

Share output request



this output was generated:

Could you share the output with me so I can take a look?



The following is output:

```
{
  "info": {
    "name": "C2M Testing Collection",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "item": [
    {
      "name": "jobs",
      "description": "",
      "item": [
        {
          "name": "submit",
          "description": "",
          "item": [
            {
              "name": "single",
              "description": "",
              "item": [
                {
                  "name": "doc",
                  "description": "",
                  "item": [
                    {
                      "id": "72426c61-38b1-418c-9aee-5ae71112e83c",
                      "name": "Submit a single document to multiple recipients",
                      "request": {
                        "name": "Submit a single document to multiple recipients",
                        "description": {},
                        "url": {
                          "path": [
                            "jobs",
                            "submit",
                            "single",
                            "doc"
                          ]
                        }
                      }
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

```

"host": [
  "{{baseUrl}}"
],
"query": [],
"variable": []
},
"header": [
  {
    "key": "Content-Type",
    "value": "application/json"
  },
  {
    "key": "Accept",
    "value": "application/json"
  }
],
"method": "POST",
"body": {
  "mode": "raw",
  "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"strictly\"\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"Greyson\",\n      \"lastName\": \"Brody\",\n      \"address1\": \"477 Manuel Camp\",\n      \"city\": \"Weymouth Town\",\n      \"state\": \"Florida\",\n      \"zip\": \"54621-7776\",\n      \"country\": \"Slovenia\",\n      \"nickName\": \"Earnestine\",\n      \"address2\": \"7611 Garret Trail\",\n      \"address3\": \"55666 Sonia Center\",\n      \"phoneNumber\": \"1-825-738-9413 x638\"\n    },\n    {\n      \"firstName\": \"Everett\",\n      \"lastName\": \"Elza\",\n      \"address1\": \"76198 Theresia Radial\",\n      \"city\": \"Fort Bertram\",\n      \"state\": \"California\",\n      \"zip\": \"60954\",\n      \"country\": \"Ghana\",\n      \"nickName\": \"Gregoria\",\n      \"address2\": \"5466 Meta Skyway\",\n      \"address3\": \"2850 Rafaela Viaduct\",\n      \"phoneNumber\": \"(385) 341-9118\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"3793-047044-06536\",\n      \"expirationDate\": {\n        \"month\": 458,\n        \"year\": 963\n      },\n      \"cvv\": 905\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"12075860\",\n      \"amountDue\": 575.65\n    },\n    \"achDetails\": {\n      \"routingNumber\": \"separate\",\n      \"accountNumber\": \"violently\",\n      \"checkDigit\": 410\n    },\n    \"creditAmount\": {\n      \"amount\": 896.09,\n      \"currency\": \"CAD\"\n    }\n  },\n  \"tags\": [\n    \"<string>\",\n    \"<string>\"\n  ]\n}",
  "options": {
    "raw": {
      "headerFamily": "json",
      "language": "json"
    }
  }
},
"auth": null
},
"response": [
  {
    "id": "c75de882-bad1-4bc8-b181-db96cfecd9a9",
    "name": "Job submission accepted",

```

```

"originalRequest": {
  "url": {
    "path": [
      "jobs",
      "submit",
      "single",
      "doc"
    ],
    "host": [
      "{{baseUrl}}"
    ],
    "query": [],
    "variable": []
  },
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    },
    {
      "key": "Accept",
      "value": "application/json"
    },
    {
      "description": {
        "content": "Added as a part of security scheme: bearer",
        "type": "text/plain"
      },
      "key": "Authorization",
      "value": "Bearer <token>"
    }
  ],
  "method": "POST",
  "body": {
    "mode": "raw",
    "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\", \n    \n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\", \n    \"layout\": \"portrait\", \n    \"mailclass\": \"largeEnvelope\", \n    \"paperType\": \"letter\", \n    \"printOption\": \"color\", \n    \"envelope\": \"flat\" \n  }, \n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\", \n      \"lastName\": \"<string>\", \n      \"address1\": \"<string>\", \n      \"city\": \"<string>\", \n      \"state\": \"<string>\", \n      \"zip\": \"<string>\", \n      \"country\": \"<string>\", \n      \"nickName\": \"<string>\", \n      \"address2\": \"<string>\", \n      \"address3\": \"<string>\", \n      \"phoneNumber\": \"<string>\" \n    }, \n    {\n      \"firstName\": \"<string>\", \n      \"lastName\": \"<string>\", \n      \"address1\": \"<string>\", \n      \"city\": \"<string>\", \n      \"state\": \"<string>\", \n      \"zip\": \"<string>\", \n      \"country\": \"<string>\", \n      \"nickName\": \"<string>\", \n      \"address2\": \"<string>\", \n      \"address3\": \"<string>\", \n      \"phoneNumber\": \"<string>\" \n    } \n  ], \n  \"paymentDetails\": {\n    \"billingType\": \"invoice\", \n    \"creditCardDetails\": {\n      \"cardType\": \"discover\", \n      \"cardNumber\": \"<string>\", \n      \"expirationDate\": {\n        \"month\": \"<integer>\", \n        \"year\": \"<integer>\" \n      }, \n      \"cvv\": \"<integer>\" \n    }, \n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\", \n      \"amountDue\": \"<number>\" \n    }, \n    \"achDetails\": {\n"
  }
}

```

```

\"routingNumber\": \"<string>\",\n  \"accountNumber\": \"<string>\",\n  \"checkDigit\": \"
<integer>\",\n  \"creditAmount\": {\n    \"amount\": \"<number>\",\n    \"currency\":
\"CAD\"\n  },\n  \"tags\": [\n    \"<string>\",\n    \"<string>\"\n  ]\n}\",
  \"options\": {
    \"raw\": {
      \"headerFamily\": \"json\",
      \"language\": \"json\"
    }
  }
},
\"status\": \"OK\",
\"code\": 200,
\"header\": [
  {
    \"key\": \"Content-Type\",
    \"value\": \"application/json\"
  }
],
\"body\": \"{\\n  \\\"jobId\\\": \\\"pro\\\",\\n  \\\"status\\\": \\\"vacation\\\"\\n}\",
\"cookie\": [],
\"_postman_previewlanguage\": \"json\"
},
{
  \"id\": \"0f0cdd33-2fe4-4be5-aa48-3510f580dce0\",
  \"name\": \"Invalid request\",
  \"originalRequest\": {
    \"url\": {
      \"path\": [
        \"jobs\",
        \"submit\",
        \"single\",
        \"doc\"
      ],
      \"host\": [
        \"{{baseUrl}}\"
      ],
      \"query\": [],
      \"variable\": []
    },
    \"header\": [
      {
        \"key\": \"Content-Type\",
        \"value\": \"application/json\"
      },
      {
        \"description\": {
          \"content\": \"Added as a part of security scheme: bearer\",
          \"type\": \"text/plain\"
        },
        \"key\": \"Authorization\",

```

```

      "value": "Bearer <token>"
    }
  ],
  "method": "POST",
  "body": {
    "mode": "raw",
    "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\",\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    },\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"<string>\",\n      \"expirationDate\": {\n        \"month\": \"<integer>\",\n        \"year\": \"<integer>\"\n      },\n      \"cvv\": \"<integer>\"\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\",\n      \"amountDue\": \"<number>\",\n      \"achDetails\": {\n        \"routingNumber\": \"<string>\",\n        \"accountNumber\": \"<string>\",\n        \"checkDigit\": \"<integer>\"\n      },\n      \"creditAmount\": {\n        \"amount\": \"<number>\",\n        \"currency\": \"CAD\"\n      }\n    },\n    \"tags\": [\n      \"<string>\",\n      \"<string>\"\n    ]\n  },\n  \"options\": {\n    \"raw\": {\n      \"headerFamily\": \"json\",\n      \"language\": \"json\"\n    }\n  }\n}\n",
    "status": "Bad Request",
    "code": 400,
    "header": [],
    "cookie": [],
    "_postman_previewlanguage": "text"
  },
  {
    "id": "36bd10fe-c455-40b4-bef0-391a0eb8dddb",
    "name": "Unauthorized",
    "originalRequest": {
      "url": {
        "path": [
          "jobs",
          "submit",
          "single",
          "doc"
        ],
        "host": [

```

```

    "{{baseUrl}}"
  ],
  "query": [],
  "variable": []
},
"header": [
  {
    "key": "Content-Type",
    "value": "application/json"
  },
  {
    "description": {
      "content": "Added as a part of security scheme: bearer",
      "type": "text/plain"
    },
    "key": "Authorization",
    "value": "Bearer <token>"
  }
],
"method": "POST",
"body": {
  "mode": "raw",
  "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\"\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    },\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"<string>\",\n      \"expirationDate\": {\n        \"month\": \"<integer>\",\n        \"year\": \"<integer>\"\n      },\n      \"cvv\": \"<integer>\"\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\",\n      \"amountDue\": \"<number>\",\n      \"achDetails\": {\n        \"routingNumber\": \"<string>\",\n        \"accountNumber\": \"<string>\",\n        \"checkDigit\": \"<integer>\"\n      },\n      \"creditAmount\": {\n        \"amount\": \"<number>\",\n        \"currency\": \"CAD\"\n      }\n    },\n    \"tags\": [\n      \"<string>\",\n      \"<string>\"\n    ]\n  },\n  \"options\": {\n    \"raw\": {\n      \"headerFamily\": \"json\",\n      \"language\": \"json\"\n    }\n  }\n}\n",
  "status": "Unauthorized",
  "code": 401,

```

```

    "header": [],
    "cookie": [],
    "_postman_previewlanguage": "text"
  }
],
"event": [],
"protocolProfileBehavior": {
  "disableBodyPruning": true
}
},
{
  "name": "jobTemplate",
  "description": "",
  "item": [
    {
      "id": "da8c2514-24f3-4b99-b212-4a3eac840a7d",
      "name": "Submit a document using a job template",
      "request": {
        "name": "Submit a document using a job template",
        "description": {},
        "url": {
          "path": [
            "jobs",
            "submit",
            "single",
            "doc",
            "jobTemplate"
          ],
          "host": [
            "{{baseUrl}}"
          ],
          "query": [],
          "variable": []
        },
        "header": [
          {
            "key": "Content-Type",
            "value": "application/json"
          }
        ],
        "method": "POST",
        "body": {
          "mode": "raw",
          "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"indeed\"\n  },\n  \"jobTemplate\": \"provided\",\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"Nickolas\",\n      \"lastName\": \"Gardner\",\n      \"address1\": \"454 Park Road\",\n      \"city\": \"South Adrianna\",\n      \"state\": \"Indiana\",\n      \"zip\": \"92487-2650\",\n      \"country\": \"Dominica\",\n      \"nickName\": \"Eleanora\",\n      \"address2\": \"894 Gillian Glen\",\n      \"address3\": \"908 Rolfson Green\",\n      \"phoneNumber\": \"684.230.5430 x35208\"\n    },\n    {\n      \"firstName\": \"Neha\",\n      \"lastName\": \"Taurean\",\n      \"address1\": \"50802 Peter Ford\",\n      \"city\": \"Fort Anahiberg\",\n      \"state\": \"New Jersey\",\n      \"zip\": \"45615-

```

```
8952",\n  \"country\": \"Dominica\", \n  \"nickName\": \"Nick\", \n  \"address2\": \"253
Emmalee Groves\", \n  \"address3\": \"27185 Willms Harbor\", \n  \"phoneNumber\": \"(833)
543-6492 x29927\" \n } \n ], \n \"jobOptions\": { \n  \"documentClass\": \"businessLetter\", \n
\"layout\": \"portrait\", \n  \"mailclass\": \"priorityMail\", \n  \"paperType\": \"letter\", \n
\"printOption\": \"grayscale\", \n  \"envelope\": \"postcard\" \n }, \n \"paymentDetails\": { \n
\"billingType\": \"ach\", \n  \"creditCardDetails\": { \n  \"cardType\": \"mastercard\", \n
\"cardNumber\": \"6011-8060-0084-7529\", \n  \"expirationDate\": { \n  \"month\": 70, \n
\"year\": 393 \n  }, \n  \"cvv\": 463 \n  }, \n  \"invoiceDetails\": { \n  \"invoiceNumber\":
\"85785810\", \n  \"amountDue\": 66.26 \n  }, \n  \"achDetails\": { \n  \"routingNumber\":
\"from\", \n  \"accountNumber\": \"fencing\", \n  \"checkDigit\": 367 \n  }, \n
\"creditAmount\": { \n  \"amount\": 601.6, \n  \"currency\": \"USD\" \n  } \n }, \n \"tags\": [ \n
\"<string>\", \n  \"<string>\" \n ] \n }\",
  \"options\": {
    \"raw\": {
      \"headerFamily\": \"json\",
      \"language\": \"json\"
    }
  },
  \"auth\": null
},
```

it was generated by the following node program:

