Texto

Descrição gerada automaticamente com confiança média

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

LAB2

import yargs from "yargs";

import { hideBin } from "yargs/helpers";

import {

  currencyFormatter,

  provinces,

  fullNameAndProvincePromise,

  transferPaymentsFromWebPromise,

  transferPaymentForProvincePromise,

} from "./lab2\_routines.js";

const argv = yargs(hideBin(process.argv))

  .options({

    firstName: {

      demandOption: true,

      alias: "fname",

      describe: "Resident’s first name",

      string: true,

    },

    lastName: {

      demandOption: true,

      alias: "lname",

      describe: "Resident’s last name",

      string: true,

    },

    province: {

      demandOption: true,

      alias: "prov",

      describe: "Resident’s home province",

      string: true,

      choices: provinces.map((province) => province.code),

    },

  })

  .help()

  .alias("help", "h")

    .parse();

  chainFunction(argv.firstName,argv.lastName,argv.province)

function chainFunction(firstName, lastName, provincialCode) {

  return fullNameAndProvincePromise(firstName, lastName, provincialCode)

      .then((result1) => {

          console.log()

      return transferPaymentsFromWebPromise();

    })

      .then((result2) => {

          console.log("Lab 2")

          console.log()

      return transferPaymentForProvincePromise(

        result2,

        provincialCode.toLowerCase()

      );

    })

      .then((result3) => {

          let provinceName = provinces.find(x=>x.code.toLowerCase() === provincialCode.toLowerCase()).name

      console.log(

        `${firstName}, ${lastName} lives in ${provinceName}. It received ${result3} in transfer payments.`

      );

      return result3;

    })

    .catch((error) => {

      console.error("Error in promise chain:", error);

      throw error;

    });

}

// Example usage:

// chainFunction("John", "Doe", "SK");

LAB2\_ROUTINES

import got from "got";

const provinces = [

  { code: "NS", name: "Nova Scotia" },

  { code: "NL", name: "Newfoundland" },

  { code: "NB", name: "New Brunswick" },

  { code: "PE", name: "Prince Edward Island" },

  { code: "QC", name: "Quebec" },

  { code: "ON", name: "Ontario" },

  { code: "MB", name: "Manitoba" },

  { code: "SK", name: "Saskatchewan" },

  { code: "AB", name: "Alberta" },

  { code: "BC", name: "British Columbia" },

  { code: "NT", name: "North West Territories" },

  { code: "NU", name: "Nunavut" },

  { code: "YT", name: "Yukon Territory" },

];

const FISCALYEAR = "2022-2023";

// Create a currency formatter.

const currencyFormatter = (numberToFormat) =>

  new Intl.NumberFormat("en-US", {

    style: "currency",

    currency: "USD",

    minimumFractionDigits: 0,

  }).format(numberToFormat);

function fullNameAndProvincePromise(firstName, lastName, provincialCode) {

  return new Promise((resolve, reject) => {

      const provinceName = provinces.find((x) => x.code == provincialCode).name;

      const result = firstName + " " + lastName + " " + provinceName;

      resolve(result);

  });

}

function transferPaymentsFromWebPromise() {

  let srcAddr =

    "http://www.infrastructure.gc.ca/alt-format/opendata/transfer-program-programmes-de-transfert-bil.json";

  return new Promise((resolve, reject) => {

      got(srcAddr, { responseType: "json" })

        .then((response) => {

          // let ont = response.body.gtf.on["2022-2023"];

          let ont = response;

          let result = ont;

          resolve(result);

        })

        .catch((err) => {

          console.log(`Error ==> ${err}`);

          reject(err);

        });

  });

}

function transferPaymentForProvincePromise(gocData,provCode) {

  return new Promise((resolve, reject) => {

      let ont = gocData.body.gtf[provCode]["2022-2023"];

      let result = ont;

      resolve(currencyFormatter( result));

    }, 1000);

}

export {

  provinces,

  currencyFormatter,

  fullNameAndProvincePromise,

  transferPaymentsFromWebPromise,

  transferPaymentForProvincePromise,

};