

Improve your API with OpenAPI

Rob Allen

Longhorn PHP, November 2022

APIs Power the Internet



APIs Power the Internet

API Descriptions Power APIs



The OpenAPI Specification (OAS) defines a standard, programming language-agnostic interface description for HTTP APIs, which allows both humans and computers to discover and understand the capabilities of a service

<https://spec.openapis.org/oas/latest.html>



It's about
documentation



It's about
design-first



It's about
communicating changes



It's about
development workflows



It's about
standardisation



It's about
a contract



"Using a consistent API description will help increase adoption of APIs across government by reducing time spent in understanding different APIs."

gov.uk



Anatomy of the specification

openapi.yaml

```
openapi: "3.1.0"  # or "3.0.3"
info: # ...
servers: # ...
paths: # ...
webhooks: # ...
components: # ...
security: # ...
tags: # ...
externalDocs: # ...
```

Metadata

```
info:  
  title: Rock-Paper-Scissors  
  version: "1.0.0"  
  description: An implementation of Rock-Paper-Scissors.  
  contact:  
    name: "Rob Allen"  
  license:  
    name: The MIT License
```

```
servers:  
  - url: https://rock-paper-scissors.example.com  
    description: "RPS production API"
```



Endpoints

```
paths:
  /games:
    get:
      # ...
    post:
      # ...
  '/games/{game_id}/moves':
    post:
      # ...
  '/games/{game_id}/judgement':
    get:
      # ...
```



Endpoints

```
paths:
  /games:
    post:
      operationId: createGame
      summary: Create a new game
      description: Create a new game of Rock-Paper-Scissors.
      requestBody:
        # ...
      responses:
        # ...
```



Endpoints

paths:

`/games`:

post:

operationId: createGame

summary: Create a new game

description: Create a new game of Rock-Paper-Scissors.

requestBody:

...

responses:

...



Endpoints

```
paths:
  /games:
    post:
      operationId: createGame
      summary: Create a new game
      description: Create a new game of Rock-Paper-Scissors.
      requestBody:
        # ...
      responses:
        # ...
```



Endpoints

```
paths:  
  /games:  
    post:  
      operationId: createGame  
      summary: Create a new game  
      description: Create a new game of Rock-Paper-Scissors.  
      requestBody:  
        # ...  
      responses:  
        # ...
```



RequestBody

```
requestBody:  
  description: Game to add  
  required: true  
  content:  
    application/json:  
      schema:  
        $ref: '#/components/schemas/NewGameRequest'
```


RequestBody

```
requestBody:  
  description: Game to add  
  required: true  
  content:  
    application/json:  
      schema:  
        $ref: '#/components/schemas/NewGameRequest'
```



RequestBody

```
requestBody:  
  description: Game to add  
  required: true  
  content:  
    application/json:  
      schema:  
        $ref: '#/components/schemas/NewGameRequest'
```



Reuse of objects

\$ref allows us to define once & use in many places

components:

 schemas:

 GameId:

 type: string

 format: "uuid"

 examples:

 – "2BC08389-885A-4322-80D0-EF0DE2D7CD37"

 Player:

 type: string

 example: "Lucy"



Reuse of objects

\$ref allows us to define once & use in many places

components:

 schemas:

 GameId:

 type: string

 format: "uuid"

 examples:

 - "2BC08389-885A-4322-80D0-EF0DE2D7CD37"

 Player:

 type: string

 example: "Lucy"



Reuse of objects

\$ref allows us to define once & use in many places

components:

 schemas:

 GameId:

 type: string

 format: "uuid"

 examples:

 – "2BC08389-885A-4322-80D0-EF0DE2D7CD37"

 Player:

 type: string

 example: "Lucy"



Reuse of objects

\$ref allows us to define once & use in many places

components:

 schemas:

 GameId:

 type: string

 format: "uuid"

 examples:

 - "2BC08389-885A-4322-80D0-EF0DE2D7CD37"

 Player:

 type: string

 example: "Lucy"



Reuse of objects

\$ref allows us to define once & use in many places

components:

 schemas:

 GameId:

 type: string

 format: "uuid"

 examples:

 - "2BC08389-885A-4322-80D0-EF0DE2D7CD37"

 Player:

 type: string

 example: "Lucy"



Reuse of objects

\$ref allows us to define once & use in many places

components:

 schemas:

 GameId:

 type: string

 format: "uuid"

 examples:

 – "2BC08389-885A-4322-80D0-EF0DE2D7CD37"

 Player:

 type: string

 example: "Lucy"



Build on top of other components

```
schemas:  
  NewGameRequest:  
    properties:  
      player1:  
        $ref: '#/components/schemas/Player'  
      player2:  
        $ref: '#/components/schemas/Player'  
    required:  
      - player1  
      - player2  
    examples:  
      - '{"player1":"Lucy", "player2":"Dave"}'
```

Build on top of other components

schemas:

NewGameRequest:

properties:

player1:

\$ref: '#/components/schemas/Player'

player2:

\$ref: '#/components/schemas/Player'

required:

- player1
- player2

examples:

- '{"player1": "Lucy", "player2": "Dave"}'



Build on top of other components

```
schemas:  
  NewGameRequest:  
    properties:  
      player1:  
        $ref: '#/components/schemas/Player'  
      player2:  
        $ref: '#/components/schemas/Player'  
    required:  
      - player1  
      - player2  
    examples:  
      - '{"player1": "Lucy", "player2": "Dave"}'
```



Build on top of other components

```
schemas:  
  NewGameRequest:  
    properties:  
      player1:  
        $ref: '#/components/schemas/Player'  
      player2:  
        $ref: '#/components/schemas/Player'  
  required:  
    - player1  
    - player2  
  examples:  
    - '{"player1":"Lucy", "player2":"Dave"}'
```



Build on top of other components

```
schemas:  
  NewGameRequest:  
    properties:  
      player1:  
        $ref: '#/components/schemas/Player'  
      player2:  
        $ref: '#/components/schemas/Player'  
    required:  
      - player1  
      - player2  
examples:  
  - '{"player1":"Lucy", "player2":"Dave"}'
```