```
#!/usr/bin/env node
```

```
/**
 * Adds random test data only inside "request.body.raw" (JSON bodies) of a Postman Collection.
 * Replaces placeholders (<string>, <number>, <integer>, <boolean>) with type-specific random
 values.
 * Optionally injects erroneous values based on --error-rate percentage.
 *
 * New: --preview option prints a before/after diff of the first modified "raw" body instead of
 writing the file.
 *
 * Usage:
 * node addRandomDataToRaw.js --input input.json --output output.json [--error-rate 20] [--
 preview]
 */
```

```
const fs = require('fs');
const path = require('path');
const { faker } = require('@faker-js/faker');
```

```
// --- Parse CLI args ---
function parseArgs() {
  const args = process.argv.slice(2);
  const argMap = {};
  for (let i = 0; i < args.length; i++) {
    if (args[i].startsWith('--')) {
      const key = args[i].substring(2);
```



```

    const val = args[i + 1] && !args[i + 1].startsWith('--') ? args[i + 1] : true;
    argMap[key] = val;
  }
}
if (!argMap.input) {
  console.error('Usage: node addRandomDataToRaw.js --input <file> --output <file> [--error-rate
<0-100>] [--preview]');
  process.exit(1);
}
argMap.errorRate = parseInt(argMap['error-rate'] || '0', 10);
return argMap;
}

// --- Generate type-based random values ---
function typeBasedRandomValue(typeHint) {
  switch (typeHint) {
    case '<number>':
      return parseFloat((Math.random() * 1000).toFixed(2));
    case '<integer>':
      return Math.floor(Math.random() * 1000);
    case '<boolean>':
      return Math.random() > 0.5;
    case '<string>':
    default:
      return faker.word.sample();
  }
}

function randomValue(key, typeHint = '<string>') {
  if (typeHint !== '<string>') return typeBasedRandomValue(typeHint);

  const k = key.toLowerCase();
  if (k.includes('first') || k.includes('name')) return faker.person.firstName();
  if (k.includes('last')) return faker.person.lastName();
  if (k.includes('email')) return faker.internet.email();
  if (k.includes('phone')) return faker.phone.number();
  if (k.includes('address')) return faker.location.streetAddress();
  if (k.includes('city')) return faker.location.city();
  if (k.includes('state')) return faker.location.state();
  if (k.includes('zip') || k.includes('postal')) return faker.location.zipCode();
  if (k.includes('country')) return faker.location.country();
  if (k.includes('invoice')) return faker.finance.accountNumber();
  if (k.includes('amount')) return faker.finance.amount();
  if (k.includes('card')) return faker.finance.creditCardNumber();
  if (k.includes('cvv')) return faker.finance.creditCardCVV();
  if (k.includes('month')) return faker.number.int({ min: 1, max: 12 }).toString();
  if (k.includes('year')) return (new Date().getFullYear() + 1).toString();
  return faker.word.sample();
}

function erroneousValue() {

```

```

const pool = ['!!!@@@###', '', '<script>alert(1)</script>', null, undefined, '⚠INVALID⚠'];
return pool[Math.floor(Math.random() * pool.length)];
}

// --- Replace placeholders in raw JSON string ---
function fillRaw(rawStr, errorRate) {
  try {
    const jsonObj = JSON.parse(rawStr);
    replaceValues(jsonObj, errorRate);
    return JSON.stringify(jsonObj, null, 2);
  } catch {
    return rawStr; // If body is not valid JSON, leave it as is
  }
}

function replaceValues(obj, errorRate) {
  if (Array.isArray(obj)) {
    obj.forEach((item) => replaceValues(item, errorRate));
  } else if (typeof obj === 'object' && obj !== null) {
    Object.keys(obj).forEach((key) => {
      if (typeof obj[key] === 'string' && obj[key].startsWith('<') && obj[key].endsWith('>')) {
        const injectError = Math.random() < errorRate / 100;
        const typeHint = obj[key];
        obj[key] = injectError ? erroneousValue() : randomValue(key, typeHint);
      } else if (typeof obj[key] === 'object') {
        replaceValues(obj[key], errorRate);
      }
    });
  }
}

// --- Traverse collection ---
function processCollection(item, errorRate, previewMode) {
  let previewData = { before: null, after: null, captured: false };

  function processItem(it) {
    if (Array.isArray(it)) {
      it.forEach((sub) => processItem(sub));
    } else if (it && typeof it === 'object') {
      if (it.request && it.request.body && it.request.body.raw) {
        const originalRaw = it.request.body.raw;
        const newRaw = fillRaw(originalRaw, errorRate);
        it.request.body.raw = newRaw;

        if (previewMode && !previewData.captured) {
          previewData.before = originalRaw;
          previewData.after = newRaw;
          previewData.captured = true;
        }
      }
    }
    if (it.response) {

```

```

        it.response.forEach((resp) => {
            if (resp.body) resp.body = fillRaw(resp.body, errorRate);
        });
    }
    if (it.item) processItem(it.item);
}

processItem(item);
return previewData;
}

// --- Main ---
(function main() {
    const { input, output, errorRate, preview } = parseArgs();
    const collection = JSON.parse(fs.readFileSync(path.resolve(input), 'utf-8'));

    const previewData = processCollection(collection.item, errorRate, preview);

    if (preview) {
        console.log('--- PREVIEW MODE ---');
        console.log('\n--- BEFORE ---');
        console.log(previewData.before);
        console.log('\n--- AFTER ---');
        console.log(previewData.after);
        console.log('\n(No files were written.)');
    } else {
        fs.writeFileSync(path.resolve(output), JSON.stringify(collection, null, 2));
        console.log('✅ (U+2705)Random data added. Output saved to ${output}');
    }
})();

```

This is an example of the input:

```

{
  "info": {
    "name": "C2M Testing Collection",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "item": [
    {
      "name": "jobs",
      "description": "",
      "item": [
        {
          "name": "submit",
          "description": "",
          "item": [
            {
              "name": "single",

```

```

"description": "",
"item": [
  {
    "name": "doc",
    "description": "",
    "item": [
      {
        "id": "72426c61-38b1-418c-9aee-5ae71112e83c",
        "name": "Submit a single document to multiple recipients",
        "request": {
          "name": "Submit a single document to multiple recipients",
          "description": {},
          "url": {
            "path": [
              "jobs",
              "submit",
              "single",
              "doc"
            ],
            "host": [
              "{{baseUrl}}"
            ],
            "query": [],
            "variable": []
          },
          "header": [
            {
              "key": "Content-Type",
              "value": "application/json"
            },
            {
              "key": "Accept",
              "value": "application/json"
            }
          ],
          "method": "POST",
          "body": {
            "mode": "raw",
            "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\",\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    },\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"

```

```
<string>\",\n  \"expirationDate\": {\n    \"month\": \"<integer>\",\n    \"year\": \"<integer>\"\n  },\n  \"cvv\": \"<integer>\"\n },\n  \"invoiceDetails\": {\n    \"invoiceNumber\": \"<string>\",\n    \"amountDue\": \"<number>\",\n    \"achDetails\": {\n      \"routingNumber\": \"<string>\",\n      \"accountNumber\": \"<string>\",\n      \"checkDigit\": \"<integer>\"\n    },\n    \"creditAmount\": {\n      \"amount\": \"<number>\",\n      \"currency\": \"CAD\"\n    }\n  },\n  \"tags\": [\n    \"<string>\",\n    \"<string>\"\n ]\n }\",
  \"options\": {
    \"raw\": {
      \"headerFamily\": \"json\",
      \"language\": \"json\"
    }
  },
  \"auth\": null
},
\"response\": [
  {
    \"id\": \"c75de882-bad1-4bc8-b181-db96cfecd9a9\",
    \"name\": \"Job submission accepted\",
    \"originalRequest\": {
      \"url\": {
        \"path\": [
          \"jobs\",
          \"submit\",
          \"single\",
          \"doc\"
        ],
        \"host\": [
          \"{{baseUrl}}\"
        ],
        \"query\": [],
        \"variable\": []
      },
      \"header\": [
        {
          \"key\": \"Content-Type\",
          \"value\": \"application/json\"
        },
        {
          \"key\": \"Accept\",
          \"value\": \"application/json\"
        },
        {
          \"description\": {
            \"content\": \"Added as a part of security scheme: bearer\",
            \"type\": \"text/plain\"
          },
          \"key\": \"Authorization\",
          \"value\": \"Bearer <token>\"
        }
      ]
    }
  },
  {
    \"id\": \"c75de882-bad1-4bc8-b181-db96cfecd9a9\",
    \"name\": \"Job submission accepted\",
    \"originalRequest\": {
      \"url\": {
        \"path\": [
          \"jobs\",
          \"submit\",
          \"single\",
          \"doc\"
        ],
        \"host\": [
          \"{{baseUrl}}\"
        ],
        \"query\": [],
        \"variable\": []
      },
      \"header\": [
        {
          \"key\": \"Content-Type\",
          \"value\": \"application/json\"
        },
        {
          \"key\": \"Accept\",
          \"value\": \"application/json\"
        },
        {
          \"description\": {
            \"content\": \"Added as a part of security scheme: bearer\",
            \"type\": \"text/plain\"
          },
          \"key\": \"Authorization\",
          \"value\": \"Bearer <token>\"
        }
      ]
    }
  }
]
```

```

"method": "POST",
"body": {
  "mode": "raw",
  "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\\\"\\n
tions\\\": {\n    \"documentClass\\\": \"businessLetter\\\", \n    \"layout\\\": \"portrait\\\", \n
\\\"largeEnvelope\\\", \n    \"paperType\\\": \"letter\\\", \n    \"printOption\\\": \"color\\\", \n
\\\"flat\\\"\\n  }, \n  \"recipientAddresses\\\": [\n    {\n      \"firstName\\\": \"<string>\\\", \n
\\\"<string>\\\", \n      \"address1\\\": \"<string>\\\", \n      \"city\\\": \"<string>\\\", \n
string>\\\", \n      \"zip\\\": \"<string>\\\", \n      \"country\\\": \"<string>\\\", \n      \"nickName\\\":
\\\"<string>\\\"\\n    }, \n    {\n      \"firstName\\\": \"<string>\\\", \n      \"lastName\\\": \"
\\\"address1\\\": \"<string>\\\", \n      \"city\\\": \"<string>\\\", \n      \"state\\\": \"
\\\"zip\\\": \"<string>\\\", \n      \"country\\\": \"<string>\\\", \n      \"nickName\\\": \"
\\\"address2\\\": \"<string>\\\", \n      \"address3\\\": \"<string>\\\", \n
ber\\\": \"<string>\\\"\\n    } \n  ], \n  \"paymentDetails\\\": {\n    \"billingType\\\":
\\\"creditCardDetails\\\": {\n      \"cardType\\\": \"discover\\\", \n      \"cardNumber\\\": \"
\\\"expirationDate\\\": {\n        \"month\\\": \"<integer>\\\", \n        \"year\\\": \"
\\\" \n      }, \n      \"cvv\\\": \"<integer>\\\"\\n    }, \n    \"invoiceDetails\\\": {\n
ber\\\": \"<string>\\\", \n      \"amountDue\\\": \"<number>\\\"\\n    }, \n    \"achDetails\\\": {\n
ber\\\": \"<string>\\\", \n      \"accountNumber\\\": \"<string>\\\", \n      \"checkDigit\\\": \"
\\\" \n    }, \n    \"creditAmount\\\": {\n      \"amount\\\": \"<number>\\\", \n      \"currency\\\":
\\\" \n    }, \n    \"tags\\\": [\n      \"<string>\\\", \n      \"<string>\\\"\\n    ] \n  },
  \"options\\\": {
    \"raw\\\": {
      \"headerFamily\\\": \"json\",
      \"language\\\": \"json\"
    }
  }
},
\"status\\\": \"OK\",
\"code\\\": 200,
\"header\\\": [
  {
    \"key\\\": \"Content-Type\",
    \"value\\\": \"application/json\"
  }
],
\"body\\\": \"{\\n  \\\"jobId\\\": \"<string>\\\", \n  \\\"status\\\": \"<string>\\\"\\n}\",
\"cookie\\\": [],
\"_postman_previewlanguage\\\": \"json\"
},
{
  \"id\\\": \"0f0cdd33-2fe4-4be5-aa48-3510f580dce0\",
  \"name\\\": \"Invalid request\",
  \"originalRequest\\\": {
    \"url\\\": {
      \"path\\\": [
        \"jobs\",
        \"submit\",
        \"single\",

```

```

"doc"
],
"host": [
  "{{baseUrl}}"
],
"query": [],
"variable": []
},
"header": [
  {
    "key": "Content-Type",
    "value": "application/json"
  },
  {
    "description": {
      "content": "Added as a part of security scheme: bearer",
      "type": "text/plain"
    },
    "key": "Authorization",
    "value": "Bearer <token>"
  }
],
"method": "POST",
"body": {
  "mode": "raw",
  "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\"\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    },\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"<string>\",\n      \"expirationDate\": {\n        \"month\": \"<integer>\",\n        \"year\": \"<integer>\"\n      },\n      \"cvv\": \"<integer>\"\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\",\n      \"amountDue\": \"<number>\",\n      \"achDetails\": {\n        \"routingNumber\": \"<string>\",\n        \"accountNumber\": \"<string>\",\n        \"checkDigit\": \"<integer>\"\n      },\n      \"creditAmount\": {\n        \"amount\": \"<number>\",\n        \"currency\": \"CAD\"\n      }\n    }\n  },\n  \"tags\": [\n    \"<string>\",\n    \"<string>\"\n  ]\n}"
  "options": {
    "raw": {
      "headerFamily": "json",
      "language": "json"
    }
  }
}

```

