

# Content

## GET 강의ID로 질문, 답변 가져오기

http://comman.cf/content/lecturequestion/{lectureID}

Req : { lectureID }

Res : Questions and answers about the lecture

Des : 해당 lecture\_id의 질문, 답변가져오기

### HEADERS

<b>Content-Type</b>	application/x-www-form-urlencoded
<b>authorization</b>	tokenValue

## POST 강의ID로 질문 DB에 삽입

http://comman.cf/content/lecturequestion/insertanswer

Req : { lectureID, question\_text }

Res : 삽입된 정보

Des : lectureID와 question\_text를 받아  
DB.lecture\_question에 삽입 후  
삽입 된 정보 반환

### HEADERS

<b>Content-Type</b>	application/x-www-form-urlencoded
<b>authorization</b>	tokenValue

### BODY

<b>lectureID</b>	{ lectureID }
<b>question_text</b>	{ 질문 내용 }

# Content

## GET 강좌 ID로 맞보기 강의ID 가져오기

```
http://comman.cf/content/coursepage/previewvideo/{couseID}
```

```
Req : { courseID }
```

```
Res : { video_id }
```

```
Des : 강좌ID로 해당하는 강좌의 포함되는 강의 첫번째 동영상ID 불러오기
```

강좌에 동영상이 하나도 없는 경우는 없다고 가정함.

### HEADERS

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

## GET [챕터내 강의목록 (유저별 접근 여부 포함)]

```
http://comman.cf/content/lecturepage/lectureList?chapterID={chapterID}
```

```
Req : { chapterID }
```

```
Res : lecture List Arr
```

```
Des : 해당 chapter에 속해 있는 lecture List Arr
```

### HEADERS

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

### PARAMS

**chapterID**          {chapterID}

# Content

## GET [강의ID로 강의정보 가져오기]

```
http://comman.cf/content/lectures?lectureID={lectureID}
```

Req : { lectureID }

Res : 해당 lecture\_id의 lecture 정보

Des : video\_id => 영상강의의 youtube videoID가 저장되는 값,  
pass\_value => 퀴즈강의의 퀴즈 통과 기준 개수

### HEADERS

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

### PARAMS

**lectureID**          {lectureID}

## GET [전체 카테고리별 강좌정보]

```
http://comman.cf/content/categories/course
```

Req : -

Res : 카테고리별 강좌 정보

Des : 카테고리별 강좌 정보

### HEADERS

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

# Content

## GET [강의ID로 사진강의정보 가져오기]

```
http://comman.cf/content/lecturepicture/{lectureID}
```

```
Req : { lectureID }
```

```
Res : Info for Picture Lecture
```

```
Des : 이미지강의 정보
```

### HEADERS

---

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

## GET [강의ID로 비디오강의정보 가져오기]

```
http://comman.cf/content/lecturevideo/{lectureID}
```

```
Req : { lectureID }
```

```
Res : Info for Video Lecture
```

```
Des : Video Lecture 정보
```

### HEADERS

---

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

# Content

## GET [챕터ID로 챕터정보 가져오기]

```
http://comman.cf/content/chapters?chapterID={chapterID}
```

```
Req : { chapterID }
```

```
Res : 해당 chapter_id의 chapter 정보
```

```
Des : 해당 chapter_id의 chapter 정보
```

### HEADERS

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

### PARAMS

**chapterID**          {chapterID}

## GET [강좌별 챕터정보]

```
http://comman.cf/content/courses/{courseID}/chapters
```

```
Req : { courseID }
```

```
Res : 강좌별 챕터정보
```

```
Des : 강좌별 챕터정보
```

### HEADERS

**Content-Type**      application/x-www-form-urlencoded

## GET [강의에 대한 퀴즈정보]

```
http://comman.cf/content/lecturequiz/{lectureID}
```

```
Req : { lectureID }
```

```
Res : Quiz List Belong to Lecture
```

```
Des : 강의 ID값으로 해당 강의가 갖고있는 퀴즈 목록들 반환
```