RequestBody

```
requestBody:  
  description: Game to add  
  required: true  
  content:  
    application/json:  
      schema:  
        $ref: '#/components/schemas/NewGameRequest'
```



Responses

```
responses:
```

```
  '201':
```

```
    $ref: '#/components/responses/NewGameResponse'
```

```
  '400':
```

```
    $ref: '#/components/responses/NewGameError'
```

```
  '500':
```

```
    $ref: '#/components/responses/InternalServerError'
```



Responses

```
responses:
```

```
  '201':
```

```
    $ref: '#/components/responses/NewGameResponse'
```

```
  '400':
```

```
    $ref: '#/components/responses/NewGameError'
```

```
  '500':
```

```
    $ref: '#/components/responses/InternalServerError'
```



Responses

```
responses:
```

```
  '201':
```

```
    $ref: '#/components/responses/NewGameResponse'
```

```
  '400':
```

```
    $ref: '#/components/responses/NewGameError'
```

```
  '500':
```

```
    $ref: '#/components/responses/InternalServerError'
```

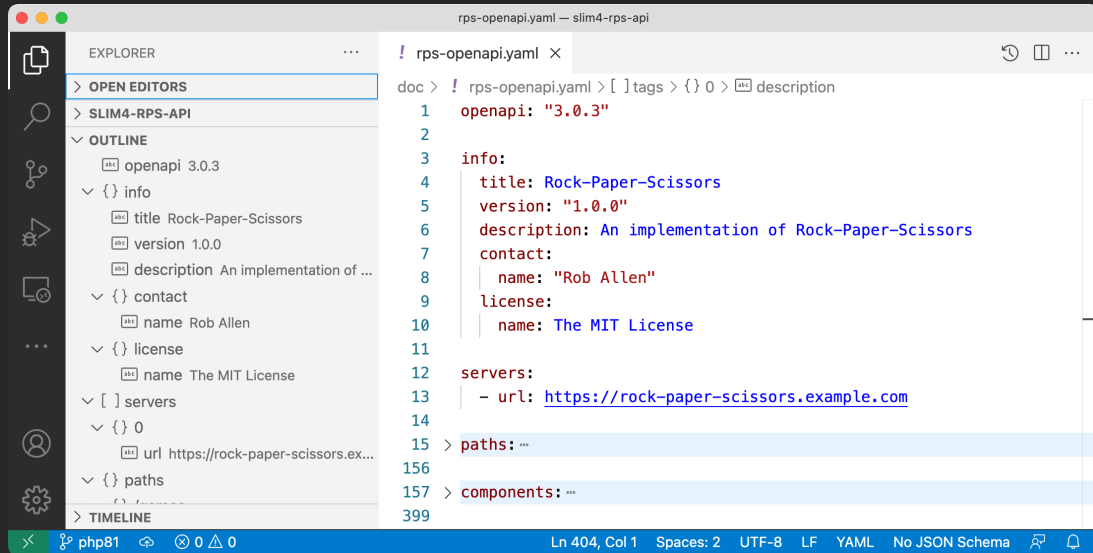




Writing your spec

Editing

It's just text!

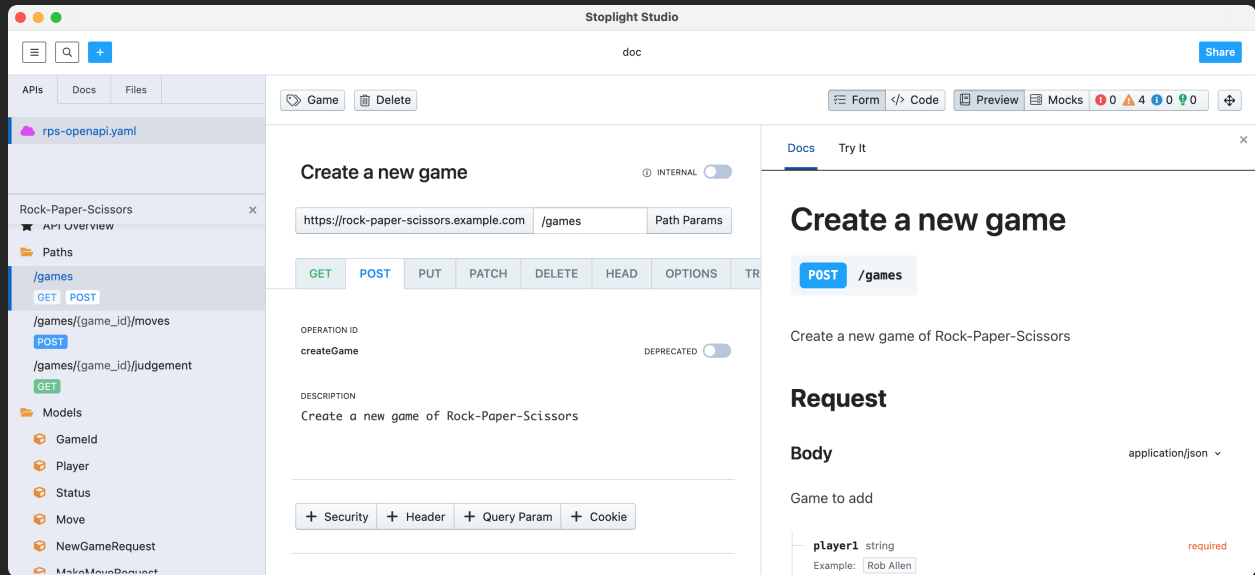


The screenshot shows a code editor window titled "rps-openapi.yaml — slim4-rps-api". The left sidebar contains an "EXPLORER" panel with a tree view of the file structure. The main editor area displays the content of "rps-openapi.yaml" with line numbers. The status bar at the bottom indicates the cursor is at line 404, column 1, with 2 spaces, UTF-8 encoding, LF line endings, and no JSON schema.

```
doc > ! rps-openapi.yaml > [ ] tags > { } 0 > description
1  openapi: "3.0.3"
2
3  info:
4    title: Rock-Paper-Scissors
5    version: "1.0.0"
6    description: An implementation of Rock-Paper-Scissors
7    contact:
8      name: "Rob Allen"
9    license:
10     name: The MIT License
11
12  servers:
13    - url: https://rock-paper-scissors.example.com
14
15  > paths: ...
156
157  > components: ...
399
```

Editing

GUI tools: Stoplight, OpenAPI-GUI, Swagger Editor



Linting & validation

CLI tools: Spectral, openapi-spec-validator, etc.

```
$ spectral lint rps-openapi.yaml
```

No results with a severity of 'error' or higher found!