```

},
"status": "Bad Request",
"code": 400,
"header": [],
"cookie": [],
"_postman_previewlanguage": "text"
},
{
  "id": "36bd10fe-c455-40b4-bef0-391a0eb8dddb",
  "name": "Unauthorized",
  "originalRequest": {
    "url": {
      "path": [
        "jobs",
        "submit",
        "single",
        "doc"
      ],
      "host": [
        "{{baseUrl}}"
      ],
      "query": [],
      "variable": []
    },
    "header": [
      {
        "key": "Content-Type",
        "value": "application/json"
      },
      {
        "description": {
          "content": "Added as a part of security scheme: bearer",
          "type": "text/plain"
        },
        "key": "Authorization",
        "value": "Bearer <token>"
      }
    ],
    "method": "POST",
    "body": {
      "mode": "raw",
      "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\"\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    },\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\"\n    }\n  ]\n}"
    }
  }
}

```



```

<string>\",\n  \"address2\": \"<string>\",\n  \"address3\": \"<string>\",\n
\"phoneNumber\": \"<string>\",\n  ],\n  \"paymentDetails\": {\n    \"billingType\":
\"invoice\", \n    \"creditCardDetails\": {\n      \"cardType\": \"discover\", \n      \"cardNumber\": \"
<string>\", \n      \"expirationDate\": {\n        \"month\": \"<integer>\", \n        \"year\": \"
<integer>\", \n      }, \n      \"cvv\": \"<integer>\", \n    }, \n    \"invoiceDetails\": {\n
      \"invoiceNumber\": \"<string>\", \n      \"amountDue\": \"<number>\", \n    }, \n    \"achDetails\": {\n
      \"routingNumber\": \"<string>\", \n      \"accountNumber\": \"<string>\", \n      \"checkDigit\": \"
<integer>\", \n    }, \n    \"creditAmount\": {\n      \"amount\": \"<number>\", \n      \"currency\":
\"CAD\", \n    }, \n    \"tags\": [\n      \"<string>\", \n      \"<string>\", \n    ]\n  },
    \"options\": {
      \"raw\": {
        \"headerFamily\": \"json\",
        \"language\": \"json\"
      }
    }
  },
  \"status\": \"Unauthorized\",
  \"code\": 401,
  \"header\": [],
  \"cookie\": [],
  \"_postman_previewlanguage\": \"text\"
},
\"event\": [],
\"protocolProfileBehavior\": {
  \"disableBodyPruning\": true
},
{
  \"name\": \"jobTemplate\",
  \"description\": \"\",
  \"item\": [
    {
      \"id\": \"da8c2514-24f3-4b99-b212-4a3eac840a7d\",
      \"name\": \"Submit a document using a job template\",
      \"request\": {
        \"name\": \"Submit a document using a job template\",
        \"description\": {},
        \"url\": {
          \"path\": [
            \"jobs\",
            \"submit\",
            \"single\",
            \"doc\",
            \"jobTemplate\"
          ],
          \"host\": [
            \"{{baseUrl}}\"
          ],
          \"query\": [],

```

```

    "variable": []
  },
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ],
  "method": "POST",
  "body": {
    "mode": "raw",
    "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\"\n  },\n  \"jobTemplate\": \"<string>\",\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    },\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    }\n  ],\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"priorityMail\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"grayscale\",\n    \"envelope\": \"postcard\"\n  },\n  \"paymentDetails\": {\n    \"billingType\": \"ach\",\n    \"creditCardDetails\": {\n      \"cardType\": \"mastercard\",\n      \"cardNumber\": \"<string>\",\n      \"expirationDate\": {\n        \"month\": \"<integer>\",\n        \"year\": \"<integer>\"\n      },\n      \"cv\": \"<integer>\"\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\",\n      \"amountDue\": \"<number>\"\n    },\n    \"achDetails\": {\n      \"routingNumber\": \"<string>\",\n      \"accountNumber\": \"<string>\",\n      \"checkDigit\": \"<integer>\"\n    },\n    \"creditAmount\": {\n      \"amount\": \"<number>\",\n      \"currency\": \"USD\"\n    }\n  },\n  \"tags\": [\n    \"<string>\",\n    \"<string>\"\n  ]\n}",
    "options": {

```

It seems to add examples into some of the endpoint bodies but not all of them

The reason your script adds random data to **some request bodies but not all** likely comes from how `processCollection` traverses the Postman collection and the conditions in `replaceValues`.

Key Points of the Script's Behavior

1. Only targets `request.body.raw` with valid JSON

- The function `fillRaw(rawStr, errorRate)` tries `JSON.parse(rawStr)`.
- If the body is not valid JSON (e.g., it's empty, malformed, or contains non-JSON text), the script **skips replacing placeholders**.

2. Only replaces strings that are placeholders

- `replaceValues` looks for strings that start with `<` and end with `>`, like `<string>` or `<number>`.
- If a body doesn't have placeholders (e.g., it's already example data or just a schema), no replacements occur.

3. Responses are handled separately

- The code modifies responses (`resp.body`) but again only if the response body contains placeholders.

4. Recursive traversal through `item`

- The script calls `processItem(it.item)` recursively. If the `item` array is missing or structured differently in some branches, some nodes may not be processed.

5. Objects without `request.body.raw` are ignored

- If a request uses `formdata`, `urlencoded`, or `graphql` mode instead of `raw`, the code will skip it entirely.

Why Are Some Endpoint Bodies Unchanged?

- They may not have `<string>` or `<number>` placeholders.
- They may have malformed JSON in `raw` (e.g., comments or trailing commas).
- They might be missing `request.body.raw` entirely.
- They might be defined in a different body mode (like `formdata`).

How to Fix This (Force Replacement Everywhere)

Option A: Add fallback replacement for non-JSON bodies

Modify `fillRaw` to attempt placeholder replacement even if `JSON.parse` fails. For example:

```
js

function fillRaw(rawStr, errorRate) {
  try {
    const jsonObj = JSON.parse(rawStr);
    replaceValues(jsonObj, errorRate);
    return JSON.stringify(jsonObj, null, 2);
  } catch {
    // fallback: do a regex-based placeholder replacement
    return rawStr.replace(/<string>/g, faker.word.sample())
      .replace(/<number>/g, () => (Math.random() * 1000).toFixed(2))
      .replace(/<integer>/g, () => Math.floor(Math.random() * 1000));
  }
}
```

Option B: Ensure all your `raw` bodies are valid JSON

Run a quick validator or use `jq` to verify that the JSON bodies are syntactically correct.

Option C: Expand `replaceValues` logic

If some placeholders are in arrays or nested structures that are being skipped, ensure recursion isn't missing them by logging keys during traversal:

```
js

console.log('Processing key:', key, 'value:', obj[key]);
```

Next Step

Would you like me to **update your script to guarantee placeholder replacement for all `request.body.raw` strings**, even if they are not valid JSON?



yes please

Here's an updated version of your script that ensures **placeholder replacement works for all `request.body.raw` strings**, even if the content is not valid JSON.

Key Changes

1. Fallback regex-based replacement:

If `JSON.parse` fails, the script now replaces `<string>`, `<number>`, `<integer>`, and `<boolean>` directly using regex.

2. Consistent random value generation:

Even non-JSON raw strings will get meaningful fake values.

Updated Script

```
js

#!/usr/bin/env node

/**
 * Adds random test data to "request.body.raw" fields of a Postman Collection.
 * Works for both valid JSON bodies and non-JSON raw bodies by using a regex fallback.
 */

const fs = require('fs');
const path = require('path');
const { faker } = require('@faker-js/faker');

// --- Parse CLI args ---
function parseArgs() {
  const args = process.argv.slice(2);
  const argMap = {};
  for (let i = 0; i < args.length; i++) {
    if (args[i].startsWith('--')) {
      const key = args[i].substring(2);
      const val = args[i + 1] && !args[i + 1].startsWith('--') ? args[i + 1] : true;
      argMap[key] = val;
    }
  }
  if (!argMap.input) {
    console.error('Usage: node addRandomDataToRaw.js --input <file> --output <file> [--error-rate <0-100>] [--preview]');
    process.exit(1);
  }
  argMap.errorRate = parseInt(argMap['error-rate'] || '0', 10);
  return argMap;
}

// --- Random Value Generators ---
function typeBasedRandomValue(typeHint) {
  switch (typeHint) {
    case '<number>': return parseFloat((Math.random() * 1000).toFixed(2));
    case '<integer>': return Math.floor(Math.random() * 1000);
  }
}
```

```

    case '<boolean>': return Math.random() > 0.5;
    case '<string>':
    default: return faker.word.sample();
  }
}

function randomValue(key, typeHint = '<string>') {
  if (typeHint !== '<string>') return typeBasedRandomValue(typeHint);
  const k = key.toLowerCase();
  if (k.includes('first') || k.includes('name')) return faker.person.firstName();
  if (k.includes('last')) return faker.person.lastName();
  if (k.includes('email')) return faker.internet.email();
  if (k.includes('phone')) return faker.phone.number();
  if (k.includes('address')) return faker.location.streetAddress();
  if (k.includes('city')) return faker.location.city();
  if (k.includes('state')) return faker.location.state();
  if (k.includes('zip') || k.includes('postal')) return faker.location.zipCode();
  if (k.includes('country')) return faker.location.country();
  if (k.includes('invoice')) return faker.finance.accountNumber();
  if (k.includes('amount')) return faker.finance.amount();
  if (k.includes('card')) return faker.finance.creditCardNumber();
  if (k.includes('cvv')) return faker.finance.creditCardCVV();
  if (k.includes('month')) return faker.number.int({ min: 1, max: 12 }).toString();
  if (k.includes('year')) return (new Date().getFullYear() + 1).toString();
  return faker.word.sample();
}

function erroneousValue() {
  const pool = ['!!!@@@###', '', '<script>alert(1)</script>', null, undefined, '⊗INVALID⊗'];
  return pool[Math.floor(Math.random() * pool.length)];
}

// --- JSON-aware Replacement ---
function replaceValues(obj, errorRate) {
  if (Array.isArray(obj)) {
    obj.forEach((item) => replaceValues(item, errorRate));
  } else if (typeof obj === 'object' && obj !== null) {
    Object.keys(obj).forEach((key) => {
      if (typeof obj[key] === 'string' && obj[key].startsWith('<') && obj[key].endsWith('>')) {
        const injectError = Math.random() < errorRate / 100;
        const typeHint = obj[key];
        obj[key] = injectError ? erroneousValue() : randomValue(key, typeHint);
      } else if (typeof obj[key] === 'object') {
        replaceValues(obj[key], errorRate);
      }
    });
  }
}

// --- Fallback Regex Replacement for Non-JSON ---
function replacePlaceholdersInText(text, errorRate) {
  const replacers = {
    '<string>': () => (Math.random() < errorRate / 100 ? erroneousValue() :
    faker.word.sample()),
    '<number>': () => (Math.random() < errorRate / 100 ? erroneousValue() : (Math.random() *
    1000).toFixed(2)),
    '<integer>': () => (Math.random() < errorRate / 100 ? erroneousValue() :
    Math.floor(Math.random() * 1000)),
    '<boolean>': () => (Math.random() < errorRate / 100 ? erroneousValue() : (Math.random() >
    0.5)),
  };
  return text.replace(/<string>|<number>|<integer>|<boolean>/g, (match) => replacers[match]());
}

function fillRaw(rawStr, errorRate) {
  try {
    const jsonObj = JSON.parse(rawStr);
    replaceValues(jsonObj, errorRate);
    return JSON.stringify(jsonObj, null, 2);
  } catch {
    // Fallback for non-JSON raw strings
    return replacePlaceholdersInText(rawStr, errorRate);
  }
}

```