### HEADERS

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

# Content

## GET [강좌ID로 강좌정보 가져오기]

```
http://comman.cf/content/courses?courseID={courseID}
```

Req : {courseID}

Res : 해당 courseID의 course 정보

Des : 해당 courseID의 course 정보

### HEADERS

**Content-Type** application/x-www-form-urlencoded

**authorization** tokenvalue

### PARAMS

**courseID** {courseID}

## GET [현재 강의ID로 다음에 들어야할 강의ID 가져오기]

```
http://comman.cf/content/lecturepage/nextLecture?lectureID={lectureID}
```

Req : { lectureID }

Res : 다음에 들어야할 강의 id,

-1 : 해당 강좌를 다 들었을시 반환

Dec : {lectureID} 값을 통해 다음에 들어야할 강의 id를 가져옴

### HEADERS

**Content-Type** application/x-www-form-urlencoded

**authorization** tokenValue

### PARAMS

**lectureID** {lectureID}

# Search

## GET [카테고리 단위 검색결과]

http://comman.cf/search/courses/categories/{categoryID}

Req : {categoryID}

Res : category와 연관된 강좌들의 정보

Dec : {categoryID}에 관련된 강좌 정보 출력

### HEADERS

---

**Content-Type**      application/x-www-form-urlencoded

**authorization**      tokenValue

## POST [자음 단위 검색]

http://comman.cf/search/courses

Req : {search}

Res : search와 연관된 강좌들의 정보

Des : req로 받은 search 값에 대하여

관련 문자열 검색(자음단위), 조회수 기준으로 정렬

### HEADERS

---

**Content-Type**      application/x-www-form-urlencoded

**authorization**      tokenValue

### BODY

---

**search**              {검색할 단어}

# User

## POST [유저의 강의 수강 기록]

http://comman.cf/users/lectureHistory

Req : {lectureID}

Res : message

Dec : 강의 수강을 누를시 해당 유저의 수강기록이 생성됨(수강 중 상태로)

### HEADERS

---

**Content-Type**      application/x-www-form-urlencoded

**authorization**      tokenValue

### BODY

---

**lectureID**            { lectureID }

## GET [유저별 강좌 구매 여부]

http://comman.cf/users/purchase/{courseID}

Req : { courseID }

Res :

- result : 0 ==> 구매안함
- result : 1 ==> 구매함

Des : tokenValue의 유저가 {courseID}강좌 구매 여부 확인

### HEADERS

---

**Content-Type**      application/x-www-form-urlencoded

**authorization**      tokenValue



# User

## POST [유저별 강좌 등록]

http://comman.cf/users/register

Req : { courseID }

Res : Message

Des : 유저 (토큰 값으로 전달된)가 특정 강좌를 등록,  
등록테이블에 유저 정보 생성됨

### HEADERS

**Content-Type** application/x-www-form-urlencoded

**authorization** tokenValue

### BODY

**courseID** { courseID }

## GET [수강중인 강좌 진도율]

http://comman.cf/users/main/progressCourse

Req : -

Res : Progress rate for each course

Dec : Progress rate

분자( = A ) : User가 해당 강좌에서 들었거나 듣고있는 강의 수

분모( = B ) : 해당 강좌의 모든 강의 수

$==> A / B * 100$

### HEADERS

**Content-Type** application/x-www-form-urlencoded

**authorization** tokenValue

# User

## GET [인삿말]

http://comman.cf/users/main/greeting

Req : -

Res : "ment", "userImg"

Des : 유저가 접속시 띄워줄 랜덤 멘트, 유저 썸네일

### HEADERS

---

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

## POST [유저 등록]

http://comman.cf/users/insert\_user\_info

Req : accessToken

Res : message, token, user\_info

Des : accessToken값을 통해 token 값과 user\_info를 반환

### HEADERS

---

**Content-Type**      application/x-www-form-urlencoded

### BODY

---

**accessToken**        { accessToken Value }

# User

## GET [시청하던 강의]

http://comman.cf/users/lectureRecentWatch/{lectureID}

Req : {lectureID}

Res : course\_ID, course\_title, chapter\_priority,  
lecture\_title, lecture\_priority, lecture\_type

Des : 최근 시청강의에 전달되는 값, lecture\_type값으로 분기하여  
퀴즈, 동영상, 글그림 결정 후 알맞은 아이콘 사용.

