

```

export default async function main(args) {
  // Explanation video: https://youtu.be/7EuXJvsNhDc

  // Validate input variables from args - find the "validateInputs" function below
  const invalidFieldTrace = validateInputs(args)

  if (invalidFieldTrace) return invalidFieldTrace;

  // Extract input variables from args (destructuring)
  const {
    User_lastName,
    User_firstName,
    User_PhoneNumber,
    User_email,
    User_Age,
    User_Ethnicity,
    System_decision,
    webhook
  } = args.inputVars;

  /**
   * Build the payload required by your Webhook
   */
  const payload = {
    lastName: User_lastName,
    firstName: User_firstName,
    phoneNumber: User_PhoneNumber,
    email: User_email,
    age: User_Age,
    ethnicity: User_Ethnicity,
    decision: System_decision
  };

  /**
   * Configure the fetch request
   */
  const config = {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify(payload)
  };

  try {
    // Make the API call
    const response = await fetch(webhook, config);
  }

```

```

// Check if the response status is OK
if (!response.ok) {
  throw new Error(`HTTP status code ${response.status}`);
}

// Extract the JSON body from the response
const responseBody = await response.json();

// Validate the responseBody structure as expected
if (!responseBody || typeof responseBody !== 'object') {
  throw new Error("Invalid or missing response body");
}

// Create the success return object with extracted data
return {
  next: {
    path: 'success'
  },
  trace: [
    {
      type: "debug",
      payload: {
        message: 'Webhook call successful'
      }
    },
  ],
};
} catch (error) {
  return {
    next: {
      path: 'error'
    },
    trace: [{
      type: "debug",
      payload: {
        message: `Webhook call error: ${error.message}`
      }
    },
  ]
};
}

/**
 * Validate all the inputs of the Voiceflow function
 */
function validateInputs(args) {
  // Extract input variables from args (destructuring)
  const {
    User_lastName,
  }

```

```

    User_firstName,
    User_PhoneNumber,
    User_email,
    User_Age,
    User_Ethnicity,
    System_decision,
    webhook
} = args.inputVars;

// Check each field
if (!User_lastName) {
    return {
        next: { path: 'error' },
        trace: [{ type: "debug", payload: { message: "Missing required input variable:
User_lastName" } }]
    };
}
if (!User_firstName) {
    return {
        next: { path: 'error' },
        trace: [{ type: "debug", payload: { message: "Missing required input variable:
User_firstName" } }]
    };
}
if (!User_PhoneNumber) {
    return {
        next: { path: 'error' },
        trace: [{ type: "debug", payload: { message: "Missing required input variable:
User_PhoneNumber" } }]
    };
}
if (!User_email) {
    return {
        next: { path: 'error' },
        trace: [{ type: "debug", payload: { message: "Missing required input variable:
User_email" } }]
    };
}
if (!User_Age) {
    return {
        next: { path: 'error' },
        trace: [{ type: "debug", payload: { message: "Missing required input variable: User_Age"
} }]
    };
}
if (!User_Ethnicity) {
    return {
        next: { path: 'error' },

```

```

        trace: [{ type: "debug", payload: { message: "Missing required input variable:
User_Ethnicity" } }]
    };
}
if (!System_decision) {
    return {
        next: { path: 'error' },
        trace: [{ type: "debug", payload: { message: "Missing required input variable:
System_decision" } }]
    };
}
if (!webhook) {
    return {
        next: { path: 'error' },
        trace: [{ type: "debug", payload: { message: "Missing required input variable: webhook"
} }]
    };
}

    return null;
}

}

```