

Output API

The following document highlights the details of the Output API.

API Description

Objective

The Output API provides a summary of the application . It is applicable for both SDK integration and onboard links .

Input	Output
The transaction ID	The summary of the user journey(or application)

API URL

```
https://ind.idv.hyperverge.co/v1/output
```

API Endpoint

output

Overview

The Output API is RESTful and uses standard HTTP verbs and status codes. The responses are in JSON format and you should send all data in JSON format through a POST request.

Method - POST

Authentication

You need a unique pair of application ID (appld) and application key (appKey) from HyperVerge to verify your identity for accessing the API.

IP WHITELISTING

For security purposes, the Output API requires IP whitelisting. You can whitelist your IP addresses through the HyperVerge Dashboard . This is a one-time setup that needs to be done during your initial onboarding. Once whitelisted, your IP will have permanent access to the API.

Only users with admin access can perform IP whitelisting. Due to security restrictions, only adding new IPs is supported - editing or deleting existing entries is not allowed.

For additional support or queries, please contact our support team at support@hyperverge.co .

Headers

Header	Mandatory / Optional	Description	Input Format
content-type	Mandatory	This parameter defines the media type for the request payload	application/json
appId	Mandatory	The application identifier shared by HyperVerge. You can find the details in the dashboard's credentials tab	This should be a unique value
appKey	Mandatory	The application key shared by HyperVerge. You can find the details in the dashboard's credentials tab	This should be a unique value

Inputs

The following table provides the details of the parameters required for the Output API's request body:

Parameter	Mandatory / Optional	Type	Description	Input Format	Default Value
transactionId	Mandatory	string	A unique identifier for tracking a user journey. For more details, refer the transaction ID.	This should be both unique and easily associated with the specific user's interaction in your application(s)	Not Applicable
workflowId	Optional	string	Name/ID of the workflow configured in the HyperVerge dashboard	Not Applicable	Not Applicable
sendReviewDetails	Optional	string	Returns the reviewer email in the response who manually approved or declined the application from the dashboard	"yes" or "no"	Not Applicable
sendDebugInfo	Optional	string	<ul style="list-style-type: none">Returns the last module ID (latestModule) from where either the user dropped from the flow or an error occurred. The debugInfo object (containing the latestModule key) is returned after enabling the sendDebugInfo parameter.The response returns a	"yes" or "no"	"no"

Parameter	Mandatory / Optional	Type	Description	Input Format	Default Value
			moduleId when the status is "user_cancelled" or "error". • For any other case statuses such as "auto_approve", "auto_declined" and "needs_review", the response skips the debugInfo object (and the latestModule key in it)		
sendAllDebugInfo	Optional	string	<ul style="list-style-type: none"> • Returns detailed debug information and complete error context (such as "Device is rooted" messages) that may be missing in the standard debugInfo response. • The standard debugInfo returned during specific states like "error" or "user_cancelled" does not include the full debug details • This parameter addresses the missing error details by returning the full error context when explicitly requested 	"yes" or "no"	"no"

Request

The following code snippet demonstrates a standard curl request for the Output API:

```
curl --location --request POST 'https://ind.idv.hyperverge.co/v1/output' \
--header 'Content-Type: application/json' \
--header 'appId: <Enter_the_HyperVerge_app_ID>' \
--header 'appKey: <Enter_the_HyperVerge_app_Key>' \
--data '{
  "transactionId": "<Enter_the_Transaction_ID>",
  "workflowId": "<Enter_the_Workflow_ID>"
}'
```

Test this API

Success Responses

The following code snippet demonstrates a success response from the Output API:

- Successful Execution
- Auto Declined
- User Cancelled

```
{
  "status": "success",
  "statusCode": 200,
  "metadata": {
    "requestId": "<Request_Identifier>"
  },
  "result": {
    "flags": [
      {
        "source": "module_2",
        "subSource": "front",
        "values": {
          "422": "Document not detected",
          "312": "Not clear"
        }
      }
    ],
    {
      "source": "module_3",
      "values": {
        "200": "Success"
      }
    }
  ],
  "userDetails": {
    "name": "dev",
    "croppedImageUrl": "<Image_URL>"
  },
  "status": "auto_approved",
  "transactionId": "k8s-test",
  "failureReason": {
    "text": "Aadhaar number match failed",
    "id": "condition_3_needs_review"
  }
}
```

