
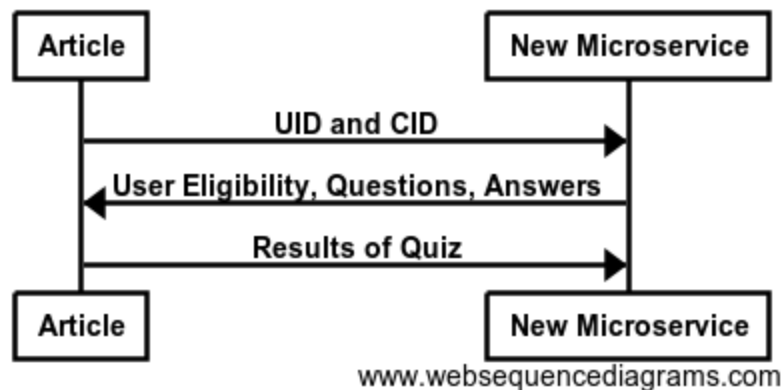


## User Story

Name	Quiz - API Microservice	Design Lead	N/A
Platform Impacts		Development Lead	Johnny Developer
Platform		Current Status	
Total Points	80 (40 points over 2 sprints)		



## Narrative

Users want the ability to “test drive” the quality of content with the LA Times before becoming a full paid subscriber. One of the methods being tested as a method to earn the ability to see an article is answering 2 basic questions about the article for a reader’s comprehension of the content.

Because this functionality is in the early stages we want to validate that it creates the proper user response in a small scale test before making large modifications to the CMS system to support it. To that end, we have envisioned a small microservice that receives a user ID and content ID from the article paywall and then returns a user’s eligibility for the quiz offer as well as the questions, answers, metadata needed to drive the UX experience. Previous User Stories (Quiz-UX Design) will define the mockups needed for the experience.

Once the UX flow is completed, the service will receive an update from the article page with how the user did with the quiz. If those results determine a user abandoned a quiz without completing it then that user’s ID record should be updated to remove their eligibility for future

questions. If they answer correctly, they should receive future quiz offers but not for the same content they have already answered for. If they answer incorrectly then the user is presented with other articles to read on the same topic (associated content). Envisioned request/response JSON is included below but can be modified as needed.

This story is envisioned to span 2 sprints for design, development, and deployment.

```
{
  "contentid": "12345667",
  "userid": "5678"
}
```

-Envisioned Request to Service

(<http://www.objigen.com/json/models/yjVPN>)

```
{
  "eligibility": "true",
  "question": "Rapper 21 Savage was detained by which government agency?",
  "answer1": "Pentagon",
  "answer2": "IRS",
  "answer3": "ICE",
  "answer4": "Dept of Housing and Urban Development",
  "associatedContentIDs": [
    [
      "1234",
      "56778"
    ]
  ]
}
```

-Envisioned Response to Service

(<http://www.objigen.com/json/models/oZi>)

```
{
  "contentid": "12345667",
  "userid": "5678",
  "result": "correct"
}
```

-Envisioned Result API Call  
(<http://www.obigen.com/json/models/iWDY>)

## Requirements to Satisfy

ID	Requirement	Acceptance
Q-01	<b>Quality of Service:</b> API must return the result within 550 microseconds.	Manual Test via curl command
Q-02	<b>OUT OF SCOPE:</b> All data loading will be handled under a separate user story. Only a data structure is needed here.	All test data removed and fresh data model
Q-03	<b>Input:</b> Service should take INT parameters from the article page of the User ID and the Content ID of the article.	Manual Test via curl command
Q-04	<b>Output:</b> Service should return eligibility, questions, answers and associated IDs of content in case the user answers incorrectly. Design of Mockups in previous sprints could influence this requirement.	Manual Test via curl command
Q-05	<b>Privacy:</b> Data about a users performance cannot be stored outside of an encrypted user data store. Data in motion does not have to be encrypted however.	GIT check-in requires a diagram of user data upon final check-in
Q-06	<b>Business Rule:</b> If result = abandon without answering then user's eligibility is revoked permanently	Manual Test via curl command
Q-07	<b>Business Rule:</b> If result = correct answer then the user is never promoted again for that article.	Manual Test via curl command
Q-08	<b>Business Rule:</b> If result = incorrect answer user can be prompted again on that article.	Manual Test via curl command
Q-09	<b>Business Rule:</b> If result = error of any kind then the user can be prompted again on that article.	Manual Test via curl command
Q-10	<b>As-Built Documentation:</b> GIT check-ins must include JSON schemas of all input and output from API	Files included with the Source Code during check-in