### HEADERS

**Content-Type** application/x-www-form-urlencoded

**authorization** tokenValue

## PUT [유저의 수강 종료 갱신]

http://comman.cf/users/lectureHistory/{lectureID}

Req : {lectureID}

Res : message

Dec : 강의 종료에 대응되는 이벤트 발생시  
해당 유저의 수강기록이 변경됨 (수강 종료 상태로)

### HEADERS

**Content-Type** application/x-www-form-urlencoded

**authorization** tokenValue

# User

## GET [유저별 강좌 등록 여부]

http://comman.cf/users/register/{courseID}

Req : {cousrID}

Res :

- result : 1 ==> 등록 됨

- result : 0 ==> 등록 안됨

### HEADERS

---

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

## PUT [유저별 강좌 구매]

http://comman.cf/users/purchase/{courseID}

Req : { courseID }

Res : Message

Des : 토큰 값으로 넘어온 유저의 해당 강좌 구매

### HEADERS

---

**Content-Type**      application/x-www-form-urlencoded

**authorization**    tokenValue

# Admin

## POST 강사 등록

http://localhost:8080/instructor

Req : { name }, { instructorProfile }

Res : -

Des : 강사 이름과 프로필 사진 등록

### BODY

**name** { name }

**instructorProfile**

## POST 단원 업데이트

http://localhost:8080/course/chapter/lecture/video

Req : { info for Video Lecture }

Res : -

Des : 단원 업데이트 필요한 데이터

### HEADERS

**Content-Type** application/x-www-form-urlencoded

### BODY

**title** { title }

**info** { info }

**priority** { priority }

**courseld** { courseld }

# Admin

## POST 비디오 강의 업데이트

http://localhost:8080/course/chapter/lecture/video

Req : { info for Video Lecture }

Res : -

Des : 비디오 강의 업데이트

### HEADERS

**Content-Type**      application/x-www-form-urlencoded

### BODY

**title**                { title }

**info**                { info }

**priority**            { priority }

**chapterId**          { chapterId }

**type**                { type }

**videoid**            { videoid }

**playTime**          { playTime }

## POST 강의에 퀴즈 업로드

http://localhost:8080/course/chapter/lecture/quiz

Req : 퀴즈 정보

Res : -

Des : 강의에 대한 퀴즈 업로드

### HEADERS

**Content-Type**      application/x-www-form-urlencoded

### BODY

**title**                { title }

**priority**            { priority }

**lectureId**          { lectureId }

**explanation**        { explanation }

**choices**            { choices }

**answerIndex**      { answerIndex }

**questionPicture**   { questionPicture }

**e**

# Admin

## GET 단원 리스트 호출

http://localhost:8080/course/chapter?id={ course ID }

Req : { course ID }

Res : 단원 리스트 호출

Des : 단원 리스트 호출

### PARAMS

**id** { course ID }

## GET 관리자 페이지 접근

http://localhost:8080/instructors

Req : -

Res : 강사 리스트 출력

Des : 강사 리스트 출력

## GET 강좌ID로 단원리스트를 반환

http://localhost:8080/course/chapter?id={courseID}

Req : { courseID }

Res : chapterList

Des : 강좌 ID로 해당하는 단원리스트 출력

### PARAMS

**id** {courseID}

## GET 강좌 등록하기 접근

http://localhost:8080/course?id={ supplier ID }

Req : { supplier ID }

Res : 강좌 출력

Des : 공급자가 갖고 있는 강좌들 출력

### PARAMS

**id** { supplier ID }

# Admin

## GET 단원ID로 퀴즈 리스트 반환

```
http://localhost:8080/course/chapter/lecture/quiz?chapterId={chapterID}
```

Req : { chapterID }

Res : quizLectureList

Des : 단원 ID로 해당하는 퀴즈 강좌 리스트 호출

### PARAMS

**chapterId** {chapterID}

## GET 퀴즈 강의 ID로 퀴즈 리스트 반환

```
http://localhost:8080/course/chapter/lecture/quiz/question?lectureId={lectureID}
```

Req : { lectureId }

Res : questionList

Des : 퀴즈 강좌 ID로 해당하는 문제 리스트 출력

### PARAMS

**lectureId** {lectureID}

## GET 단원 ID로 강좌 리스트 호출

```
http://localhost:8080/course/chapter/lecture?id={chapterID}
```

Req : { chapterID }

Res : lectureList

Des : 단원 ID로 해당하는 강좌리스트 호출

### HEADERS

**Content-Type** application/x-www-form-urlencoded

### PARAMS

**id** {chapterID}



# Admin

## POST 사진 강의 업데이트

http://localhost:8080/course/chapter/lecture/picture

Req : { 사진 }

Res : -

Des : 사진 강의 업데이트

BODY

---

**quizPicture**