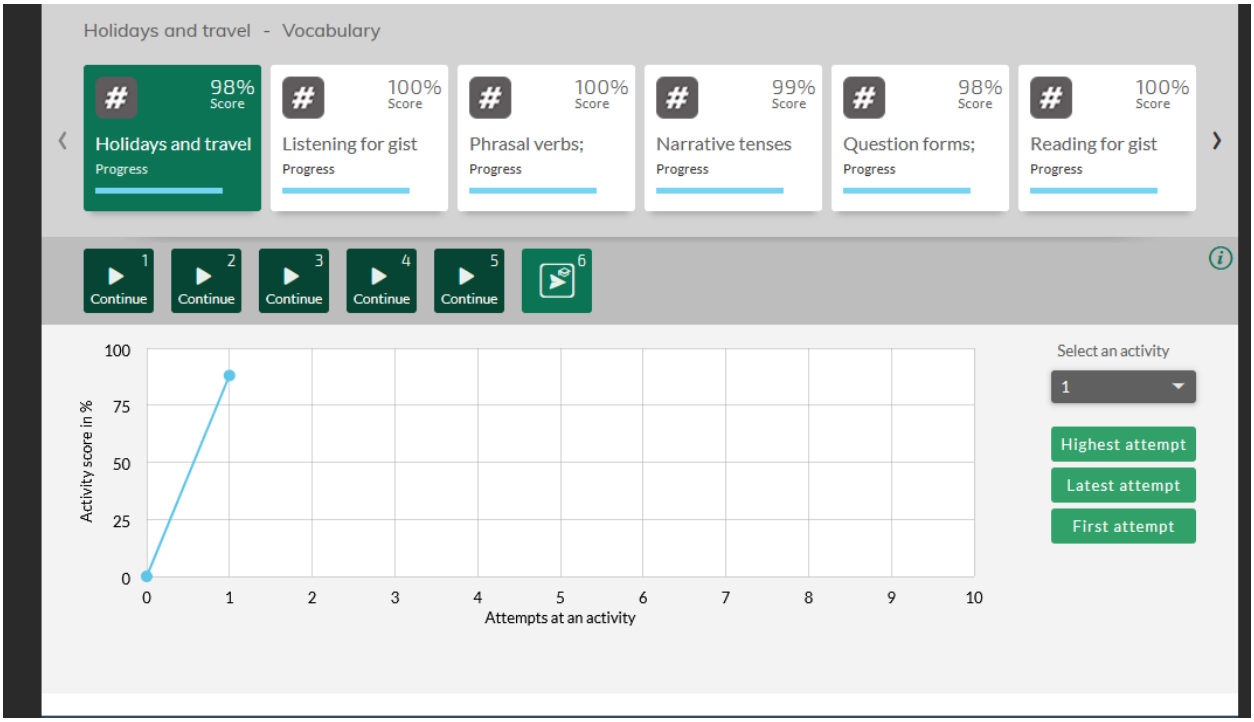
# Exploiting MacMillan e-learning system.

## How to

So first you need to select subject you want to complete.