Validation error

```
$ spectral lint rps-openapi.yaml
```

```
.../slim4-rps-api/doc/rps-openapi.yaml
```

```
3:6 warning info-contact Info object must have  
    "contact" object. info
```

```
× 1 problem (0 errors, 1 warning, 0 infos, 0 hints)
```



Docs

Docs

The screenshot displays a REST client interface with a sidebar on the left, a main documentation area, and a right-hand panel for request and response samples.

Left Sidebar:

- Search bar: Search...
- Game (dropdown)
- GET List all games
- POST Create a new game
- POST Make a move
- GET Judge the game
- Documentation Powered by ReDoc

Main Documentation Area:

Create a new game

Create a new game of Rock-Paper-Scissors

REQUEST BODY SCHEMA: application/json

Game to add

player1	string (Player)
required	
player2	string (Player)
required	

Responses

- > 201 The game was successfully created.
The link required to play the game by making the first move is provided in the `_links` property of the returned payload.
- > 400 Unable to create game due to a problem with the provided information
- > 404 A Not Found error occurred
- > 405 A Method Not Allowed error occurred
- > 500 An internal server error occurred

Right Panel:

POST /games

Request samples

Payload

Content type: application/json

Copy Expand all Collapse all

```
{  "player1": "Lucy",  "player2": "Dave"}
```

Response samples


201 400 404 405 500

Content type: application/hal+json

Copy Expand all Collapse all

```
{  "game_id": "2BC08389-885A-4322-80D0-EF0DE2D7CD",  "_links": {    - "makeNextMove": {      "href": "/games/2BC08389-885A-4322-80D0-      "description": "Make a player's move"    }  }}
```


Docs



< ALL DOCS

API

- Overview
- Libraries
- API versioning
- Postman Collection
- Product endpoints
- Item endpoints**
- Institution endpoints
- Account endpoints and schemas
- Token endpoints
- Processor endpoints
- Sandbox endpoints
- Webhooks

Search

API Version: 2020-09-14

Plaid.com

Get API keys

/item/get

Retrieve an Item

Returns information about the status of an Item.

Request fields and example

client_id	string
Your Plaid API client_id. The client_id is required and may be provided either in the PLAID-CLIENT-ID header or as part of a request body.	
secret	string
Your Plaid API secret. The secret is required and may be provided either in the PLAID-SECRET header or as part of a request body.	
access_token	required , string
The access token associated with the Item data is being requested for.	

Response fields and example

Collapse all

item	object
Metadata about the Item.	

Current libraries

Legacy libraries

```
/item/get

1 const request: ItemGetRequest = {
2   access_token: accessToken,
3 };
4 try {
5   const response = await plaidClient.itemGet(request);
6   const item = response.data.item;
7   const status = response.data.status;
8 } catch (error) {
9   // handle error
10 }
```

API Object

```
1 {
```

Rob Allen ~ @akrabad

Docs

The screenshot shows the GitHub repository page for `plaid / plaid-openapi`. The repository is in the `master` branch, has 1.20.6+ versions, 19 branches, and 10 tags. It was last updated 5 days ago with 62 commits. The repository contains a `.github/workflows` directory, a `2020-09-14.yml` file, a `CHANGELOG.md` file, and a `README.md` file. The `README.md` file is selected and shows the title `plaid-OpenAPI`. The content of the `README.md` file is as follows:

plaid-OpenAPI

Plaid uses the `OpenAPI 3.0.0` specification to schematize our [docs](#) and to generate our supported client libraries. This provides for a consistent typing experience across our external interfaces. Below we have listed some examples and issues we have found when iterating on the specification.

Using the OpenAPI generator

You can find examples on the official [OpenApiGenerator docs](#).

On the right side of the repository page, there is a sidebar with the following information:

- API version 2020-09-14
- [plaid.com/docs](#)
- 39 stars
- 35 watching
- 18 forks
- 1 year old
- 1.8.1-beta: Merge pull request ... on 10 Jun 2021
- Contributors 8

Docs

The screenshot shows a web browser displaying the GitHub repository for `Nexmo/api-specification`. The repository has 27 stars and 12 issues. The file `application.v2.yml` is selected, showing its content and commit history. The commit history indicates that `francesco-fipertani-vonage` added the version to messages capability in commit `#450` on November 9, 2021. The file content is a YAML configuration for an OpenAPI specification.

Search or jump to... Pull requests Issues Marketplace Explore

Nexmo / api-specification ✓ Watch Fork Star 27

< Code Issues 12 Pull requests 8 Actions 5 Releases 193

main number-pools-2.0.2 api-specification / definitions / application.v2.yml

francesco-fipertani-vonage added version to messages capability (#450) Latest commit 70c61ff on 9 Nov 2021 History

8 contributors

838 lines (826 sloc) 39.6 KB Raw Blame

```
1 ---
2 openapi: "3.0.0"
3 info:
4   version: 2.1.1
5   title: "Application API"
6   description: |
7     Vonage provides an Application API to allow management of your Vonage Applications.
8
9   This API is backwards compatible with version 1. Applications created using version 1 of the API can also be managed using version 2 (this version) of the API.
10  contact:
11    name: Vonage
12    url: "https://developer.nexmo.com/"
13    email: devrel@nexmo.com
14  servers:
15    - url: https://api.nexmo.com/v2/applications
16  security:
17    - basicAuth: []
18  paths:
19    - /:
```

Docs

Available Operations

Errors

Application API

Vonage provides an Application API to allow management of your Vonage Applications.

