

Write an API that will function in following manner

API Basic Details

API URL	http://<ip>/bid	As it is a local environment. So, we are considering http and IP.
Method	POST	
Request Content Type	application/json	
Header	Authorization: Bearer <token>	Only selective tokens will be accepted. For rest return invalid authorization.

Request Payload (That is in JSON format)

Field Name	Type	Mandatory	Validation
id	Alphanumeric	Yes	An entry needed to present in Redis
width	Int	No	If passed, then it needed to be integer only. And if width passed then height also needed to be passed.
height	Int	No	If passed, then it needed to be integer only. And if height passed then width also needed to be passed.
banner.type	enum(1,2)	Yes	1: JavaScript Tag 2: XML Tag

Sample request

```
curl --location --request POST 'http://127.0.0.1/bid' \  
--header 'Content-Type: application/json' \  
--header 'Authorization: Bearer test1234' \  
--data-raw '{
```

```
"id": "id_123",  
"width": 600,  
"height": 328,  
"banner": {  
  "type": 1  
}  
'
```

Note:

1)Validation and checks for graceful failure handling needed to be covered.

Sample Data

Below sample data is stored in Redis. If you are not familiar with Redis then consider any SQL/NoSQL of your choice and store these keys in your known SQL/NoSQL.

In Redis there will be three key presents

- 1)"id_123"
- 2)"id_123_js"
- 3)"id_123_xml"

id_123

will contain impression/click/video_url/video_start/video_end

```
127.0.0.1:6379> hset id_123 impression http://127.0.0.1/impression/123 click
http://127.0.0.1/click/123 video_url http://127.0.0.1/video/123 video_start
http://127.0.0.1/start/123 video_end http://127.0.0.1/end/123
```

(integer) 5

```
127.0.0.1:6379> hgetall id_123
```

- 1) "impression"
- 2) "http://127.0.0.1/impression/123"
- 3) "click"
- 4) "http://127.0.0.1/click/123"
- 5) "video_url"
- 6) "http://127.0.0.1/video/123"
- 7) "video_start"
- 8) "http://127.0.0.1/start/123"
- 9) "video_end"

10) "http://127.0.0.1/end/123"

id_123_js

```
127.0.0.1:6379> set id_123_js "<a href='{click}'><img src='{impression}'></a>"
```

OK

```
127.0.0.1:6379> get id_123_js
```

```
"<a href='{click}'><img src='{impression}'></a>"
```

id_123_xml

```
127.0.0.1:6379> set id_123_xml
```

```
"<Response><impression><![CDATA[{impression}]]</impression><click><![CDATA[{click}]]</click><video_url><![CDATA[{video_url}]]</video_url><video_start><![CDATA[{video_start}]]</video_start><video_end><![CDATA[{video_end}]]</video_end></Response>"
```

OK

```
127.0.0.1:6379> get id_123_xml
```

```
"<Response><impression><![CDATA[{impression}]]</impression><click><![CDATA[{click}]]</click><video_url><![CDATA[{video_url}]]</video_url><video_start><![CDATA[{video_start}]]</video_start><video_end><![CDATA[{video_end}]]</video_end></Response>"
```

Expected Functionality

1)On API call.

2)It will validate the Bearer token

3)Validate the request payload. As mentioned in request payload section.

4)Check if id key("id_123") is present. Fetch it for the impression/click/video_url/video_start/video_end

5)On the basis of type 1/2

i)If "1" is passed then check id_123_js and replace {impression}/{click} with impression/click value fetched in point 4. And response back with content-type as "text/javascript"

ii) If "2" is passed then check "id_123_vast" and replace {impression}/{click}/{video_start}/{video_url}/{video_end} with value fetched in point 4. And response back with content-type as "text/xml"

JavaScript Sample

Request Payload

```
{
  "id": "id_123",
  "width": 600,
  "height": 328,
  "banner": {
    "type": 1
  }
}
```

Response (content-type: text/javascript)

XML Sample

Request Payload

```
{  
  "id": "id_123",  
  "width": 600,  
  "height": 328,  
  "banner": {  
    "type": 2  
  }  
}
```

Response(content-type: text/xml)

```
<Response><impression><![CDATA[http://127.0.0.1/impression/123]]></impression><click><![CDATA[http://127.0.0.1/click/123]]></click><video_url><![CDATA[http://127.0.0.1/video/123]]></video_url><video_start><![CDATA[http://127.0.0.1/start/123]]></video_start><video_end><![CDATA[http://127.0.0.1/end/123]]></video_end></Response>
```