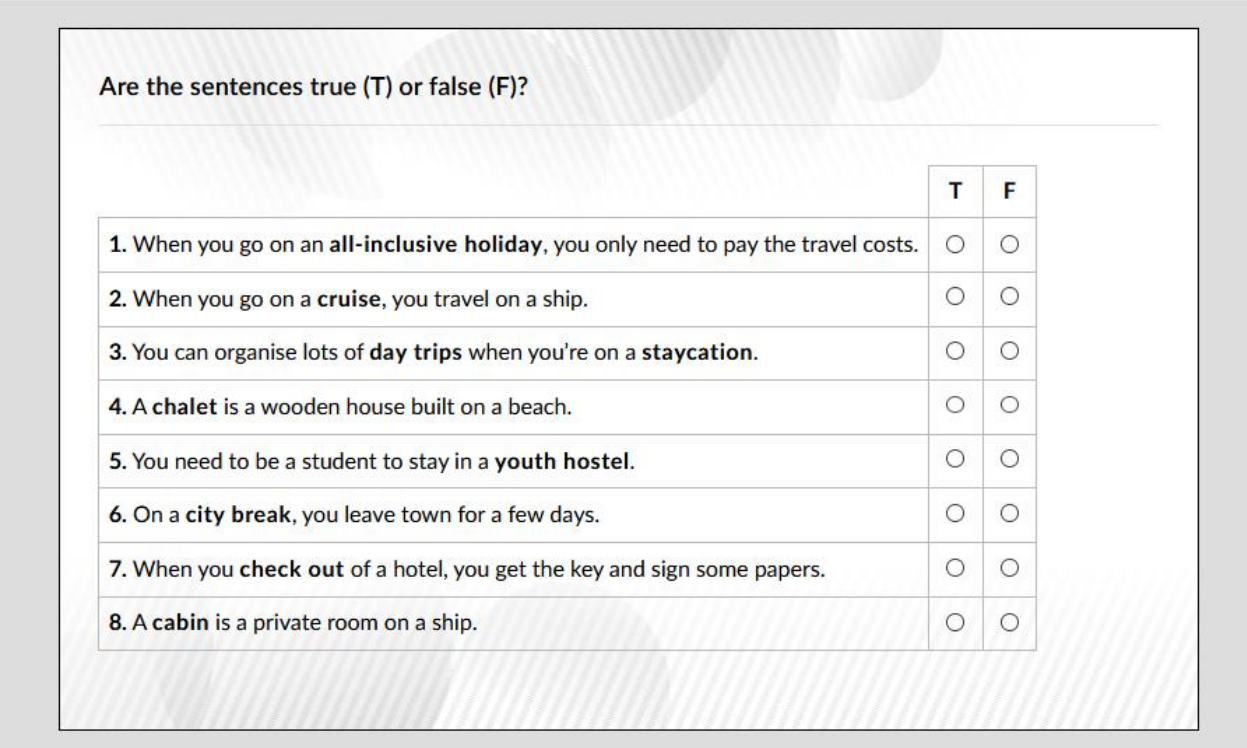

Now after clicking on subject, your URL will look like this

https://prod.lms.macmillaneducation.com/unit?productId=139&unitId=73667&activitySetId=73694&activityId=efd873f930ee4958b8c63be71d3d85c2

Now we need to remember value of parameter **activitySetId.** It contains 6 digit value. It is "key" for the current subject. It is used to get answers for exercises in current subject.

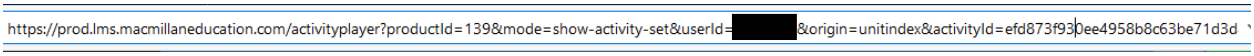
After getting "The Key", we need to select exercise we want to complete.

In this example I selected first exercise from "Holidays and travel" subject.




Now if we take a look at the URL, we will see another important parameter- **activityId.**

It is ID of current exercise.




# Exploiting

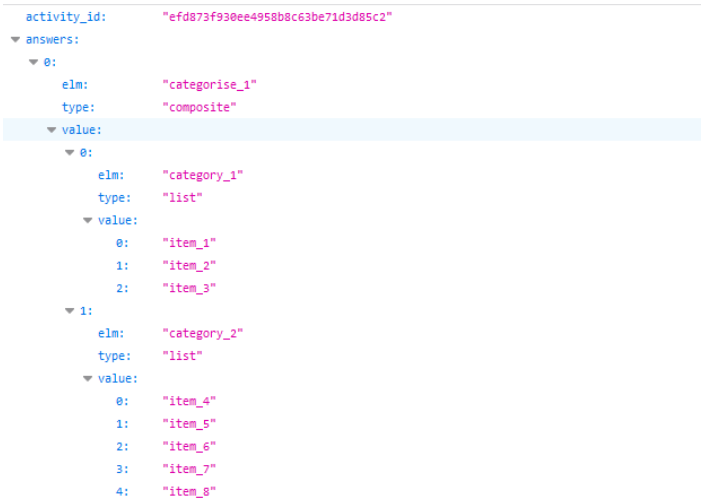
Now MacMillan requests this URL when you click submit answer.

