            "format": "int64",
            "example": 45
          },
          "startDate": {
            "type": "integer",
            "description": "Start date in epoch time",
            "format": "int64",
            "example": 1726403385000
          },
          "timezone": {
            "type": "string",
            "description": "Valid timezone for interview. Use /api/interview/timezones
(getAllTimezones) to get a list of valid timezones",
            "example": "Asia/Kolkata"
          },
          "sendEmail": {
            "type": "boolean",
            "description": "Will email be send once rescheduled successfully?",
            "example": true
          },
          "location": {
            "type": "string",
            "description": "Location of interview. Only required when mode is 'OFFLINE'",
            "example": "221B Baker Street"
          }
        }
      },
      "ScheduleInterviewApiRequest": {
        "required": [
          "candidates",
          "duration",
          "interviewers",
          "timezone"
        ],
        "type": "object",
        "properties": {
          "interviewers": {
            "type": "array",
            "description": "List of interviewers for the interview",
            "items": {
              "$ref": "#/components/schemas/InterviewerInfo"
            }
          },
          "candidates": {
            "type": "array",
            "description": "List of candidates for the interview",
            "items": {
              "$ref": "#/components/schemas/CandidateInfo"
            }
          },
          "startDate": {
            "type": "integer",
            "description": "Start date in epoch time",
            "format": "int64",
            "example": 1726403385000
          },
          "timezone": {
            "type": "string",
            "description": "Valid timezone for interview. Use /api/interview/timezones (getAllTimezones) to get a list of valid timezones",
            "example": "Asia/Kolkata"
          },
          "duration": {
            "type": "integer",
            "description": "Duration for interview in minutes. Max duration: 180",
            "format": "int64",
            "example": 120
          },
          "jobProfileUuid": {
            "type": "string",
            "description": "Unique id of the job profile. Use /api/jobProfile/all to get a list of job profiles for client",
            "example": "7ce6643b-e287-44c3-93ca-5d31ce8dc56d"
          },
          "webhookUrl": {
            "type": "string",
            "description": "A webhook url where interview events are published",
            "example": "https://any-web-hook-url.com"
          },
          "settings": {
            "$ref": "#/components/schemas/InterviewSettingRequest"
          },
          "name": {
            "type": "string",
            "description": "Name of the interview. Only applicable when your interview type is 'GROUP_DISCUSSION'",
            "example": "Interview with john"
          },
          "type": {
            "type": "string",
            "description": "Type of interview",
            "example": "INTERVIEW",
            "enum": [
              "INTERVIEW",
              "GROUP_DISCUSSION",
              "ROLE_PLAY"
            ]
          },
          "mode": {
            "type": "string",
            "example": "ONLINE",
            "enum": [
              "OFFLINE",
              "ONLINE"
            ]
          },
          "location": {
            "type": "string",
            "description": "Location of interview. Only required when mode is 'OFFLINE'",
            "example": "221B Baker Street"
          }
        },
        "description": "This model is used to schedule interview."
      },
      "ScheduleInterviewApiResponse": {
        "type": "object",
        "properties": {
          "uuid": {
            "type": "string",
            "description": "A unique identifier of the newly created interview",
            "example": "4de94450"
          },
          "pin": {
            "type": "string",
            "description": "PIN to access interview for interviewers",
            "example": "HTG7SF"
          },
          "participants": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/UserWithLinkResponse"
            }
          }
        }
      },
      "UserResponse": {
        "type": "object",
        "properties": {
          "uuid": {
            "type": "string",
            "description": "A unique identifier for the newly created user. This uuid must
be used to perform and operation for a user.",
            "example": "cc338777-9e3f-4c98-b8cc-ef8f87de1b0d"
          },
          "firstName": {
            "type": "string",
            "description": "First name of the user",
            "example": "John"
          },
          "lastName": {
            "type": "string",
            "description": "Last name of the user",
            "example": "Doe"
          },
          "status": {
            "type": "string",
            "description": "Status of the user",
            "example": "ENABLED",
            "enum": [
              "ENABLED",
              "DISABLED"
            ]
          },
          "email": {
            "type": "string",
            "description": "Email of the user",
            "example": "john.doe@example.com"
          },
            "type": "string",
            "description": "Type of user.  CANDIDATE - If this object is a Candidate.
INTERVIEWER - If this object is an Interviewer",
            "example": "CANDIDATE",
            "enum": [
              "CANDIDATE",
              "INTERVIEWER"
            ]
          }
        },
        "description": "Details of the user"
      },
      "UserWithLinkResponse": {
        "type": "object",
        "properties": {
          "uuid": {
            "type": "string",
            "description": "A unique identifier for the newly created user. This uuid must
be used to perform and operation for a user.",
            "example": "cc338777-9e3f-4c98-b8cc-ef8f87de1b0d"
          },
          "firstName": {
            "type": "string",
            "description": "First name of the user",
            "example": "John"
          },
          "lastName": {
            "type": "string",
            "description": "Last name of the user",
            "example": "Doe"
          },
          "status": {
            "type": "string",
            "description": "Status of the user",
            "example": "ENABLED",
            "enum": [
              "ENABLED",
              "DISABLED"
            ]
          },
          "email": {
            "type": "string",
            "description": "Email of the user",
            "example": "john.doe@example.com"
          },
          "type": {
            "type": "string",
            "description": "Type of user.  CANDIDATE - If this object is a Candidate.
INTERVIEWER - If this object is an Interviewer",
            "example": "CANDIDATE",
            "enum": [
              "CANDIDATE",
              "INTERVIEWER"
            ]
          },
          "link": {
            "type": "string",
            "description": "Link for interview.",
            "example": "https://interviews.mettl.com/candidate/d8ccc0a"
          }
        }
      }
    }
  }
}
```