Source code

**Machine learning DM deployed code**

# retrieve your $IAM\_SERVICE\_CREDENTIALS\_USERNAME, $IAM\_SERVICE\_CREDENTIALS\_PASSWORD, and $IAM\_SERVICE\_CREDENTIALS\_URL from the

# Service credentials associated with your IBM Cloud.

curl --basic --user $IAM\_SERVICE\_CREDENTIALS\_USERNAME:$IAM\_SERVICE\_CREDENTIALS\_PASSWORD $IAM\_SERVICE\_CREDENTIALS\_URL/identity/token

# the above CURL request will return an auth token that you will use as $IAM\_AUTH\_TOKEN in the scoring request below

# TODO: manually define and pass values to be scored below

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization: Bearer $IAM\_AUTH\_TOKEN" -d '{"fields": [$ARRAY\_OF\_INPUT\_FIELDS],"values": [$ARRAY\_OF\_VALUES\_TO\_BE\_SCORED, $ANOTHER\_ARRAY\_OF\_VALUES\_TO\_BE\_SCORED]}' https://us-south.ml.cloud.ibm.com/ml/v4/deployments/bf965454-01da-4318-b27e-4d9fccc32975/predictions

**Java script code**

const XMLHttpRequest = require("xmlhttprequest").XMLHttpRequest;

const btoa = require("btoa");

const iam\_credentials = new Map();

// NOTE: you must manually construct iam\_credentials hash map below using information retrieved

// from your IBM Cloud.

iam\_credentials.set("url", iam\_service\_credentials\_url);

iam\_credentials.set("username", iam\_service\_credentials\_username);

iam\_credentials.set("password", iam\_service\_credentials\_password);

function apiGet(url, username, password, loadCallback, errorCallback){

const oReq = new XMLHttpRequest();

const tokenHeader = "Basic " + btoa((username + ":" + password));

const tokenUrl = url + "/identity/token";

oReq.addEventListener("load", loadCallback);

oReq.addEventListener("error", errorCallback);

oReq.open("GET", tokenUrl);

oReq.setRequestHeader("Authorization", tokenHeader);

oReq.setRequestHeader("Content-Type", "application/json;charset=UTF-8");

oReq.send();

}

function apiPost(scoring\_url, token, payload, loadCallback, errorCallback){

const oReq = new XMLHttpRequest();

oReq.addEventListener("load", loadCallback);

oReq.addEventListener("error", errorCallback);

oReq.open("POST", scoring\_url);

oReq.setRequestHeader("Accept", "application/json");

oReq.setRequestHeader("Authorization", token);

oReq.setRequestHeader("Content-Type", "application/json;charset=UTF-8");

oReq.send(payload);

}

apiGet(iam\_credentials.get("url"),

iam\_credentials.get("username"),

iam\_credentials.get("password"),

function (res) {

let parsedGetResponse;

try {

parsedGetResponse = JSON.parse(this.responseText);

} catch(ex) {

// TODO: handle parsing exception

}

if (parsedGetResponse && parsedGetResponse.token) {

const token = parsedGetResponse.token

const iamToken = "Bearer " + token;

// NOTE: manually define and pass the array(s) of values to be scored in the next line

const payload = '{"fields": [array\_of\_input\_fields], "values": [array\_of\_values\_to\_be\_scored, another\_array\_of\_values\_to\_be\_scored]}';

const scoring\_url = "https://us-south.ml.cloud.ibm.com/ml/v4/deployments/bf965454-01da-4318-b27e-4d9fccc32975/predictions";

apiPost(scoring\_url, iamToken, payload, function (resp) {

let parsedPostResponse;

try {

parsedPostResponse = JSON.parse(this.responseText);

} catch (ex) {

// TODO: handle parsing exception

}

console.log("Scoring response");

console.log(parsedPostResponse);

}, function (error) {

console.log(error);

});

} else {

console.log("Failed to retrieve Bearer token");

}

}, function (err) {

console.log(err);

}

);

Final Prediction code

{ rd: 2355, ms: 23546, ad: 234678, st: "Newyork" }

10/16/2020, 5:49:36 PM[node: e099eb89.aa4258](https://node-red-kfhxm-2020-10-10.eu-gb.mybluemix.net/red/)msg.payload : Object

{ access\_token: "eyJraWQiOiIyMDIwMDkyMjE4MzMiLC…", refresh\_token: "OKByi-g\_0OtIoYJQ9rAR\_TQWSSVH7n…", token\_type: "Bearer", expires\_in: 3600, expiration: 1602854373 … }

10/16/2020, 5:49:38 PM[node: 9ff10849.8c96a8](https://node-red-kfhxm-2020-10-10.eu-gb.mybluemix.net/red/)msg.payload : Object

{ predictions: *array[1]* }

10/16/2020, 5:49:39 PM[node: 7e4eed17.357514](https://node-red-kfhxm-2020-10-10.eu-gb.mybluemix.net/red/)msg.payload : number

222.81243896484375

10/16/2020, 5:49:40 PM[node: 3a7f8360.5298ac](https://node-red-kfhxm-2020-10-10.eu-gb.mybluemix.net/red/)msg.payload : Object

{ rd: 2355, ms: 23546, ad: 234678, st: "New york" }

10/16/2020, 5:49:42 PM[node: e099eb89.aa4258](https://node-red-kfhxm-2020-10-10.eu-gb.mybluemix.net/red/)msg.payload : Object

{ access\_token: "eyJraWQiOiIyMDIwMDkyMjE4MzMiLC…", refresh\_token: "OKA1Rv5pcg7QqMz9REuU4qtyxE0Wu1…", token\_type: "Bearer", expires\_in: 3600, expiration: 1602854379 … }

10/16/2020, 5:49:43 PM[node: 9ff10849.8c96a8](https://node-red-kfhxm-2020-10-10.eu-gb.mybluemix.net/red/)msg.payload : Object

{ predictions: *array[1]* }

10/16/2020, 5:49:44 PM[node: 7e4eed17.357514](https://node-red-kfhxm-2020-10-10.eu-gb.mybluemix.net/red/)msg.payload : number

222.81243896484375