This API is backwards compatible with version 1. Applications created using version 1 of the API can also be managed using version 2 (this version) of the API.

Download OAS 3 Definition

Improve this specification

Available Operations:

- GET List available applications
- POST Create an application
- GET Get an application
- PUT Update an application
- DELETE Delete an application

List available applications

GET https://api.nexmo.com/v2/applications

Authentication

Key	Description	Example	Default
Authorization	Base64 encoded API key and secret joined by a colon. Read more	Basic <base64>	None

Query Parameter

There are multiple versions of this API available

VERSION 1 | VERSION 2

Feedback

Example Responses

200

400

401

405

406

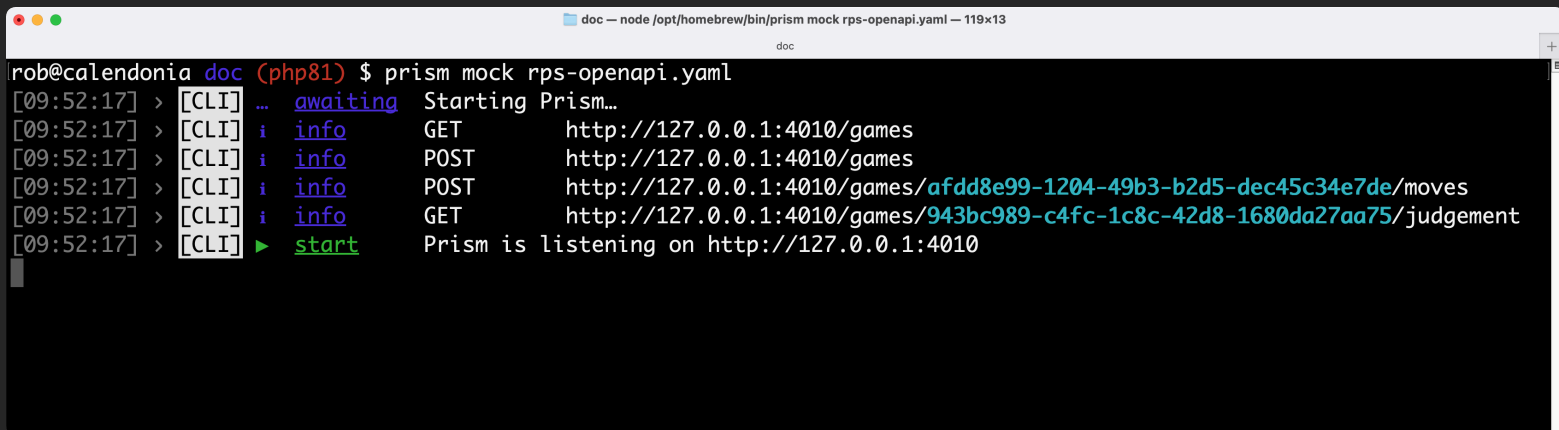
```
{
  "page_size": 10,
  "page": 1,
  "total_items": 6,
  "total_pages": 1,
  "_embedded": {
    "applications": [
      {
        "id": "78d335fa-323d-0114-9c3d-d6f0d48968cf",
```

Rob Allen ~ @akrabad

Developers

Mock server

```
$ prism mock rps-openapi.yaml
```

A terminal window titled "doc — node /opt/homebrew/bin/prism mock rps-openapi.yaml — 119x13" with a "doc" tab. The prompt is "rob@calendonia doc (php81) \$". The command "prism mock rps-openapi.yaml" has been executed. The output shows a series of log messages from the Prism CLI, including "Starting Prism...", "GET http://127.0.0.1:4010/games", "POST http://127.0.0.1:4010/games", "POST http://127.0.0.1:4010/games/afdd8e99-1204-49b3-b2d5-dec45c34e7de/moves", "GET http://127.0.0.1:4010/games/943bc989-c4fc-1c8c-42d8-1680da27aa75/judgement", and finally "Prism is listening on http://127.0.0.1:4010".

```
rob@calendonia doc (php81) $ prism mock rps-openapi.yaml
[09:52:17] > [CLI] ... awaiting Starting Prism...
[09:52:17] > [CLI] i info GET http://127.0.0.1:4010/games
[09:52:17] > [CLI] i info POST http://127.0.0.1:4010/games
[09:52:17] > [CLI] i info POST http://127.0.0.1:4010/games/afdd8e99-1204-49b3-b2d5-dec45c34e7de/moves
[09:52:17] > [CLI] i info GET http://127.0.0.1:4010/games/943bc989-c4fc-1c8c-42d8-1680da27aa75/judgement
[09:52:17] > [CLI] ► start Prism is listening on http://127.0.0.1:4010
```