A response when the status is "auto_declined" with the sendReviewDetails and sendDebugInfo parameters enabled.

```
{
  "status": "success",
  "statusCode": 200,
  "metadata": {
    "requestId": "<Request_ID>"
  },
  "result": {
    "flags": [
      {
        "source": "module_9fXhGf_id_card_validation",
        "values": {
          "130": "Document captured from screen"
        },
        "subSource": "front"
      }
    ],
    "userDetails": {},
    "status": "auto_declined",
  }
```

```

"transactionId": "<Transaction_ID>",
"reviewDetails": {
  "reviewer": ""
}
}
}

```

A response when the status is "user_cancelled" with the sendReviewDetails and sendDebugInfo parameters enabled.

```

{
  "status": "success",
  "statusCode": 200,
  "metadata": {
    "requestId": "<Request_ID>"
  },
  "result": {
    "flags": [],
    "userDetails": {},
    "status": "user_cancelled",
    "transactionId": "<Transaction_ID>",
    "reviewDetails": {
      "reviewer": ""
    },
    "debugInfo": {
      "latestModule": "module_videoKYC"
    }
  }
}

```

- Error
- Error (sendDebugInfo only)
- Error (sendDebugInfo & sendAllDebugInfo)

A response when the status is "error" with the sendAllDebugInfo parameter set to "yes". This returns the full debug information including latestModule , latestCondition , errorCode , and errorMessage .

```

{
  "status": "success",
  "statusCode": 200,
  "metadata": {
    "requestId": "abc-123",
    "transactionId": "<Transaction_ID>"
  },
  "result": {
    "flags": {},
    "userDetails": {},
    "status": "error",
    "transactionId": "<Transaction_ID>",
    "failureReason": null,
    "debugInfo": {
      "latestModule": "liveness",
      "latestCondition": "device_check_failed",
      "errorCode": 114,
      "errorMessage": "Device is rooted"
    }
  }
}

```

A response when the status is "error" with sendDebugInfo set to "yes" and sendAllDebugInfo set to "no" (or not provided). This returns only the latestModule in the debugInfo object.

```
{
  "status": "success",
  "statusCode": 200,
  "metadata": {
    "requestId": "<Request_ID>",
    "transactionId": "<Transaction_ID>"
  },
  "result": {
    "flags": {},
    "userDetails": {},
    "status": "error",
    "transactionId": "<Transaction_ID>",
    "failureReason": null,
    "debugInfo": {
      "latestModule": "liveness"
    }
  }
}
```

A response when the status is "error" with both sendDebugInfo and sendAllDebugInfo set to "yes". This returns the full debug information.

```
{
  "status": "success",
  "statusCode": 200,
  "metadata": {
    "requestId": "<Request_ID>",
    "transactionId": ""
  },
  "result": {
    "flags": {},
    "userDetails": {},
    "status": "error",
    "transactionId": "<Transaction_ID>",
    "failureReason": null,
    "debugInfo": {
      "latestModule": "liveness",
      "latestCondition": "device_check_failed",
      "errorCode": 114,
      "errorMessage": "Device is rooted"
    }
  }
}
```

Success Response Details

The following table outlines the details of the success response from the Output API:

Parameter	Type	Description
status	string	The status of the API request
statusCode	integer	The HTTP status code of the response
metadata	object	Metadata information about the request