```
// --- Traverse Collection ---
function processCollection(item, errorRate, previewMode) {
  let previewData = { before: null, after: null, captured: false };

  function processItem(it) {
    if (Array.isArray(it)) {
      it.forEach((sub) => processItem(sub));
    } else if (it && typeof it === 'object') {
      if (it.request && it.request.body && it.request.body.raw) {
        const originalRaw = it.request.body.raw;
        const newRaw = fillRaw(originalRaw, errorRate);
        it.request.body.raw = newRaw;

        if (previewMode && !previewData.captured) {
          previewData.before = originalRaw;
          previewData.after = newRaw;
          previewData.captured = true;
        }
      }
      if (it.response) {
        it.response.forEach((resp) => {
          if (resp.body) resp.body = fillRaw(resp.body, errorRate);
        });
      }
      if (it.item) processItem(it.item);
    }
  }

  processItem(item);
  return previewData;
}

// --- Main ---
(function main() {
  const { input, output, errorRate, preview } = parseArgs();
  const collection = JSON.parse(fs.readFileSync(path.resolve(input), 'utf-8'));
  const previewData = processCollection(collection.item, errorRate, preview);

  if (preview) {
    console.log('--- PREVIEW MODE ---');
    console.log('\n--- BEFORE ---\n', previewData.before);
    console.log('\n--- AFTER ---\n', previewData.after);
    console.log('\n(No files were written.)');
  } else {
    fs.writeFileSync(path.resolve(output), JSON.stringify(collection, null, 2));
    console.log(`✅ Random data added. Output saved to ${output}`);
  }
})();
```

Next Step

Would you like me to **add a --force flag** that fills in *empty* raw bodies (even if they don't have placeholders) with a **default template JSON** (e.g., { "example": "value" })? This would ensure **every** request body is populated for testing.



yes please

Here's the **enhanced script with a --force flag** that fills empty or missing raw bodies with a default JSON template.

New Behavior

- If `--force` is provided and `request.body.raw` is missing or empty, it will insert:

```
json

{
  "example": "<string>"
}
```

- The `<string>` placeholder will then be replaced with random data.

Updated Script

```
js

#!/usr/bin/env node

/**
 * Adds random test data to "request.body.raw" fields of a Postman Collection.
 * - Replaces placeholders (<string>, <number>, <integer>, <boolean>) with random values.
 * - With --force, inserts default template JSON into empty/missing raw bodies.
 * - Handles both valid JSON and non-JSON raw bodies.
 *
 * Usage:
 * node addRandomDataToRaw.js --input input.json --output output.json [--error-rate 20] [--
preview] [--force]
 */

const fs = require('fs');
const path = require('path');
const { faker } = require('@faker-js/faker');

// --- Parse CLI args ---
function parseArgs() {
  const args = process.argv.slice(2);
  const argMap = {};
  for (let i = 0; i < args.length; i++) {
    if (args[i].startsWith('--')) {
      const key = args[i].substring(2);
      const val = args[i + 1] && !args[i + 1].startsWith('--') ? args[i + 1] : true;
      argMap[key] = val;
    }
  }
  if (!argMap.input) {
    console.error('Usage: node addRandomDataToRaw.js --input <file> --output <file> [--error-
rate <0-100>] [--preview] [--force]');
    process.exit(1);
  }
  argMap.errorRate = parseInt(argMap['error-rate'] || '0', 10);
  argMap.force = !!argMap.force;
  return argMap;
}

// --- Random Value Generators ---
function typeBasedRandomValue(typeHint) {
  switch (typeHint) {
    case '<number>': return parseFloat((Math.random() * 1000).toFixed(2));
    case '<integer>': return Math.floor(Math.random() * 1000);
    case '<boolean>': return Math.random() > 0.5;
    case '<string>':
    default: return faker.word.sample();
  }
}

function randomValue(key, typeHint = '<string>') {
  if (typeHint !== '<string>') return typeBasedRandomValue(typeHint);
  const k = key.toLowerCase();
  if (k.includes('first') || k.includes('name')) return faker.person.firstName();
  if (k.includes('last')) return faker.person.lastName();
}
```

```

if (k.includes('email')) return faker.internet.email();
if (k.includes('phone')) return faker.phone.number();
if (k.includes('address')) return faker.location.streetAddress();
if (k.includes('city')) return faker.location.city();
if (k.includes('state')) return faker.location.state();
if (k.includes('zip') || k.includes('postal')) return faker.location.zipCode();
if (k.includes('country')) return faker.location.country();
if (k.includes('invoice')) return faker.finance.accountNumber();
if (k.includes('amount')) return faker.finance.amount();
if (k.includes('card')) return faker.finance.creditCardNumber();
if (k.includes('cvv')) return faker.finance.creditCardCVV();
if (k.includes('month')) return faker.number.int({ min: 1, max: 12 }).toString();
if (k.includes('year')) return (new Date().getFullYear() + 1).toString();
return faker.word.sample();
}

function erroneousValue() {
  const pool = ['!!!@@@###', '', '<script>alert(1)</script>', null, undefined, '⊗INVALID⊗'];
  return pool[Math.floor(Math.random() * pool.length)];
}

// --- JSON-aware Replacement ---
function replaceValues(obj, errorRate) {
  if (Array.isArray(obj)) {
    obj.forEach((item) => replaceValues(item, errorRate));
  } else if (typeof obj === 'object' && obj !== null) {
    Object.keys(obj).forEach((key) => {
      if (typeof obj[key] === 'string' && obj[key].startsWith('<') && obj[key].endsWith('>')) {
        const injectError = Math.random() < errorRate / 100;
        const typeHint = obj[key];
        obj[key] = injectError ? erroneousValue() : randomValue(key, typeHint);
      } else if (typeof obj[key] === 'object') {
        replaceValues(obj[key], errorRate);
      }
    });
  }
}

// --- Fallback Regex Replacement for Non-JSON ---
function replacePlaceholdersInText(text, errorRate) {
  const replacers = {
    '<string>': () => (Math.random() < errorRate / 100 ? erroneousValue() :
    faker.word.sample()),
    '<number>': () => (Math.random() < errorRate / 100 ? erroneousValue() : (Math.random() *
    1000).toFixed(2)),
    '<integer>': () => (Math.random() < errorRate / 100 ? erroneousValue() :
    Math.floor(Math.random() * 1000)),
    '<boolean>': () => (Math.random() < errorRate / 100 ? erroneousValue() : (Math.random() >
    0.5)),
  };
  return text.replace(/<string>|<number>|<integer>|<boolean>/g, (match) => replacers[match]());
}

function fillRaw(rawStr, errorRate) {
  try {
    const jsonObj = JSON.parse(rawStr);
    replaceValues(jsonObj, errorRate);
    return JSON.stringify(jsonObj, null, 2);
  } catch {
    // Fallback for non-JSON raw strings
    return replacePlaceholdersInText(rawStr, errorRate);
  }
}

// --- Traverse Collection ---
function processCollection(item, errorRate, previewMode, forceMode) {
  let previewData = { before: null, after: null, captured: false };

  function processItem(it) {
    if (Array.isArray(it)) {
      it.forEach((sub) => processItem(sub));
    } else if (it && typeof it === 'object') {
      if (it.request && it.request.body) {
        if (!it.request.body.raw && forceMode) {
          // Insert default JSON template if raw is empty/missing

```



```

    it.request.body.raw = JSON.stringify({ example: "<string>" }, null, 2);
  }

  if (it.request.body.raw) {
    const originalRaw = it.request.body.raw;
    const newRaw = fillRaw(originalRaw, errorRate);
    it.request.body.raw = newRaw;

    if (previewMode && !previewData.captured) {
      previewData.before = originalRaw;
      previewData.after = newRaw;
      previewData.captured = true;
    }
  }
}

if (it.response) {
  it.response.forEach((resp) => {
    if (resp.body) resp.body = fillRaw(resp.body, errorRate);
  });
}

if (it.item) processItem(it.item);
}
}

processItem(item);
return previewData;
}

// --- Main ---
(function main() {
  const { input, output, errorRate, preview, force } = parseArgs();
  const collection = JSON.parse(fs.readFileSync(path.resolve(input), 'utf-8'));
  const previewData = processCollection(collection.item, errorRate, preview, force);

  if (preview) {
    console.log('--- PREVIEW MODE ---');
    console.log('\n--- BEFORE ---\n', previewData.before);
    console.log('\n--- AFTER ---\n', previewData.after);
    console.log('\n(No files were written.)');
  } else {
    fs.writeFileSync(path.resolve(output), JSON.stringify(collection, null, 2));
    console.log(`✅ Random data added. Output saved to ${output}`);
  }
})();

```

How to Use

1. Default (no force):

```

bash

node addRandomDataToRaw.js --input input.json --output output.json

```

2. With --force to fill empty/missing bodies:

```

bash

node addRandomDataToRaw.js --input input.json --output output.json --force

```

3. Preview first modified request (before/after):

```
bash
```

```
node addRandomDataToRaw.js --input input.json --output output.json --preview --force
```

Next Step

Would you like me to **add a --default-template <path> option** so you can provide a custom JSON template instead of the hardcoded `{ "example": "<string>" }`?



I dont understand template

Let me explain what I meant by **"template"** in this context.

Right now, when the `--force` flag is used, the script inserts this default JSON into **any request body that is missing or empty**:

```
json
{
  "example": "<string>"
}
```

This is just a **basic placeholder structure**—a "template" for the body.

What a Custom Template Means

Imagine you want all empty request bodies to have a **specific shape**. For example:

```
json
{
  "documentSourceIdentifier": { "documentId": "<string>" },
  "recipient": { "firstName": "<string>", "lastName": "<string>" }
}
```

If we allow a `--default-template <path>` argument, you could:

1. Create a JSON file (e.g., `template.json`) with your desired structure.
2. Pass it to the script:

```
bash
```

```
node addRandomDataToRaw.js --input input.json --output output.json --force --default-template template.json
```

3. The script will use that structure **instead of** `{ "example": "<string>" }`.

Why This Is Useful

- If your endpoints always expect a certain schema, you can predefine that schema and have it filled with random data.
- It prevents the script from inserting a useless `{"example": "<string>"}` for all missing bodies.

Question for You

Would you like me to **add support for** `--default-template` so you can provide a JSON file that the script will insert whenever `--force` is used?



```
{
  "info": {
    "name": "C2M Testing Collection",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "item": [
    {
      "name": "jobs",
      "description": "",
      "item": [
        {
          "name": "submit",
          "description": "",
          "item": [
            {
              "name": "single",
              "description": "",
              "item": [
                {
                  "name": "doc",
                  "description": "",
                  "item": [
                    {
                      "id": "72426c61-38b1-418c-9aee-5ae71112e83c",
                      "name": "Submit a single document to multiple recipients",
                      "request": {
                        "name": "Submit a single document to multiple recipients",
                        "description": {}
                      },
                      "url": {
                        "path": [
                          "jobs",
                          "submit",
                          "single",
                          "doc"
                        ],
                        "host": [
                          "{{baseUrl}}"
                        ]
                      }
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

```

"query": [],
"variable": []
},
"header": [
{
"key": "Content-Type",
"value": "application/json"
},
{
"key": "Accept",
"value": "application/json"
}
],
"method": "POST",
"body": {
"mode": "raw",
"raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"yuck\"\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"Armand\",\n      \"lastName\": \"Jaren\",\n      \"address1\": \"847 Neal Courts\",\n      \"city\": \"Port Candelario\",\n      \"state\": \"Virginia\",\n      \"zip\": \"97259-0982\",\n      \"country\": \"Cote d'Ivoire\",\n      \"nickName\": \"Lonie\",\n      \"address2\": \"2270 Gottlieb Hollow\",\n      \"address3\": \"10175 Center Avenue\",\n      \"phoneNumber\": \"1-887-668-9810 x5489\"\n    },\n    {\n      \"firstName\": \"Brice\",\n      \"lastName\": \"Damon\",\n      \"address1\": \"42959 Eva Harbor\",\n      \"city\": \"Fort Cade\",\n      \"state\": \"Nevada\",\n      \"zip\": \"11974-0351\",\n      \"country\": \"Senegal\",\n      \"nickName\": \"Grayson\",\n      \"address2\": \"78184 Gleichner Knoll\",\n      \"address3\": \"5221 S Chestnut Street\",\n      \"phoneNumber\": \"271.551.1848\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"6011-7128-0643-6501\",\n      \"expirationDate\": {\n        \"month\": 608,\n        \"year\": 757\n      },\n      \"cvv\": 643\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"44406345\",\n      \"amountDue\": 368.41\n    },\n    \"achDetails\": {\n      \"routingNumber\": \"sick\",\n      \"accountNumber\": \"broken\",\n      \"checkDigit\": 947\n    },\n    \"creditAmount\": {\n      \"amount\": 810.17,\n      \"currency\": \"CAD\"\n    }\n  },\n  \"tags\": [\n    \"<string>\",\n    \"<string>\"\n  ]\n}",
"options": {
"raw": {
"headerFamily": "json",
"language": "json"
}
}
},
"auth": null
},
"response": [
{
"id": "c75de882-bad1-4bc8-b181-db96cfecd9a9",
"name": "Job submission accepted",
"originalRequest": {
"url": {
"path": [

```

```

    "jobs",
    "submit",
    "single",
    "doc"
  ],
  "host": [
    "{{baseUrl}}"
  ],
  "query": [],
  "variable": []
},
"header": [
  {
    "key": "Content-Type",
    "value": "application/json"
  },
  {
    "key": "Accept",
    "value": "application/json"
  },
  {
    "description": {
      "content": "Added as a part of security scheme: bearer",
      "type": "text/plain"
    },
    "key": "Authorization",
    "value": "Bearer <token>"
  }
],
"method": "POST",
"body": {
  "mode": "raw",
  "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\",\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    },\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"<string>\",\n      \"expirationDate\": {\n        \"month\": \"<integer>\",\n        \"year\": \"<integer>\"\n      },\n      \"cvv\": \"<integer>\"\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\",\n      \"amountDue\": \"<number>\",\n      \"achDetails\": {\n        \"routingNumber\": \"<string>\",\n        \"accountNumber\": \"<string>\",\n        \"checkDigit\": \"<integer>\"\n      },\n      \"creditAmount\": {\n        \"amount\": \"<number>\",\n        \"currency\": \"CAD\"\n      }\n    },\n    \"tags\": [\n      \"<string>\",\n      \"<string>\"\n    ]\n  },

