

QuizController. getQuizReq(res)

deconstruct concept_id.

```
class quizController (req, res) {
  private moduleService: ModuleService
  constructor (ModuleService: ModuleService) {
    this. ModuleService = ModuleService
  }

  async getQuiz (req: Req, res: Res) {
    const moduleId = req.parse.ModuleId || req.query.moduleId
    if (moduleId === NaN) {
      res.status(400).json({ error: 'Module ID must be a number' })
      return
    }

    const quiz = moduleService. getQuiz(moduleId)
    res.json(quiz)
  }
}
```

1. why we only need a moduleService? It's not a absolute answer?

class ModuleService {

private concept: ConceptRep.

constructor (conceptRep: ConceptRep) {

this.concept =

promise<Quiz>

async getQuiz (moduleId: number): ~~Quiz~~ {

1. const concept = await this.conceptRepo.findConceptWithQuiz(moduleId)

2. Validation

if (!concept || !concept.quiz) {

throw new Error(`Quiz not found for concept ID: \${conceptId}`)

3. return

const quiz = concept.quiz

- - -
- - -
- - -
- - -

return quiz