Parameter	Type	Description
metadata.requestId	string	A unique identifier for the request
metadata.transactionId	string	The transaction ID associated with the request (optional, appears in error responses)
result	object	The result object containing the application summary
result.flags	array	An array of flag objects that contain messages explaining why the application was auto-declined or tagged for review
result.flags[].source	string	The source module identifier that generated the flag
result.flags[].subSource	string	The sub-source identifier, typically indicating the document side (e.g., "front")
result.flags[].values	object	An object containing key-value pairs where keys are error codes and values are descriptive messages
result.userDetails	object	User details extracted from the verification process
result.userDetails.name	string	The name of the user extracted from the documents
result.userDetails.croppedImageUrl	string	The URL of the cropped image of the user
result.status	string	The status of the application (e.g., "auto_approved", "auto_declined", "needs_review", "user_cancelled", "error")
result.transactionId	string	The unique transaction identifier for tracking the user journey
result.failureReason	object	An object containing details about why the application failed or needs review (null if not applicable)
result.failureReason.text	string	A descriptive text explaining the failure reason
result.failureReason.id	string	A unique identifier for the failure condition
result.reviewDetails	object	Review details returned when sendReviewDetails parameter is enabled (optional)
result.reviewDetailsReviewer	string	The email address of the reviewer who manually approved or declined the application from the dashboard
result.debugInfo	object	Debug information returned when sendDebugInfo or sendAllDebugInfo parameters are enabled (optional)
result.debugInfo.latestModule	string	The last module ID from where either the user dropped from the flow or an error occurred
result.debugInfo.latestCondition	string	The latest condition that was evaluated (returned when sendAllDebugInfo is enabled)
result.debugInfo.errorCode	integer	The error code associated with the error (returned when sendAllDebugInfo is enabled)
result.debugInfo.errorMessage	string	A detailed error message explaining the error (returned when sendAllDebugInfo is enabled)

NOTE

The messages displayed under the flag fields are configurable. Please check with the HyperVerge team for details relevant to your workflow.

Error Responses

The following are some error responses from the Output API:

- Transaction ID Not Found
- Authentication Error
- 5xx Errors
- Invalid sendAllDebugInfo Value

The Output API returns the following response when the user's verification input is incomplete or hasn't finished processing after receiving the initial input. It also returns this response when the request has an invalid transaction ID.

```
{
  "status": "failure",
  "statusCode": 400,
  "metadata": {
    "requestId": "<Request_ID>"
  },
  "result": {
    "error": "TransactionId not found"
  }
}
```

```
{
  "status": "failure",
  "statusCode": 401,
  "metadata": {
    "requestId": "<Request_ID>"
  },
  "result": {
    "error": "Missing/Invalid credentials"
  }
}
```

```
{
  "statusCode": 500,
  "status": "failure",
  "result": {
    "error": "Internal Server Error"
  },
  "metadata": {
    "requestId": "<Request_ID>"
  }
}
```


The Output API returns the following response when an invalid value is provided for the `sendAllDebugInfo` parameter.

```
{
  "status": "failure",
  "statusCode": 422,
  "metadata": {
    "requestId": "<Request_ID>",
    "transactionId": null
  },
  "result": {
    "error": "\"sendAllDebugInfo\" must be one of [yes, no]"
  }
}
```

Error Response Details

A failure or error response contains a failure status with a relevant status code and error message. The following table lists all error responses:

Status Code	Error Message	Error Description	Error Resolution
400	TransactionId not found	The request has an invalid transaction ID, or the user's verification input is incomplete or hasn't finished processing after receiving the initial input	Provide a valid transaction ID in the request
401	Missing/Invalid credentials	The request is either missing the mandatory <code>appld</code> and <code>appKey</code> combination or has invalid values	Provide valid <code>appld</code> and <code>appKey</code> credentials in the request
422	"sendAllDebugInfo" must be one of [yes, no]	The request has an invalid value for the <code>sendAllDebugInfo</code> parameter that is not one of the allowed enum values	Provide valid <code>sendAllDebugInfo</code> parameter with value "yes" or "no" in the request
429	Rate limit Error	The request has exceeded the rate limit for API calls	Please contact the HyperVerge team
500	Internal Server Error	The request has an internal server error. Please check the request headers or contact the HyperVerge team for investigation	Please check the request headers or contact the HyperVerge team for resolution
503	Server Busy	The service is currently unavailable. Please try again later or contact the HyperVerge team for investigation	Please try again later or contact the HyperVerge team for resolution