Make API calls

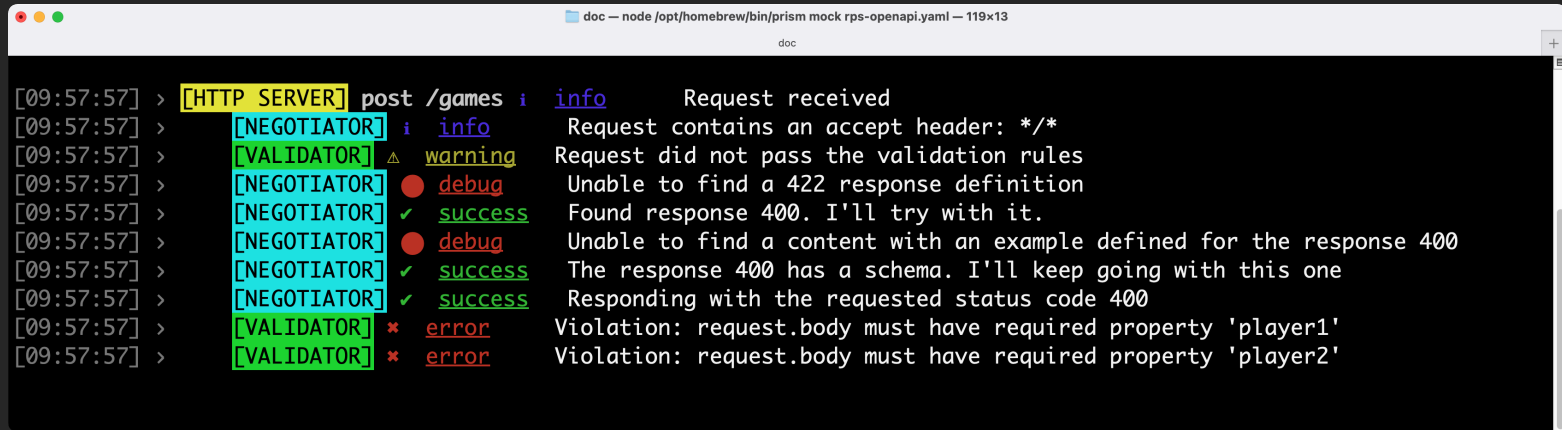
```
$ curl http://127.0.0.1:4010/games -d '{}'
```

Make API calls

```
$ curl http://127.0.0.1:4010/games -d '{}'  
{"message": "Must provide both player1 and player2"}
```


Make API calls

```
$ curl http://127.0.0.1:4010/games -d '{}'  
{"message":"Must provide both player1 and player2"}
```



The screenshot shows a terminal window titled "doc — node /opt/homebrew/bin/prism mock rps-openapi.yaml — 119x13". The terminal displays a series of log messages from an HTTP server and its negotiator/validator components. The messages are timestamped [09:57:57] and show the flow of a request from reception to validation and response.

```
[09:57:57] > [HTTP SERVER] post /games i info Request received  
[09:57:57] > [NEGOTIATOR] i info Request contains an accept header: /*  
[09:57:57] > [VALIDATOR] Δ warning Request did not pass the validation rules  
[09:57:57] > [NEGOTIATOR] ● debug Unable to find a 422 response definition  
[09:57:57] > [NEGOTIATOR] ✓ success Found response 400. I'll try with it.  
[09:57:57] > [NEGOTIATOR] ● debug Unable to find a content with an example defined for the response 400  
[09:57:57] > [NEGOTIATOR] ✓ success The response 400 has a schema. I'll keep going with this one  
[09:57:57] > [NEGOTIATOR] ✓ success Responding with the requested status code 400  
[09:57:57] > [VALIDATOR] * error Violation: request.body must have required property 'player1'  
[09:57:57] > [VALIDATOR] * error Violation: request.body must have required property 'player2'
```



Validation

The schema section can be used to validate the request *and* response

- Validate early and return a 422
- Validate that we return what we say we will
- Put it in CI to prevent regressions

But I already have validation!

Your code:

- isn't good enough!
- isn't reusable!
- doesn't match the docs!



But I already have validation!

Your code:

- isn't good enough!
- isn't reusable!
- doesn't match the docs!

However...

Business logic validation still needed!



Validation in PHP

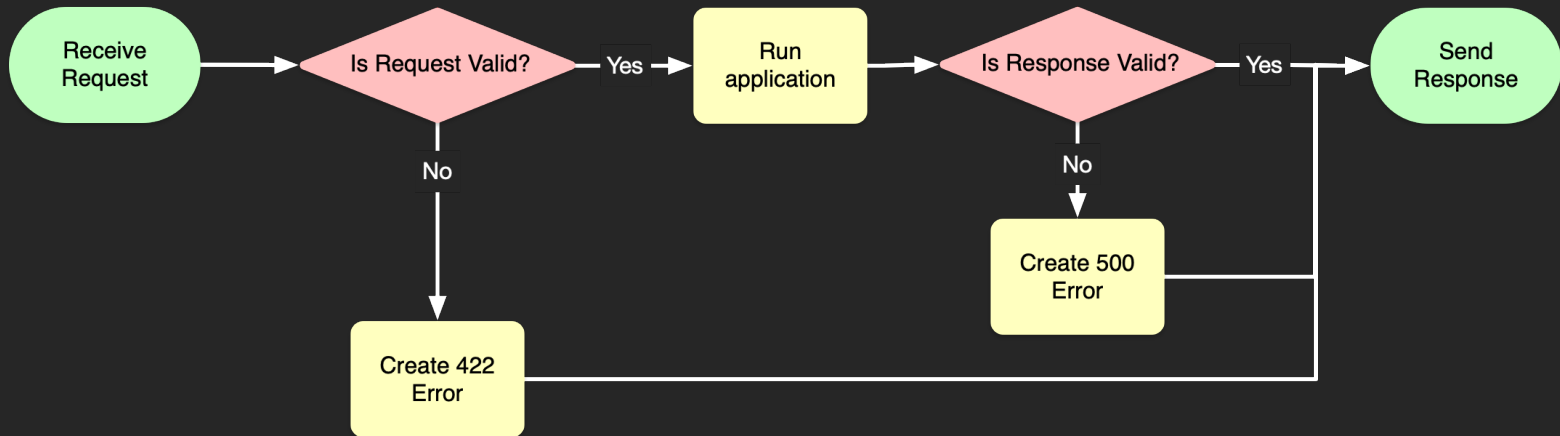
OpenAPI 3.0:

`league/openapi-psr7-validator`

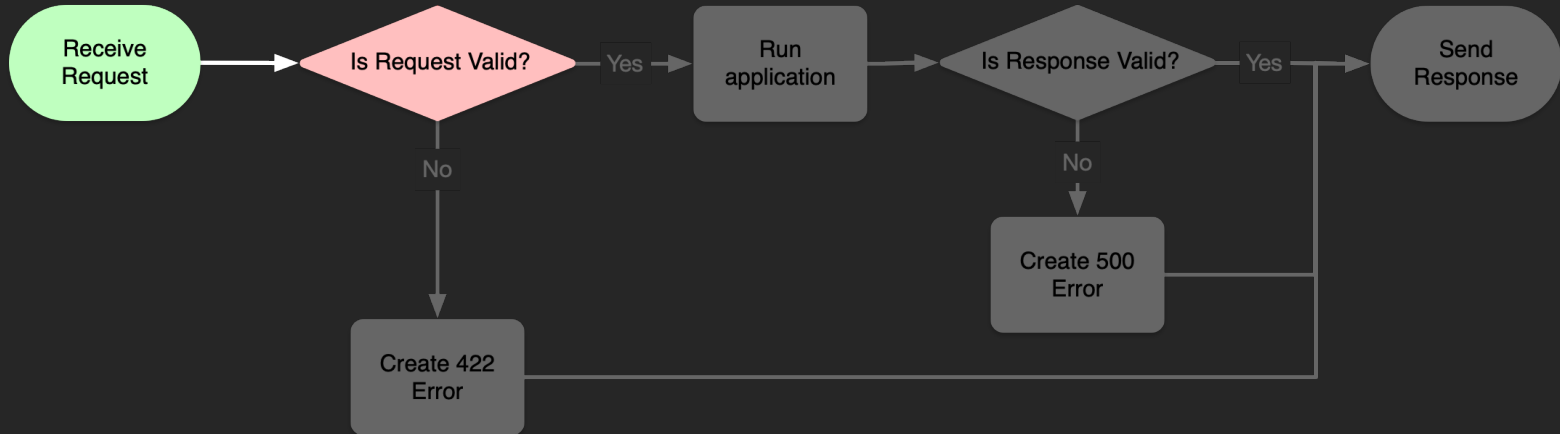
OpenAPI 3.1:

`opis/json-schema`

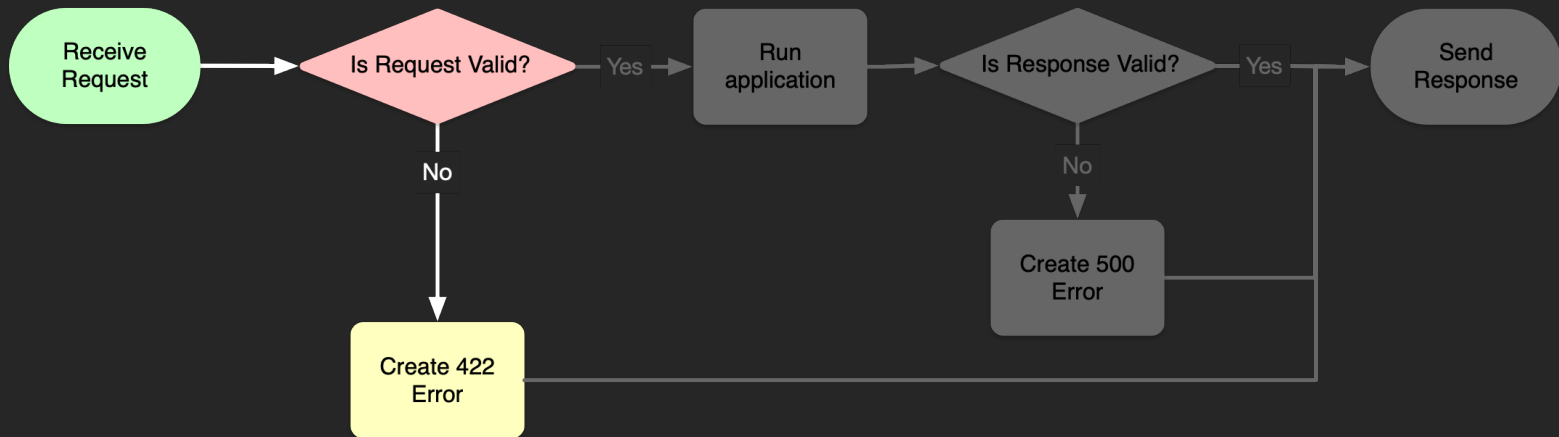
Validation middleware



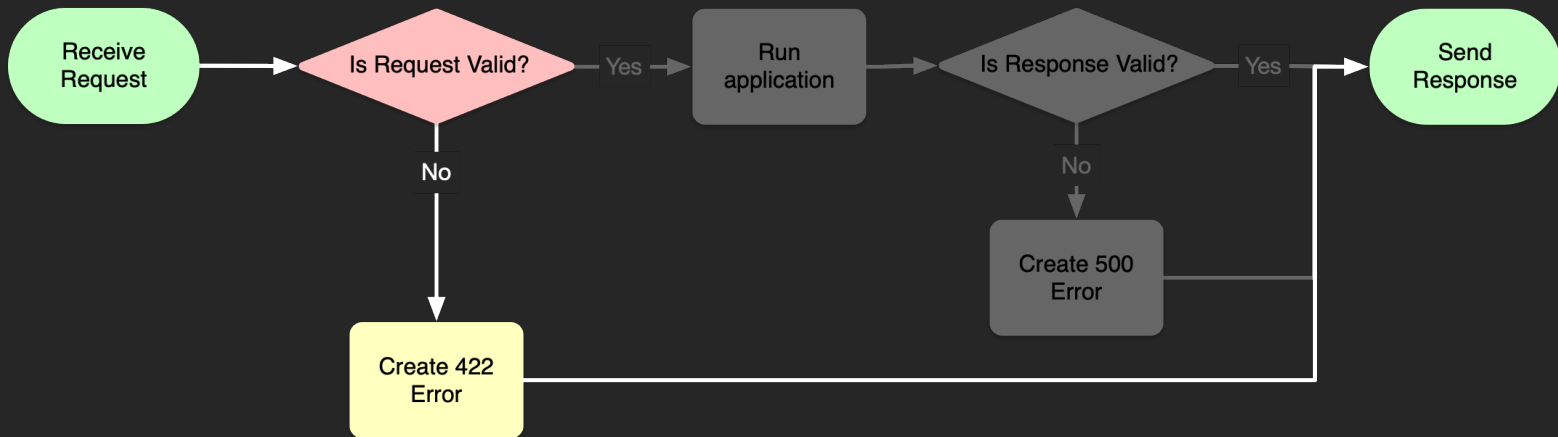
Test Request



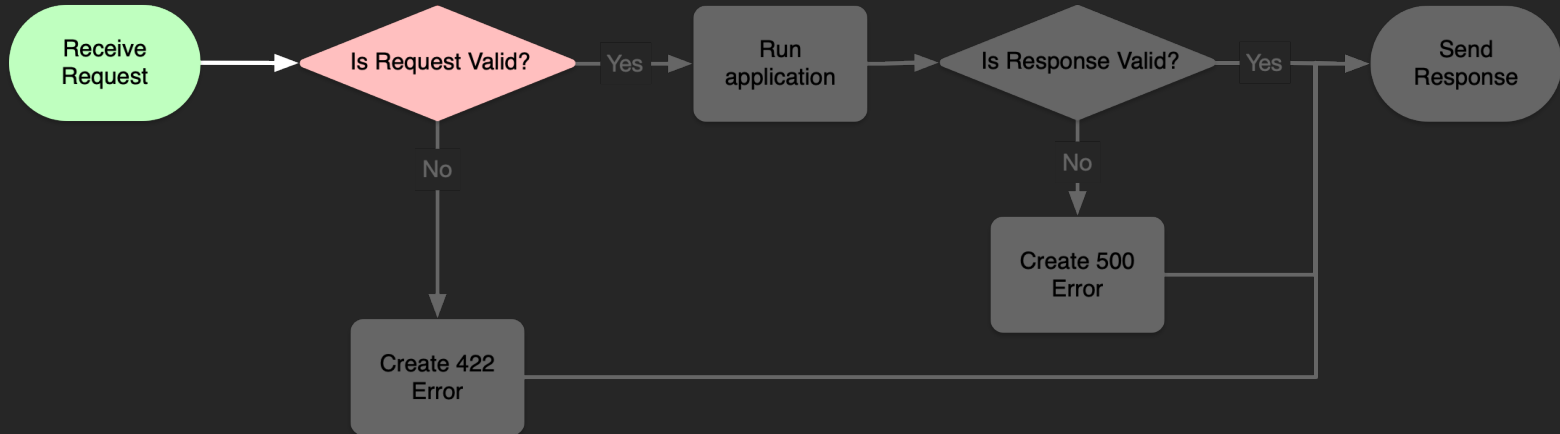
Request is invalid