```

```

    "options": {
      "raw": {
        "headerFamily": "json",
        "language": "json"
      }
    }
  },
  "status": "OK",
  "code": 200,
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ],
  "body": "{\n  \"jobId\": \"aw\", \n  \"status\": \"terrible\"\n}",
  "cookie": [],
  "_postman_previewlanguage": "json"
},
{
  "id": "0f0cdd33-2fe4-4be5-aa48-3510f580dce0",
  "name": "Invalid request",
  "originalRequest": {
    "url": {
      "path": [
        "jobs",
        "submit",
        "single",
        "doc"
      ],
      "host": [
        "{{baseUrl}}"
      ],
      "query": [],
      "variable": []
    },
    "header": [
      {
        "key": "Content-Type",
        "value": "application/json"
      }
    ],
    {
      "description": {
        "content": "Added as a part of security scheme: bearer",
        "type": "text/plain"
      },
      "key": "Authorization",
      "value": "Bearer <token>"
    }
  ],

```

```
"method": "POST",
"body": {
  "mode": "raw",
  "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\\\" \n\nations\": {\n    \"documentClass\": \"businessLetter\", \n    \"layout\": \"portrait\", \n    \"largeEnvelope\", \n    \"paperType\": \"letter\", \n    \"printOption\": \"color\", \n    \"flat\" \n }, \n    \"recipientAddresses\": [\n      {\n        \"firstName\": \"<string>\", \n        \"address1\": \"<string>\", \n        \"city\": \"<string>\", \n        \"state\": \"<string>\", \n        \"zip\": \"<string>\", \n        \"country\": \"<string>\", \n        \"nickName\": \"<string>\" \n      }, \n      {\n        \"firstName\": \"<string>\", \n        \"lastName\": \"<string>\", \n        \"address1\": \"<string>\", \n        \"city\": \"<string>\", \n        \"state\": \"<string>\", \n        \"zip\": \"<string>\", \n        \"country\": \"<string>\", \n        \"nickName\": \"<string>\" \n      } \n    ], \n    \"paymentDetails\": {\n      \"billingType\": \"creditCard\", \n      \"creditCardDetails\": {\n        \"cardType\": \"discover\", \n        \"cardNumber\": \"<string>\", \n        \"expirationDate\": {\n          \"month\": \"<integer>\", \n          \"year\": \"<integer>\" \n        }, \n        \"cvv\": \"<integer>\" \n      }, \n      \"invoiceDetails\": {\n        \"number\": \"<string>\", \n        \"amountDue\": \"<number>\" \n      }, \n      \"achDetails\": {\n        \"number\": \"<string>\", \n        \"accountNumber\": \"<string>\", \n        \"checkDigit\": \"<string>\" \n      }, \n      \"creditAmount\": {\n        \"amount\": \"<number>\", \n        \"currency\": \"USD\" \n      } \n    }, \n    \"tags\": [\n      \"<string>\", \n      \"<string>\" \n ] \n }\n}",
  "options": {
    "raw": {
      "headerFamily": "json",
      "language": "json"
    }
  }
},
{
  "status": "Bad Request",
  "code": 400,
  "header": [],
  "cookie": [],
  "_postman_previewlanguage": "text"
}
}
```

```

"variable": []
},
"header": [
{
"key": "Content-Type",
"value": "application/json"
},
{
"description": {
"content": "Added as a part of security scheme: bearer",
"type": "text/plain"
},
"key": "Authorization",
"value": "Bearer <token>"
}
],
"method": "POST",
"body": {
"mode": "raw",
"raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\",\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    },\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"<string>\",\n      \"expirationDate\": {\n        \"month\": \"<integer>\",\n        \"year\": \"<integer>\"\n      },\n      \"cvv\": \"<integer>\"\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\",\n      \"amountDue\": \"<number>\",\n      \"achDetails\": {\n        \"routingNumber\": \"<string>\",\n        \"accountNumber\": \"<string>\",\n        \"checkDigit\": \"<integer>\"\n      },\n      \"creditAmount\": {\n        \"amount\": \"<number>\",\n        \"currency\": \"CAD\"\n      }\n    },\n    \"tags\": [\n      \"<string>\",\n      \"<string>\"\n    ]\n  },\n  \"options\": {\n    \"raw\": {\n      \"headerFamily\": \"json\",\n      \"language\": \"json\"\n    }\n  }\n}\n"
},
"status": "Unauthorized",
"code": 401,
"header": [],
"cookie": [],
"_postman_previewlanguage": "text"

```



```

    }
  ],
  "event": [],
  "protocolProfileBehavior": {
    "disableBodyPruning": true
  }
},
{
  "name": "jobTemplate",
  "description": "",
  "item": [
    {
      "id": "da8c2514-24f3-4b99-b212-4a3eac840a7d",
      "name": "Submit a document using a job template",
      "request": {
        "name": "Submit a document using a job template",
        "description": {},
        "url": {
          "path": [
            "jobs",
            "submit",
            "single",
            "doc",
            "jobTemplate"
          ],
          "host": [
            "{{baseUrl}}"
          ],
          "query": [],
          "variable": []
        },
        "header": [
          {
            "key": "Content-Type",
            "value": "application/json"
          }
        ],
        "method": "POST",
        "body": {
          "mode": "raw",
          "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"despite\"\n  },\n  \"jobTemplate\": \"gee\",\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"Larissa\",\n      \"lastName\": \"Kaylah\",\n      \"address1\": \"3532 Magnolia Loaf\",\n      \"city\": \"New Maccester\",\n      \"state\": \"Louisiana\",\n      \"zip\": \"42677-2419\",\n      \"country\": \"Botswana\",\n      \"nickName\": \"Keenan\",\n      \"address2\": \"34256 Hillside\",\n      \"address3\": \"8530 Duane Crossing\",\n      \"phoneNumber\": \"830.526.5941\"\n    },\n    {\n      \"firstName\": \"Jarod\",\n      \"lastName\": \"Lauryn\",\n      \"address1\": \"90097 O'Kon Bridge\",\n      \"city\": \"Fort Agnesbury\",\n      \"state\": \"North Dakota\",\n      \"zip\": \"24335-5202\",\n      \"country\": \"Solomon Islands\",\n      \"nickName\": \"Emiel\",\n      \"address2\": \"4372 Caroline Knolls\",\n      \"address3\": \"271 Brent Courts\",\n      \"phoneNumber\": \"993.231.6110 x974\"\n    }\n  ],\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n
```

```

"layout": "portrait",\n "mailclass": "priorityMail",\n "paperType": "letter",\n
"printOption": "grayscale",\n "envelope": "postcard"\n },\n "paymentDetails": {\n
"billingType": "ach",\n "creditCardDetails": {\n "cardType": "mastercard",\n
"cardNumber": "3528-9221-1993-2251",\n "expirationDate": {\n "month": 521,\n
"year": 412\n },\n "cvv": 347\n },\n "invoiceDetails": {\n "invoiceNumber":
"32746236",\n "amountDue": 268.15\n },\n "achDetails": {\n "routingNumber":
"who",\n "accountNumber": "merrily",\n "checkDigit": 943\n },\n
"creditAmount": {\n "amount": 277.43,\n "currency": "USD"\n } \n },\n "tags": [\n
"<string>",\n "<string>"\n ]\n },
"options": {
"raw": {
"headerFamily": "json",
"language": "json"
}
},
"auth": null
},
"response": [
{
"id": "71812c92-f1a2-4273-8ce5-2464e99dcc38",
"name": "OK",
"originalRequest": {
"url": {
"path": [
"jobs",
"submit",
"single",
"doc",
"jobTemplate"
],
"host": [
"{{baseUrl}}"
],
"query": [],
"variable": []
},
"header": [
{
"key": "Content-Type",
"value": "application/json"
},
{
"description": {
"content": "Added as a part of security scheme: bearer",
"type": "text/plain"
},
"key": "Authorization",
"value": "Bearer <token>"
}
],

```

```

    "method": "POST",
    "body": {
      "mode": "raw",
      "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"\n    <string>\\\"\\n },\n  \"jobTemplate\": \"<string>\\\",,\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\\\",,\n      \"lastName\": \"<string>\\\",,\n      \"address1\": \"<string>\\\",,\n      \"city\": \"<string>\\\",,\n      \"state\": \"<string>\\\",,\n      \"zip\": \"<string>\\\",,\n      \"country\": \"\n      <string>\\\",,\n      \"nickName\": \"<string>\\\",,\n      \"address2\": \"<string>\\\",,\n      \"address3\": \"\n      <string>\\\",,\n      \"phoneNumber\": \"<string>\\\"\\n },\n    {\n      \"firstName\": \"<string>\\\",,\n      \"lastName\": \"<string>\\\",,\n      \"address1\": \"<string>\\\",,\n      \"city\": \"<string>\\\",,\n      \"state\": \"<string>\\\",,\n      \"zip\": \"<string>\\\",,\n      \"country\": \"<string>\\\",,\n      \"nickName\": \n      \"<string>\\\",,\n      \"address2\": \"<string>\\\",,\n      \"address3\": \"<string>\\\",,\n      \"phoneNumber\": \"<string>\\\"\\n }]\n },\n  \"jobOptions\": {\n    \"documentClass\": \n    \"businessLetter\\\",,\n    \"layout\": \"portrait\\\",,\n    \"mailclass\": \"priorityMail\\\",,\n    \"paperType\": \"letter\\\",,\n    \"printOption\": \"grayscale\\\",,\n    \"envelope\": \"postcard\\\"\\n },\n  \"paymentDetails\": {\n    \"billingType\": \"ach\\\",,\n    \"creditCardDetails\": {\n      \"cardType\": \n      \"mastercard\\\",,\n      \"cardNumber\": \"<string>\\\",,\n      \"expirationDate\": {\n        \"month\": \"\n        <integer>\\\",,\n        \"year\": \"<integer>\\\"\\n },\n      \"cvv\": \"<integer>\\\"\\n },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\\\",,\n      \"amountDue\": \"<number>\\\"\\n\n    },\n    \"achDetails\": {\n      \"routingNumber\": \"<string>\\\",,\n      \"accountNumber\": \"\n      <string>\\\",,\n      \"checkDigit\": \"<integer>\\\"\\n },\n    \"creditAmount\": {\n      \"amount\": \"\n      <number>\\\",,\n      \"currency\": \"USD\\\"\\n }]\n },\n  \"tags\": [\n    \"<string>\\\",,\n    \"<string>\\\"\\n\n  ]\n}",
      "options": {
        "raw": {
          "headerFamily": "json",
          "language": "json"
        }
      }
    },
    "status": "OK",
    "code": 200,
    "header": [],
    "cookie": [],
    "_postman_previewlanguage": "text"
  },
  {
    "id": "31dd8713-41f1-4a32-b62f-ded51036419e",
    "name": "Invalid request",
    "originalRequest": {
      "url": {
        "path": [
          "jobs",
          "submit",
          "single",
          "doc",
          "jobTemplate"
        ],
        "host": [
          "{{baseUrl}}"
        ]
      }
    }
  }
}