https://prod.lms.macmillaneducation.com/lms/v1/activityset/XXXXX/activities/YYYYY/answer

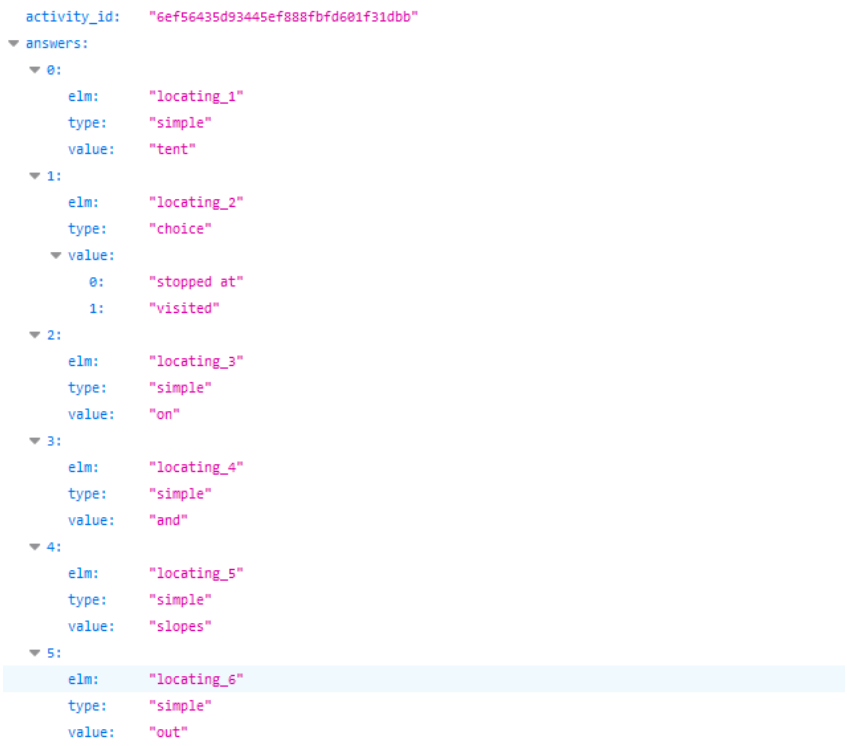
Website is not verifying if student completed the exercise before checking answers, so you can get them by knowing just **activitySetId** and **activityId.**

After changing XXXXXX to **activitySetId** and YYYYYY to **activityId** you will get answers in JSON format.

Response from server will look like this




Or it could look like this



It could also look different but the principle is the same.

