**Akamai Edgeworkers Coding Challenge**

We are allowing end users to login by n-number of devices. If user account / credentials reached n-number of IP address or devices we are blocking them.

This we can do it in by BE also but to avoid unnecessary request to origin server we decided to use Edgeworkers and EdgeKV with in DNS provider.

Edge workers nothing but during the request and response will write logical and achieve whatever we want by JavaScript Code.

**On Request**  **On Response**

onClientRequest onClientResponse

onOriginRequest onOriginResponse

responseProvider

EdgeKV is kind of another database where we can put /save our data by KEY and VALUE.

**Business Use Case**

Every login the device information captured on our end against users. By back end every night will identify the users reached n-number of device those user information will be moved to edgeKV.

When they try to login from another device will throw this error and logout from all the devices. So users need to reach out CRM team will identify whether user is normal or any kind of bot or hack.

So after resolve problem will give option to remove any devices from their devices and they can login.

**HTTP Operations**

The different operations on resources are mapped to corresponding HTTP operations. Each

operation has specific functions and characteristics as explained below.

**GET** - All safe, idempotent cacheable requests to retrieve the data from the platform.

**POST** - Non-Idempotent request which will result in the creation of a new resource or updating

an existing resource

**PUT** - Idempotent request which will result in the updating an existing resource

**DELETE** - Idempotent request to delete or inactivate an existing resource.

**Common Request Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Param** | **Type** | **Description** | **Required** |
| **Accept** | Header | application/json | Y |
| **Content-Type** | Header | application/json | Y |
| **hash-Identity** | Header | Md5(email | phone number) | Y |

**Response Format**

All responses will be JSON. The content type should be specified as application/json

**Common Responses**

Response Status Codes:

|  |  |
| --- | --- |
| **Code** | **Description** |
| **200** | Success |
| **400** | Invalid request |
| **401** | Unauthorized |
| **403** | Forbidden |
| **500** | Failure - Internal Server Error |

**Users Service**

**Login**

URL: {version}/users/login

Http method: POST

***Request***:

*Headers*:

Provided in the common request Headers

\*Important to send hash-Identity : (email | phone number | social login uid)

*Query Parameters*:

Provided in the common request params.

*Body*:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Datatype** | **Description** | **Required** |
| **JSON Body** | Object | Required JSON objects for login | Y |

***Response***:

*Success*:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Datatype** | **Description** |
| **Code** | String | Response code 200 |
| **Message** | String | Success. |
| **Data** | Object | Object |

*Error*:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Datatype** | **Description** |
| **Error** | String | Description of the error |

*Example Request*:

|  |
| --- |
| curl --location --request POST {host}/{v}/users/login' \  --header hash-Identity: 2f38d5d4fe57020a436351a253638dbf' \  --header 'Accept: application/json' \  --header 'Content-Type: application/json' \  --data-raw '{  "type": "traditional",  "deviceId":"918098351840",  "deviceBrand": "PC/MAC",  "data": {  "email": "xxx@yy.zzz",  "password": "8098351840"  }  } |

*Example* Response Body (Success): Status Code 201

|  |
| --- |
| {  "code": 200,  "message": "Success",  "data": {  "status": "ACTIVE",  "deferred": true  }  } |

Response Body (Failure): Status Code 400

|  |
| --- |
| {  error:"Access denied by Requested too many IP address/deviceid",  code:407  } |

**Edge Workers Code:**

|  |
| --- |
| **import { EdgeKV } from "./edgekv.js";**  **import { logger } from "log";**  **import { createResponse } from 'create-response';**  **export async function onClientRequest (request) {**  **let deny\_message={**  **message:"Access denied by Requested too many IP address/deviceid",**  **code:407**  **};**    **try {**  **const edgeKv = new EdgeKV({ namespace: "namespace\_name", group: "group\_name" });**  **const user\_id=request.getHeader('has-Identity')[0] || request.getHeader('has-Identity') || request.getVariable('has-Identity');;**  **logger.log(user\_id);**  **if(user\_id){**  **const resp = await edgeKv.getJson({**  **item: user\_id,**  **default\_value: "5d1aa73526576418cb51262a8759ee158989",**  **});**  **if(resp){**  **request.respondWith(407,**  **{ 'Content-Type': ['application/json'] },**  **JSON.stringify(deny\_message),"Access Denied");**  **}**  **}**    **} catch (error) {**  **logger.log("Edgekv error for getting user details!");**  **return createResponse(**  **400,**  **{ "Content-Type": ["application/json;charset=utf-8"] },**  **JSON.stringify({ error: error.toString() })**  **);**    **}**    **}** |