```

```

    ],
    "query": [],
    "variable": []
  },
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    },
    {
      "description": {
        "content": "Added as a part of security scheme: bearer",
        "type": "text/plain"
      },
      "key": "Authorization",
      "value": "Bearer <token>"
    }
  ],
  "method": "POST",
  "body": {
    "mode": "raw",
    "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"\n    <string>\\\"\\n },\n  \"jobTemplate\": \"<string>\\\"\\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\\\"\\n      \"lastName\": \"<string>\\\"\\n      \"address1\": \"<string>\\\"\\n      \"city\": \"<string>\\\"\\n      \"state\": \"<string>\\\"\\n      \"zip\": \"<string>\\\"\\n      \"country\": \"\n      <string>\\\"\\n      \"nickName\": \"<string>\\\"\\n      \"address2\": \"<string>\\\"\\n      \"address3\": \"\n      <string>\\\"\\n      \"phoneNumber\": \"<string>\\\"\\n    },\n    {\n      \"firstName\": \"<string>\\\"\\n      \"lastName\": \"<string>\\\"\\n      \"address1\": \"<string>\\\"\\n      \"city\": \"<string>\\\"\\n      \"state\": \"<string>\\\"\\n      \"zip\": \"<string>\\\"\\n      \"country\": \"<string>\\\"\\n      \"nickName\":\n      \"<string>\\\"\\n      \"address2\": \"<string>\\\"\\n      \"address3\": \"<string>\\\"\\n      \n      \"phoneNumber\": \"<string>\\\"\\n    }\n  ],\n  \"jobOptions\": {\n    \"documentClass\":\n    \"businessLetter\\\", \n    \"layout\": \"portrait\\\", \n    \"mailclass\": \"priorityMail\\\", \n    \n    \"paperType\": \"letter\\\", \n    \"printOption\": \"grayscale\\\", \n    \"envelope\": \"postcard\\\"\\n  },\n  \"paymentDetails\": {\n    \"billingType\": \"ach\\\", \n    \"creditCardDetails\": {\n      \"cardType\":\n      \"mastercard\\\", \n      \"cardNumber\": \"<string>\\\"\\n      \"expirationDate\": {\n        \"month\": \"\n        <integer>\\\", \n        \"year\": \"<integer>\\\"\\n      }, \n      \"cvv\": \"<integer>\\\"\\n    }, \n    \n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\\\"\\n      \"amountDue\": \"<number>\\\"\\n    }, \n    \n    \"achDetails\": {\n      \"routingNumber\": \"<string>\\\"\\n      \"accountNumber\": \"\n      <string>\\\"\\n      \"checkDigit\": \"<integer>\\\"\\n    }, \n    \n    \"creditAmount\": {\n      \"amount\": \"\n      <number>\\\", \n      \"currency\": \"USD\\\"\\n    }\n  }, \n  \"tags\": [\n    \"<string>\\\"\\n    \"<string>\\\"\\n  ]\n}"
  },
  "options": {
    "raw": {

```

A **template**, in this case, means a *default JSON structure* the script would insert into empty request bodies. For example:

Right now, if you run:

```
bash

node addRandomDataToRaw.js --input input.json --output output.json --force
```

and the request body is empty or missing, the script inserts:

```
json

{
  "example": "<string>"
}
```

What I Meant by a Custom Template

You could define your own JSON "shape" (e.g., `template.json`):

```
json

{
  "documentSourceIdentifier": { "documentId": "<string>" },
  "recipientAddresses": [
    { "firstName": "<string>", "lastName": "<string>" }
  ]
}
```

Then run:

```
bash

node addRandomDataToRaw.js --input input.json --output output.json --force --default-template
template.json
```

The script would insert this structure instead of `{ "example": "<string>" }` when it encounters an empty body, filling `<string>` with random data.

Would You Like Me to Add This?

I can **update your script** so you can provide any `template.json` file and it will:

- Insert your template for all empty/missing bodies.
- Automatically fill placeholders (`<string>`, `<number>`, etc.) with random data.

Would you like me to **add this --default-template feature** and show you exactly how to use it?



I just ran the updated program but it is still leaving things blank

If the updated script is still leaving some request bodies blank, there are likely a few causes:

Why It Might Still Be Blank

1. The requests don't have `request.body.raw` defined at all.

The script only checks `request.body.raw`. If the request uses `formdata`, `urlencoded`, or has no `body` object, the code won't populate it.

2. `--force` wasn't used.

Without `--force`, the script only modifies bodies that already exist and contain placeholders.

3. `--force` added `{ "example": "<string>" }`, but `<string>` was never replaced.

This could happen if the fallback `fillRaw` logic didn't parse or replace the template.

4. Nested item structures may contain objects without `request`.

If some items are folders or contain sub-items, only requests with `request.body.raw` are modified.

Quick Test

Run the script with both `--force` and `--preview`:

```
bash

node addRandomDataToRaw.js --input input.json --output output.json --force --preview
```

You should see a **BEFORE / AFTER** diff for the first modified request body.

Enhanced Debug Version

We can add **logging** to confirm what the script is seeing:

```
js

if (it.request && it.request.body) {
  console.log('Processing request:', it.name);
  console.log('Original body:', it.request.body.raw);
}
```

Place this inside `processItem()` before modification.

Next Step – Guaranteed Fill

Would you like me to:

- **Force every request to have at least** `{ "example": "randomValue" }`, even if `body` is completely missing (not just empty)?
- **Add debug logs** so you can see exactly which requests are being modified?

I can prepare a **final "guaranteed fill" version** of your script that:

1. Creates `request.body.raw` if missing.
2. Inserts the default template if `--force` is given.
3. Always replaces placeholders.

Shall I do this?



```
{
  "info": {
    "name": "C2M Testing Collection",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "item": [
    {
      "name": "jobs",
      "description": "",
      "item": [
        {
          "name": "submit",
          "description": "",
          "item": [
            {
              "name": "single",
              "description": "",
              "item": [
                {
                  "name": "doc",
                  "description": "",
                  "item": [
                    {
                      "id": "72426c61-38b1-418c-9aee-5ae71112e83c",
                      "name": "Submit a single document to multiple recipients",
                      "request": {
                        "name": "Submit a single document to multiple recipients",
                        "description": {},
                        "url": {
                          "path": [
                            "jobs",
                            "submit",
                            "single",
                            "doc"
                          ],
                          "host": [
                            "{{baseUrl}}"
                          ],
                          "query": [],
                          "variable": []
                        },
                        "header": [
                          {
                            "key": "Content-Type",
                            "value": "application/json"
                          }
                        ]
                      }
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

```

{
  "key": "Accept",
  "value": "application/json"
},
{
  "method": "POST",
  "body": {
    "mode": "raw",
    "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"yuck\"\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"Armand\",\n      \"lastName\": \"Jaren\",\n      \"address1\": \"847 Neal Courts\",\n      \"city\": \"Port Candelario\",\n      \"state\": \"Virginia\",\n      \"zip\": \"97259-0982\",\n      \"country\": \"Cote d'Ivoire\",\n      \"nickName\": \"Lonie\",\n      \"address2\": \"2270 Gottlieb Hollow\",\n      \"address3\": \"10175 Center Avenue\",\n      \"phoneNumber\": \"1-887-668-9810 x5489\"\n    },\n    {\n      \"firstName\": \"Brice\",\n      \"lastName\": \"Damon\",\n      \"address1\": \"42959 Eva Harbor\",\n      \"city\": \"Fort Cade\",\n      \"state\": \"Nevada\",\n      \"zip\": \"11974-0351\",\n      \"country\": \"Senegal\",\n      \"nickName\": \"Grayson\",\n      \"address2\": \"78184 Gleichner Knoll\",\n      \"address3\": \"5221 S Chestnut Street\",\n      \"phoneNumber\": \"271.551.1848\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"6011-7128-0643-6501\",\n      \"expirationDate\": {\n        \"month\": 608,\n        \"year\": 757\n      },\n      \"cvv\": 643\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"44406345\",\n      \"amountDue\": 368.41\n    },\n    \"achDetails\": {\n      \"routingNumber\": \"sick\",\n      \"accountNumber\": \"broken\",\n      \"checkDigit\": 947\n    },\n    \"creditAmount\": {\n      \"amount\": 810.17,\n      \"currency\": \"CAD\"\n    }\n  },\n  \"tags\": [\n    \"<string>\",\n    \"<string>\"\n  ]\n}",
    "options": {
      "raw": {
        "headerFamily": "json",
        "language": "json"
      }
    }
  },
  "auth": null
},
{
  "response": [
    {
      "id": "c75de882-bad1-4bc8-b181-db96cfecd9a9",
      "name": "Job submission accepted",
      "originalRequest": {
        "url": {
          "path": [
            "jobs",
            "submit",
            "single",
            "doc"
          ],
          "host": [
            "{{baseUrl}}"
          ]
        }
      }
    }
  ]
}

```



```

"query": [],
"variable": []
},
"header": [
  {
    "key": "Content-Type",
    "value": "application/json"
  },
  {
    "key": "Accept",
    "value": "application/json"
  },
  {
    "description": {
      "content": "Added as a part of security scheme: bearer",
      "type": "text/plain"
    },
    "key": "Authorization",
    "value": "Bearer <token>"
  }
],
"method": "POST",
"body": {
  "mode": "raw",
  "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\"\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    },\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"<string>\",\n      \"expirationDate\": {\n        \"month\": \"<integer>\",\n        \"year\": \"<integer>\"\n      },\n      \"cvv\": \"<integer>\"\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\",\n      \"amountDue\": \"<number>\",\n      \"achDetails\": {\n        \"routingNumber\": \"<string>\",\n        \"accountNumber\": \"<string>\",\n        \"checkDigit\": \"<integer>\"\n      },\n      \"creditAmount\": {\n        \"amount\": \"<number>\",\n        \"currency\": \"CAD\"\n      }\n    }\n  },\n  \"tags\": [\n    \"<string>\",\n    \"<string>\"\n  ]\n}"
  "options": {
    "raw": {
      "headerFamily": "json",
      "language": "json"
    }
  }
}
},

```

```

"status": "OK",
"code": 200,
"header": [
  {
    "key": "Content-Type",
    "value": "application/json"
  }
],
"body": "{\n  \"jobId\": \"aw\", \n  \"status\": \"terrible\"\n}",
"cookie": [],
"_postman_previewlanguage": "json"
},
{
  "id": "0f0cdd33-2fe4-4be5-aa48-3510f580dce0",
  "name": "Invalid request",
  "originalRequest": {
    "url": {
      "path": [
        "jobs",
        "submit",
        "single",
        "doc"
      ],
      "host": [
        "{{baseUrl}}"
      ],
      "query": [],
      "variable": []
    },
    "header": [
      {
        "key": "Content-Type",
        "value": "application/json"
      },
      {
        "description": {
          "content": "Added as a part of security scheme: bearer",
          "type": "text/plain"
        },
        "key": "Authorization",
        "value": "Bearer <token>"
      }
    ],
    "method": "POST",
    "body": {
      "mode": "raw",
      "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\"\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\", \n    \"layout\": \"portrait\", \n    \"mailclass\": \"largeEnvelope\", \n    \"paperType\": \"letter\", \n    \"printOption\": \"color\", \n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\", \n      \"lastName\": \"<string>\", \n      \"address1\": \"<string>\", \n      \"city\": \"<string>\", \n"

```

```

\"state\": \"<string>\",\\n  \"zip\": \"<string>\",\\n  \"country\": \"<string>\",\\n  \"nickName\": 
\"<string>\",\\n  \"address2\": \"<string>\",\\n  \"address3\": \"<string>\",\\n 
\"phoneNumber\": \"<string>\"\\n  },\\n  {\\n    \"firstName\": \"<string>\",\\n    \"lastName\": \"
<string>\",\\n    \"address1\": \"<string>\",\\n    \"city\": \"<string>\",\\n    \"state\": \"
<string>\",\\n    \"zip\": \"<string>\",\\n    \"country\": \"<string>\",\\n    \"nickName\": \"
<string>\",\\n    \"address2\": \"<string>\",\\n    \"address3\": \"<string>\",\\n 
\"phoneNumber\": \"<string>\"\\n  }\\n ],\\n  \"paymentDetails\": {\\n    \"billingType\": 
\"invoice\",\\n    \"creditCardDetails\": {\\n      \"cardType\": \"discover\",\\n      \"cardNumber\": \"
<string>\",\\n      \"expirationDate\": {\\n        \"month\": \"<integer>\",\\n        \"year\": \"
<integer>\"\\n      },\\n      \"cvv\": \"<integer>\"\\n    },\\n    \"invoiceDetails\": {\\n 
\"invoiceNumber\": \"<string>\",\\n    \"amountDue\": \"<number>\"\\n  },\\n    \"achDetails\": {\\n 
\"routingNumber\": \"<string>\",\\n    \"accountNumber\": \"<string>\",\\n    \"checkDigit\": \"
<integer>\"\\n  },\\n    \"creditAmount\": {\\n      \"amount\": \"<number>\",\\n      \"currency\": 
\"CAD\"\\n    }\\n  },\\n  \"tags\": [\\n    \"<string>\",\\n    \"<string>\"\\n  ]\\n }\", 
    \"options\": { 
      \"raw\": { 
        \"headerFamily\": \"json\", 
        \"language\": \"json\" 
      } 
    } 
  }, 
  \"status\": \"Bad Request\", 
  \"code\": 400, 
  \"header\": [], 
  \"cookie\": [], 
  \"_postman_previewlanguage\": \"text\" 
}, 
{ 
  \"id\": \"36bd10fe-c455-40b4-bef0-391a0eb8dddb\", 
  \"name\": \"Unauthorized\", 
  \"originalRequest\": { 
    \"url\": { 
      \"path\": [ 
        \"jobs\", 
        \"submit\", 
        \"single\", 
        \"doc\" 
      ], 
      \"host\": [ 
        \"{{baseUrl}}\" 
      ], 
      \"query\": [], 
      \"variable\": [] 
    }, 
    \"header\": [ 
      { 
        \"key\": \"Content-Type\", 
        \"value\": \"application/json\" 
      }, 
      { 

```