# How to understand JSON?

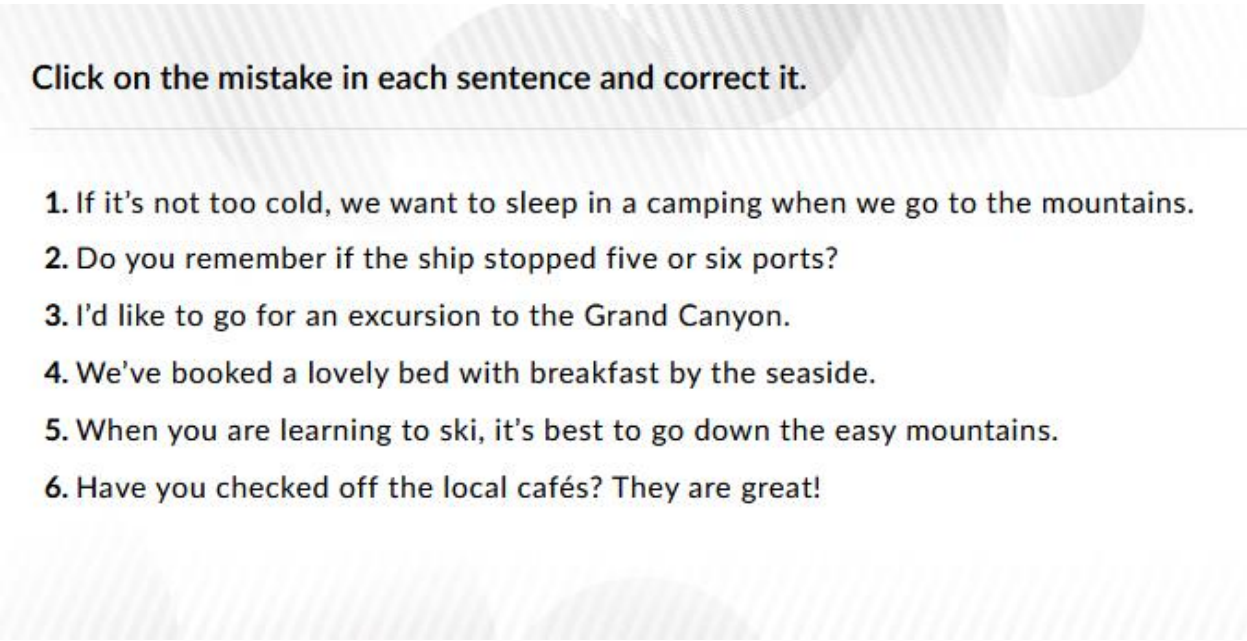
**Example 1**

The most important part is **value** data


```
activity_id:    "6ef56435d93445ef888fbfd601f31dbb"
answers:
  0:
      elm:      "locating_1"
      type:     "simple"
      value:    "tent"
  1:
      elm:      "locating_2"
      type:     "choice"
    value:
        0:      "stopped at"
        1:      "visited"
  2:
      elm:      "locating_3"
      type:     "simple"
      value:    "on"
  3:
      elm:      "locating_4"
      type:     "simple"
      value:    "and"
  4:
      elm:      "locating_5"
      type:     "simple"
      value:    "slopes"
  5:
      elm:      "locating_6"
      type:     "simple"
      value:    "out"
```


You see numbers from 0 to 5. It corresponds to gaps/things to complete in exercise. We count from 0 so there is 6 "questions" to complete. The value holds the expected input from user.

Below is the photo of exercise that I selected.




Now we look for answer for question one. **We count from *0* so answer for exercise 1 is under section 0, answer for second question is under section 1 …….**

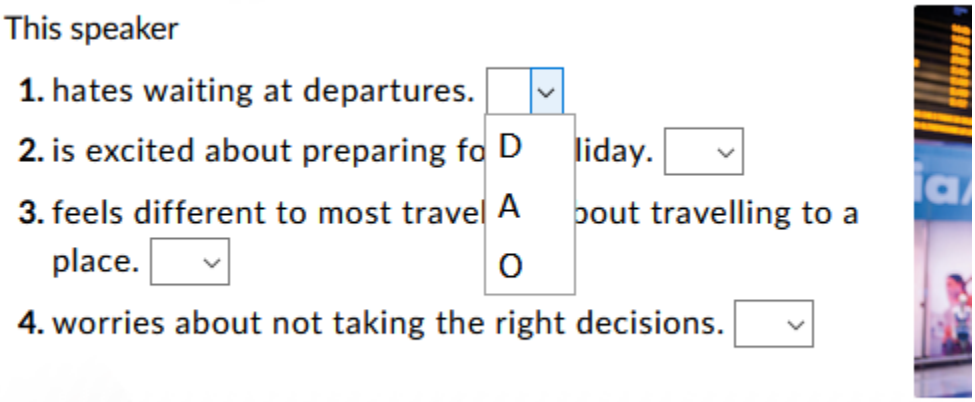
**Example 2**

In another example JSON could look like this:


```
activity_id:   "5c32f4a34caa4f44a4e61d07957b4044"
▼ answers:
  ▼ 0:
      elm:     "dropDown_1"
      type:    "simple"
      value:   "item_3"
  ▼ 1:
      elm:     "dropDown_2"
      type:    "simple"
      value:   "item_3"
  ▼ 2:
      elm:     "dropDown_3"
      type:    "simple"
      value:   "item_1"
  ▼ 3:
      elm:     "dropDown_4"
      type:    "simple"
      value:   "item_2"
```

In this situation we see that value is "**item_2**" or similar.

Our exercise looks like this:




In this situation we need to select **second item** from the list.

**Side-Note:**

There are also other variations of JSON response, but it is easy to understand.