```

      "description": {
        "content": "Added as a part of security scheme: bearer",
        "type": "text/plain"
      },
      "key": "Authorization",
      "value": "Bearer <token>"
    }
  ],
  "method": "POST",
  "body": {
    "mode": "raw",
    "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\"\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"largeEnvelope\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"color\",\n    \"envelope\": \"flat\"\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    },\n    {\n      \"firstName\": \"<string>\",\n      \"lastName\": \"<string>\",\n      \"address1\": \"<string>\",\n      \"city\": \"<string>\",\n      \"state\": \"<string>\",\n      \"zip\": \"<string>\",\n      \"country\": \"<string>\",\n      \"nickName\": \"<string>\",\n      \"address2\": \"<string>\",\n      \"address3\": \"<string>\",\n      \"phoneNumber\": \"<string>\"\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\",\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\",\n      \"cardNumber\": \"<string>\",\n      \"expirationDate\": {\n        \"month\": \"<integer>\",\n        \"year\": \"<integer>\"\n      },\n      \"cvv\": \"<integer>\"\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\",\n      \"amountDue\": \"<number>\",\n      \"achDetails\": {\n        \"routingNumber\": \"<string>\",\n        \"accountNumber\": \"<string>\",\n        \"checkDigit\": \"<integer>\"\n      },\n      \"creditAmount\": {\n        \"amount\": \"<number>\",\n        \"currency\": \"CAD\"\n      }\n    }\n  },\n  \"tags\": [\n    \"<string>\",\n    \"<string>\"\n  ]\n}"
  },
  "options": {
    "raw": {
      "headerFamily": "json",
      "language": "json"
    }
  }
},
"status": "Unauthorized",
"code": 401,
"header": [],
"cookie": [],
"_postman_previewlanguage": "text"
}
],
"event": [],
"protocolProfileBehavior": {
  "disableBodyPruning": true
}
},
{

```

```

"name": "jobTemplate",
"description": "",
"item": [
  {
    "id": "da8c2514-24f3-4b99-b212-4a3eac840a7d",
    "name": "Submit a document using a job template",
    "request": {
      "name": "Submit a document using a job template",
      "description": {},
      "url": {
        "path": [
          "jobs",
          "submit",
          "single",
          "doc",
          "jobTemplate"
        ],
        "host": [
          "{{baseUrl}}"
        ],
        "query": [],
        "variable": []
      },
      "header": [
        {
          "key": "Content-Type",
          "value": "application/json"
        }
      ],
      "method": "POST",
      "body": {
        "mode": "raw",
        "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"despite\"\n  },\n  \"jobTemplate\": \"gee\",\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"Larissa\",\n      \"lastName\": \"Kaylah\",\n      \"address1\": \"3532 Magnolia Loaf\",\n      \"city\": \"New Maccester\",\n      \"state\": \"Louisiana\",\n      \"zip\": \"42677-2419\",\n      \"country\": \"Botswana\",\n      \"nickName\": \"Keenan\",\n      \"address2\": \"34256 Hillside\",\n      \"address3\": \"8530 Duane Crossing\",\n      \"phoneNumber\": \"830.526.5941\"\n    },\n    {\n      \"firstName\": \"Jarod\",\n      \"lastName\": \"Lauryn\",\n      \"address1\": \"90097 O'Kon Bridge\",\n      \"city\": \"Fort Agnesbury\",\n      \"state\": \"North Dakota\",\n      \"zip\": \"24335-5202\",\n      \"country\": \"Solomon Islands\",\n      \"nickName\": \"Emie\",\n      \"address2\": \"4372 Caroline Knolls\",\n      \"address3\": \"271 Brent Courts\",\n      \"phoneNumber\": \"993.231.6110 x974\"\n    }\n  ],\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\",\n    \"layout\": \"portrait\",\n    \"mailclass\": \"priorityMail\",\n    \"paperType\": \"letter\",\n    \"printOption\": \"grayscale\",\n    \"envelope\": \"postcard\"\n  },\n  \"paymentDetails\": {\n    \"billingType\": \"ach\",\n    \"creditCardDetails\": {\n      \"cardType\": \"mastercard\",\n      \"cardNumber\": \"3528-9221-1993-2251\",\n      \"expirationDate\": {\n        \"month\": 521,\n        \"year\": 412\n      },\n      \"cvv\": 347\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"32746236\",\n      \"amountDue\": 268.15\n    },\n    \"achDetails\": {\n      \"routingNumber\": \"who\",\n      \"accountNumber\": \"merrily\",\n      \"checkDigit\": 943\n    },\n    \"creditAmount\": {\n      \"amount\": 277.43,\n      \"currency\": \"USD\"\n    }\n  },\n  \"tags\": [\n
```

```

\"<string>\",\n  \"<string>\\\"\\n \\n}\",
    \"options\": {
      \"raw\": {
        \"headerFamily\": \"json\",
        \"language\": \"json\"
      }
    },
    \"auth\": null
  },
  \"response\": [
    {
      \"id\": \"71812c92-f1a2-4273-8ce5-2464e99dcc38\",
      \"name\": \"OK\",
      \"originalRequest\": {
        \"url\": {
          \"path\": [
            \"jobs\",
            \"submit\",
            \"single\",
            \"doc\",
            \"jobTemplate\"
          ],
          \"host\": [
            \"{{baseUrl}}\"
          ],
          \"query\": [],
          \"variable\": []
        },
        \"header\": [
          {
            \"key\": \"Content-Type\",
            \"value\": \"application/json\"
          },
          {
            \"description\": {
              \"content\": \"Added as a part of security scheme: bearer\",
              \"type\": \"text/plain\"
            },
            \"key\": \"Authorization\",
            \"value\": \"Bearer <token>\"
          }
        ],
        \"method\": \"POST\",
        \"body\": {
          \"mode\": \"raw\",
          \"raw\": \"{\\n  \\\"documentSourceIdentifier\\\": {\\n    \\\"documentId\\\": \\\"
<string>\\\"\\n  },\\n  \\\"jobTemplate\\\": \\\"<string>\\\",\\n  \\\"recipientAddresses\\\": [\\n    {\\n
\\\"firstName\\\": \\\"<string>\\\",\\n    \\\"lastName\\\": \\\"<string>\\\",\\n    \\\"address1\\\": \\\"<string>\\\",\\n
\\\"city\\\": \\\"<string>\\\",\\n    \\\"state\\\": \\\"<string>\\\",\\n    \\\"zip\\\": \\\"<string>\\\",\\n    \\\"country\\\": \\\"
<string>\\\",\\n    \\\"nickName\\\": \\\"<string>\\\",\\n    \\\"address2\\\": \\\"<string>\\\",\\n    \\\"address3\\\": \\\"

```

```
<string>\",\n  \"phoneNumber\": \"<string>\\\"\\n  },\n  {\n    \"firstName\": \"<string>\\\",\n    \"lastName\": \"<string>\\\",\n    \"address1\": \"<string>\\\",\n    \"city\": \"<string>\\\",\n    \"state\": \"<string>\\\",\n    \"zip\": \"<string>\\\",\n    \"country\": \"<string>\\\",\n    \"nickName\": \"<string>\\\",\n    \"address2\": \"<string>\\\",\n    \"address3\": \"<string>\\\",\n    \"phoneNumber\": \"<string>\\\"\\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\", \n    \"layout\": \"portrait\", \n    \"mailclass\": \"priorityMail\", \n    \"paperType\": \"letter\", \n    \"printOption\": \"grayscale\", \n    \"envelope\": \"postcard\"\\n  },\n  \"paymentDetails\": {\n    \"billingType\": \"ach\", \n    \"creditCardDetails\": {\n      \"cardType\": \"mastercard\", \n      \"cardNumber\": \"<string>\\\", \n      \"expirationDate\": {\n        \"month\": \"<integer>\\\", \n        \"year\": \"<integer>\\\"\\n      }, \n      \"cvv\": \"<integer>\\\"\\n    }, \n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\\\", \n      \"amountDue\": \"<number>\\\"\\n    }, \n    \"achDetails\": {\n      \"routingNumber\": \"<string>\\\", \n      \"accountNumber\": \"<string>\\\", \n      \"checkDigit\": \"<integer>\\\"\\n    }, \n    \"creditAmount\": {\n      \"amount\": \"<number>\\\", \n      \"currency\": \"USD\"\\n    } \n  }, \n  \"tags\": [\n    \"<string>\\\", \n    \"<string>\\\"\\n  ]\n}\",
```

```
    "options": {
      "raw": {
        "headerFamily": "json",
        "language": "json"
      }
    },
    "status": "OK",
    "code": 200,
    "header": [],
    "cookie": [],
    "_postman_previewlanguage": "text"
  },
  {
    "id": "31dd8713-41f1-4a32-b62f-ded51036419e",
    "name": "Invalid request",
    "originalRequest": {
      "url": {
        "path": [
          "jobs",
          "submit",
          "single",
          "doc",
          "jobTemplate"
        ],
        "host": [
          "{{baseUrl}}"
        ],
        "query": [],
        "variable": []
      },
      "header": [
        {
          "key": "Content-Type",
          "value": "application/json"
        }
      ]
    }
  }
}
```

```

    },
    {
      "description": {
        "content": "Added as a part of security scheme: bearer",
        "type": "text/plain"
      },
      "key": "Authorization",
      "value": "Bearer <token>"
    }
  ],
  "method": "POST",
  "body": {
    "mode": "raw",
    "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"\n    <string>\\\"\\n },\n  \"jobTemplate\": \"<string>\\\"\\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\\\"\\n      \"lastName\": \"<string>\\\"\\n      \"address1\": \"<string>\\\"\\n      \"city\": \"<string>\\\"\\n      \"state\": \"<string>\\\"\\n      \"zip\": \"<string>\\\"\\n      \"country\": \"\n      <string>\\\"\\n      \"nickName\": \"<string>\\\"\\n      \"address2\": \"<string>\\\"\\n      \"address3\": \"\n      <string>\\\"\\n      \"phoneNumber\": \"<string>\\\"\\n    },\n    {\n      \"firstName\": \"<string>\\\"\\n      \"lastName\": \"<string>\\\"\\n      \"address1\": \"<string>\\\"\\n      \"city\": \"<string>\\\"\\n      \"state\": \"<string>\\\"\\n      \"zip\": \"<string>\\\"\\n      \"country\": \"<string>\\\"\\n      \"nickName\":\n      \"<string>\\\"\\n      \"address2\": \"<string>\\\"\\n      \"address3\": \"<string>\\\"\\n      \"phoneNumber\": \"<string>\\\"\\n    }\n  ],\n  \"jobOptions\": {\n    \"documentClass\":\n    \"businessLetter\\\"\\n    \"layout\": \"portrait\\\"\\n    \"mailclass\": \"priorityMail\\\"\\n    \n    \"paperType\": \"letter\\\"\\n    \"printOption\": \"grayscale\\\"\\n    \"envelope\": \"postcard\\\"\\n  },\n  \"paymentDetails\": {\n    \"billingType\": \"ach\\\"\\n    \"creditCardDetails\": {\n      \"cardType\":\n      \"mastercard\\\"\\n      \"cardNumber\": \"<string>\\\"\\n      \"expirationDate\": {\n        \"month\": \"\n        <integer>\\\"\\n        \"year\": \"<integer>\\\"\\n      },\n      \"cv\": \"<integer>\\\"\\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\\\"\\n      \"amountDue\": \"<number>\\\"\\n    },\n    \"achDetails\": {\n      \"routingNumber\": \"<string>\\\"\\n      \"accountNumber\": \"\n      <string>\\\"\\n      \"checkDigit\": \"<integer>\\\"\\n    },\n    \"creditAmount\": {\n      \"amount\": \"\n      <number>\\\"\\n      \"currency\": \"USD\\\"\\n    }\n  },\n  \"tags\": [\n    \"<string>\\\"\\n    \"<string>\\\"\\n\n  ]\n}",
    "options": {
      "raw": {

```

It looks like the script isn't populating all the `raw` bodies because **some request.body.raw values already have data (like yuck) but no placeholders (<string>)**, so the script doesn't modify them.

Why This Happens

The script only replaces placeholders like `<string>` or `<number>`. If the value is already a string like `"yuck"`, it won't replace it.

Solution

If you want **all fields** (even ones already filled) to be replaced with random data, I can modify the script to:

1. **Overwrite all string fields** (not just `<string>` placeholders).
2. Optionally control this behavior with a `--overwrite` flag.

Would You Like This?

Should I update the script so that **all string values are replaced**, not just `<string>` placeholders?
For example:

```
json
{"documentId": "yuck"}
```

would become something like:

```
json
{"documentId": "randomWord"}
```

Would you like me to make this change?



there is no yuck in the input file

Ah, if there is **no yuck in your input file**, but you see **yuck** in the output, that means the script **already modified that field previously** and saved it with a random string (in this case, **yuck**). So the script **is working**, but it only modifies placeholders (like `<string>`) and leaves already-randomized values alone.

Why Some Fields Are Blank

If some request bodies are still blank, it could be due to:

1. `request.body.raw` is **missing completely** (not just empty).
2. **The script isn't entering those items** because they don't have a `request` object.
3. **Those requests use other modes** (like `urlencoded`, `formdata`) and not `raw`.

What We Can Do

To guarantee **all requests have random JSON bodies**, I can modify the script to:

- Add `request.body = { "mode": "raw", "raw": "{ \"example\": \"<string>\" }" }` **if it doesn't exist**.
- Replace **existing values** (even non-placeholders) with new random values, if you want.

Next Step

Would you like me to:

1. **Force-create a `body.raw` for every POST/PUT/PATCH request if missing**, and
2. **Replace every string value with a random value (not just `<string>`)?**

I can give you a **"guaranteed fill" version of the script** that will never leave a body blank.



this output has placeholders that were not replaced:

```

    "mode": "raw",
    "raw": "{\n  \"documentSourceIdentifier\": {\n    \"documentId\": \"<string>\\n\n\n  },\n  \"jobOptions\": {\n    \"documentClass\": \"businessLetter\\n\n    \"layout\": \"portrait\\n\n    \"mailclass\": \"largeEnvelope\\n\n    \"paperType\": \"letter\\n\n    \"printOption\": \"color\\n\n    \"envelope\": \"flat\\n\n  },\n  \"recipientAddresses\": [\n    {\n      \"firstName\": \"<string>\\n\n      \"lastName\": \"<string>\\n\n      \"address1\": \"<string>\\n\n      \"city\": \"<string>\\n\n      \"state\": \"<string>\\n\n      \"zip\": \"<string>\\n\n      \"country\": \"<string>\\n\n      \"nickName\": \"<string>\\n\n      \"address2\": \"<string>\\n\n      \"address3\": \"<string>\\n\n      \"phoneNumber\": \"<string>\\n\n    },\n    {\n      \"firstName\": \"<string>\\n\n      \"lastName\": \"<string>\\n\n      \"address1\": \"<string>\\n\n      \"city\": \"<string>\\n\n      \"state\": \"<string>\\n\n      \"zip\": \"<string>\\n\n      \"country\": \"<string>\\n\n      \"nickName\": \"<string>\\n\n      \"address2\": \"<string>\\n\n      \"address3\": \"<string>\\n\n      \"phoneNumber\": \"<string>\\n\n    }\n  ],\n  \"paymentDetails\": {\n    \"billingType\": \"invoice\\n\n    \"creditCardDetails\": {\n      \"cardType\": \"discover\\n\n      \"cardNumber\": \"<string>\\n\n      \"expirationDate\": {\n        \"month\": \"<integer>\\n\n        \"year\": \"<integer>\\n\n      },\n      \"cvv\": \"<integer>\\n\n    },\n    \"invoiceDetails\": {\n      \"invoiceNumber\": \"<string>\\n\n      \"amountDue\": \"<number>\\n\n    },\n    \"achDetails\": {\n      \"routingNumber\": \"<string>\\n\n      \"accountNumber\": \"<string>\\n\n      \"checkDigit\": \"<integer>\\n\n    },\n    \"creditAmount\": {\n      \"amount\": \"<number>\\n\n      \"currency\": \"CAD\\n\n    }\n  },\n  \"tags\": [\n    \"<string>\\n\n    \"<string>\\n\n  ]\n}"
    "options": {

```

If placeholders like `<string>` or `<integer>` are still present in the **output**, it means the script did not traverse or replace them correctly in that section.

Possible Reasons

1. **Nested responses or originalRequest bodies aren't being replaced.**

Your code currently modifies:

```

js
if (it.request && it.request.body && it.request.body.raw) { ... }

```

but only handles `it.response` with:

```

js
if (it.response) { ... }

```

It doesn't descend into `response.originalRequest`.

2. `fillRaw()` is failing to parse JSON.

If `request.body.raw` or `response.body` is not valid JSON (maybe it contains `<string>` as text but isn't valid JSON), the `JSON.parse()` in `fillRaw()` fails and just returns the original string unchanged.

3. There is no placeholder check for `originalRequest`.

You must explicitly call `fillRaw()` for:

```
js
response.originalRequest.body.raw
```

How to Fix

Modify `processItem()` inside your script like this:

```
js
if (it.request && it.request.body && it.request.body.raw) {
  const originalRaw = it.request.body.raw;
  const newRaw = fillRaw(originalRaw, errorRate);
  it.request.body.raw = newRaw;
}

// NEW: Process all responses including their originalRequest
if (it.response) {
  it.response.forEach((resp) => {
    if (resp.body) {
      resp.body = fillRaw(resp.body, errorRate);
    }
    if (resp.originalRequest && resp.originalRequest.body && resp.originalRequest.body.raw) {
      resp.originalRequest.body.raw = fillRaw(resp.originalRequest.body.raw, errorRate);
    }
  });
}
```

Debugging Step

Try running:

```
bash
node addRandomDataToRaw.js --input input.json --output output.json --preview
```

and check if the placeholders in `originalRequest.body.raw` now get replaced.

Next Step

Would you like me to **update your script** (full version) so that:

- It processes both `request.body.raw` and all `response.originalRequest.body.raw`?
- It logs which files were replaced (to help debug if something was skipped)?

I can give you the **fixed, ready-to-use version** of `addRandomDataToRaw.js`.



I really appreciate your help, but can you generate a Claude Code spec for this program

Here is a **Claude Code Specification (CCS)** for your `addRandomDataToRaw.js` program, capturing its purpose, structure, inputs, outputs, and behavior in a way that could be used as a formal spec for AI code review or further development.

Claude Code Spec – addRandomDataToRaw.js

Overview

This Node.js script modifies a Postman collection JSON file by inserting **randomized test data** into all `request.body.raw` fields. It replaces placeholder values like `<string>`, `<number>`, `<integer>`, and `<boolean>` with randomly generated values using the `@faker-js/faker` library.

The script can also:

- Optionally inject erroneous data with a configurable error rate (`--error-rate`).
- Force-insert default JSON bodies into requests with missing or empty bodies (`--force`).
- Preview changes without saving (`--preview`).

Purpose

To generate **dynamic test payloads** for Postman collections that mimic realistic input data for API endpoints, ensuring better coverage for QA and automated tests.

Inputs

- **Command-line arguments:**
 - `--input <file>`: Path to the input Postman collection JSON file (required).
 - `--output <file>`: Path to save the modified output JSON file.
 - `--error-rate <0-100>`: Percentage probability (0-100%) to inject erroneous values instead of random data (default: 0).
 - `--preview`: If set, the script displays the first before/after diff and does not save the output file.
 - *(Proposed extension)* `--force`: Insert a default body if no `request.body.raw` exists.
 - *(Proposed extension)* `--default-template <file>`: JSON file defining a default body template when `--force` is used.
- **Input File Format:**

A valid Postman collection JSON (v2.1 or compatible), containing `item` objects with nested `request` structures.

Outputs

- **Modified Postman Collection JSON:**

- All `request.body.raw` fields are transformed from placeholder data to realistic randomized data.
- Example transformation:

```
json

"body": {
  "mode": "raw",
  "raw": "{ \"firstName\": \"<string>\" }"
}
```

becomes:

```
json

"body": {
  "mode": "raw",
  "raw": "{ \"firstName\": \"John\" }"
}
```

- **Preview Mode Output (Console):**

Shows a before/after diff for the first modified request body.

Key Components

1. CLI Argument Parser

- **Function:** `parseArgs()`
- **Responsibility:** Parses and validates CLI arguments, ensures `--input` is provided, sets default error rate.

2. Random Data Generator

- **Function:** `randomValue(key, typeHint)`
- **Responsibility:** Generates type-specific random data using **faker**.
 - `<string>` → random name/word.
 - `<number>` → floating-point number.
 - `<integer>` → integer (0-999).
 - `<boolean>` → `true` or `false`.
 - Special key detection (e.g., `email`, `phone`, `city`) for context-aware data.

3. Error Injector

- **Function:** `erroneousValue()`

- **Responsibility:** Returns intentionally invalid or malformed data, chosen randomly from a pool, used when the `--error-rate` condition triggers.

4. Recursive Placeholder Replacer

- **Function:** `replaceValues(obj, errorRate)`
- **Responsibility:** Recursively traverses JSON objects and arrays, replacing placeholder values with random or erroneous data.

5. Raw Body Processor

- **Function:** `fillRaw(rawStr, errorRate)`
- **Responsibility:** Parses the `raw` string as JSON, runs `replaceValues()`, and returns the modified stringified JSON.

6. Collection Traversal

- **Function:** `processCollection(item, errorRate, previewMode)`
- **Responsibility:** Walks through all `item` objects in the collection, modifying:
 - `request.body.raw`
 - `response[].body`
 - *(Proposed improvement)* `response[].originalRequest.body.raw`

7. Main Entrypoint

- Reads input file.
- Runs `processCollection()`.
- Saves modified JSON to `--output` (unless `--preview` is set).

Limitations

- Currently only modifies `request.body.raw` fields in `mode: raw` (JSON) format.
- Does not process `formdata`, `urlencoded`, or `graphql` modes.
- Placeholders outside `<string>`, `<number>`, `<integer>`, `<boolean>` are ignored.
- **Some placeholders may remain if:**
 - The raw body is invalid JSON (fails `JSON.parse`).
 - Nested placeholders in `response.originalRequest.body.raw` aren't explicitly processed.

Proposed Enhancements

- **`--force` flag:** Force-add a default body to all POST/PUT/PATCH requests missing a body.
- **`--default-template <file>`:** Use a custom JSON template to populate missing request bodies.

- **Process** `response.originalRequest` and **all nested bodies** to ensure no placeholders remain.
- **Logging mode:** Show which items were updated (e.g., request name and count).

Example Usage

```
bash

node addRandomDataToRaw.js \
  --input c2m.collection.json \
  --output c2m.collection.randomized.json \
  --error-rate 10 \
  --preview
```

Would you like me to **convert this Claude Code Spec into a README.md** for your script (with examples and usage)?



yes please - also what is the reason for mentioning limitations?

Mentioning **limitations** in a spec or README is important because:

1. **Clarifies the scope** – It sets expectations for what the script can and cannot handle (e.g., only `request.body.raw` is supported, not `formdata`).
2. **Helps developers avoid confusion** – Without knowing these limits, someone might think the script automatically replaces every placeholder in all request types.
3. **Guides future improvements** – Listing current limitations provides a roadmap for features to add later.
4. **Prevents misuse** – If a user tries to run this script on unsupported body modes (like `formdata`) and doesn't see changes, they'll know why.

README.md for addRandomDataToRaw.js

```
markdown

# addRandomDataToRaw.js

This Node.js script modifies a Postman collection JSON by inserting **randomized test data** into all `request.body.raw` fields. It replaces placeholders such as `<string>`, `<number>`, `<integer>`, and `<boolean>` with realistic random values using [@faker-js/faker] (https://github.com/faker-js/faker).
```

```
## Features
- Replaces placeholders (`<string>`, `<number>`, `<integer>`, `<boolean>`) with random values.
- Injects **erroneous data** at a configurable rate using `--error-rate`.
- Supports **preview mode** (`--preview`) to show a before/after diff of the first modified body.
- Optionally **forces default bodies** for missing or empty `request.body.raw` (future enhancement).

---
```

Usage

```
```bash
node addRandomDataToRaw.js --input input.json --output output.json [--error-rate 20] [--preview]
```

## Parameters

- **--input <file>**  
Path to the Postman collection JSON file to modify. **(Required)**
- **--output <file>**  
Path to save the modified collection. **(Required)**
- **--error-rate <0-100>**  
Percentage chance to insert **erroneous values** instead of random valid values. Defaults to **0**.
- **--preview**  
Displays the before/after difference for the first modified body instead of writing the output file.

## Example

### Input (raw body)

```
json

{
 "firstName": "<string>",
 "zip": "<string>",
 "amountDue": "<number>"
}
```

### Output (after running)

```
json

{
 "firstName": "John",
 "zip": "90210",
 "amountDue": 123.45
}
```

## How It Works

1. Reads the input Postman collection JSON.
2. Recursively traverses all **item** objects.
3. For each **request.body.raw**, it:
  - Parses the JSON.
  - Replaces placeholders using **faker** functions based on key names.
  - Optionally injects invalid data (⊗INVALID⊗) if **--error-rate** triggers.



4. Writes the updated collection to the specified output file.

## Limitations

- Only modifies `request.body.raw` fields using JSON mode.
- Does not modify `formdata`, `urlencoded`, or `graphql` request bodies.
- Placeholders outside `<string>`, `<number>`, `<integer>`, `<boolean>` are ignored.
- Raw bodies that are **not valid JSON** are skipped (left unchanged).
- `response.originalRequest.body.raw` placeholders may remain unless explicitly processed (future enhancement).

## Future Enhancements

- `--force`: Add default JSON bodies for all POST/PUT/PATCH requests missing a body.
- `--default-template <file>`: Allow a user-defined JSON template to populate empty bodies.
- **Extended support** for `formdata` and `urlencoded`.
- **Verbose logging** to show which requests were modified.

## License

MIT

```
yaml

Next Step
Would you like me to **update your script now** to handle:
- `response.originalRequest.body.raw` placeholders?
- Add a `--force` flag that inserts a default JSON body where `body.raw` is missing?
```