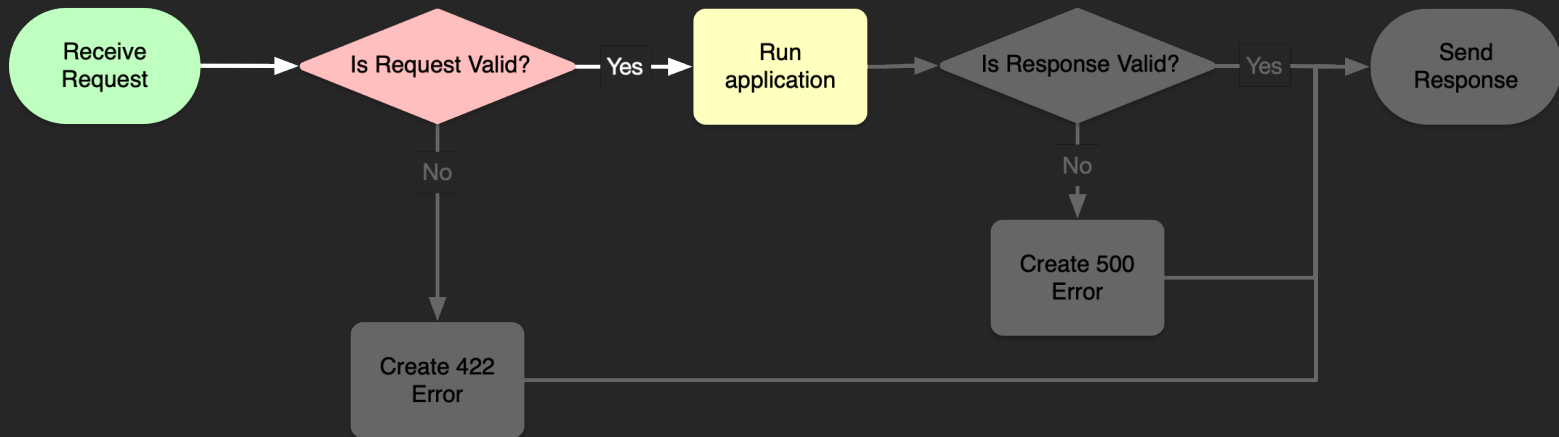
Request is invalid



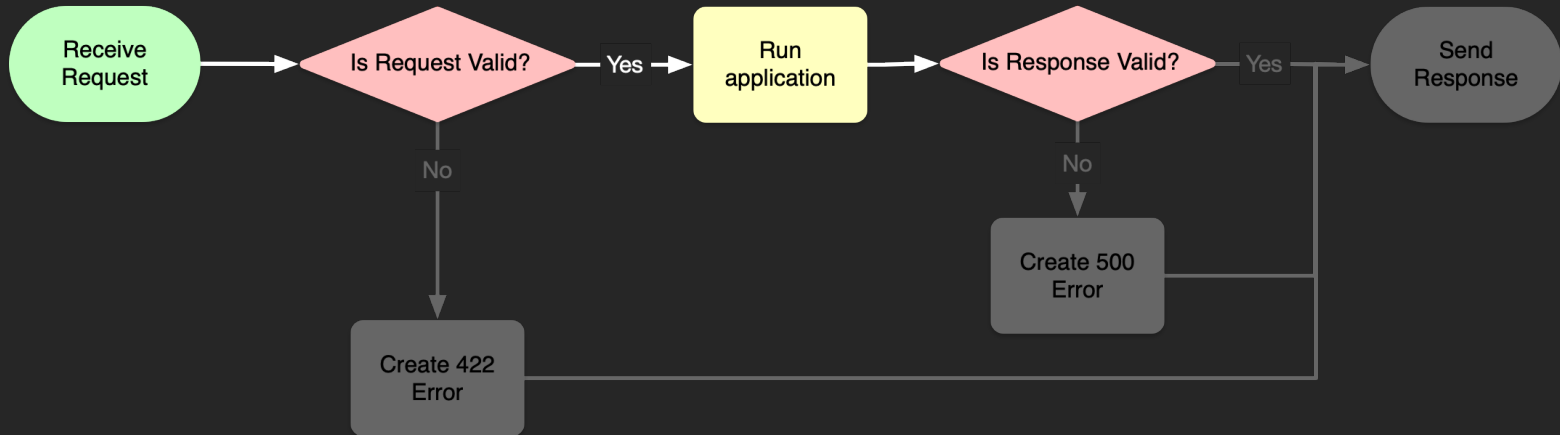
Test Request



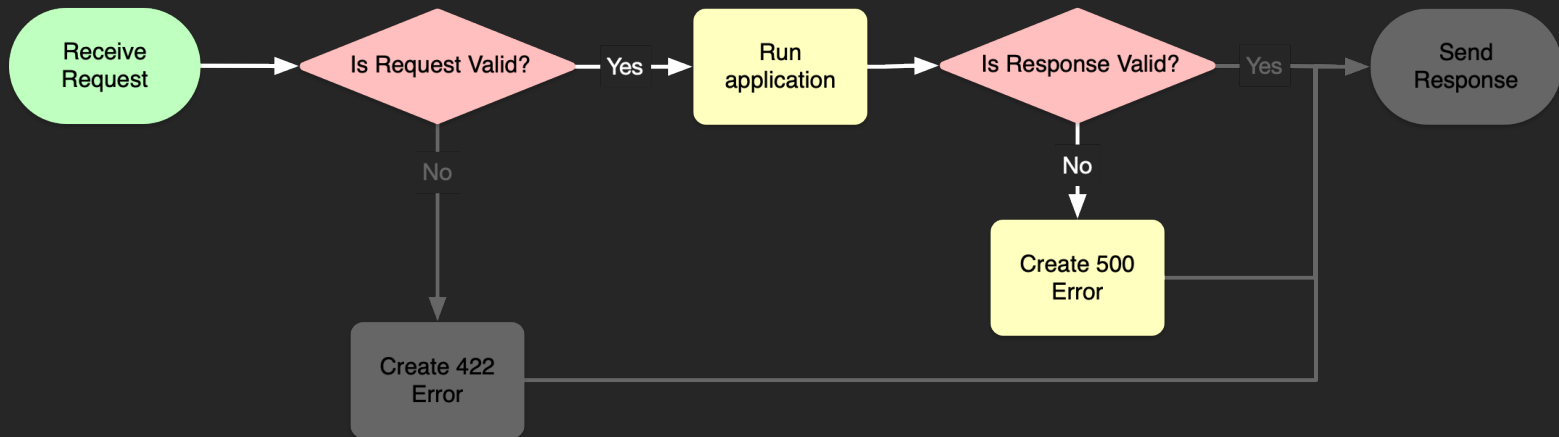
Request is valid



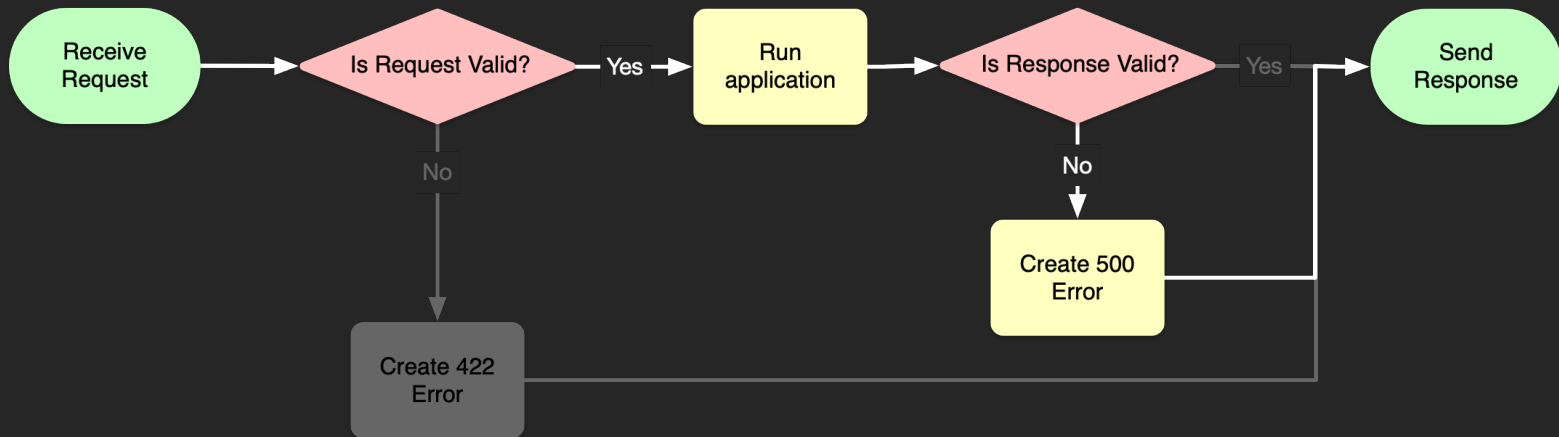
Test Response



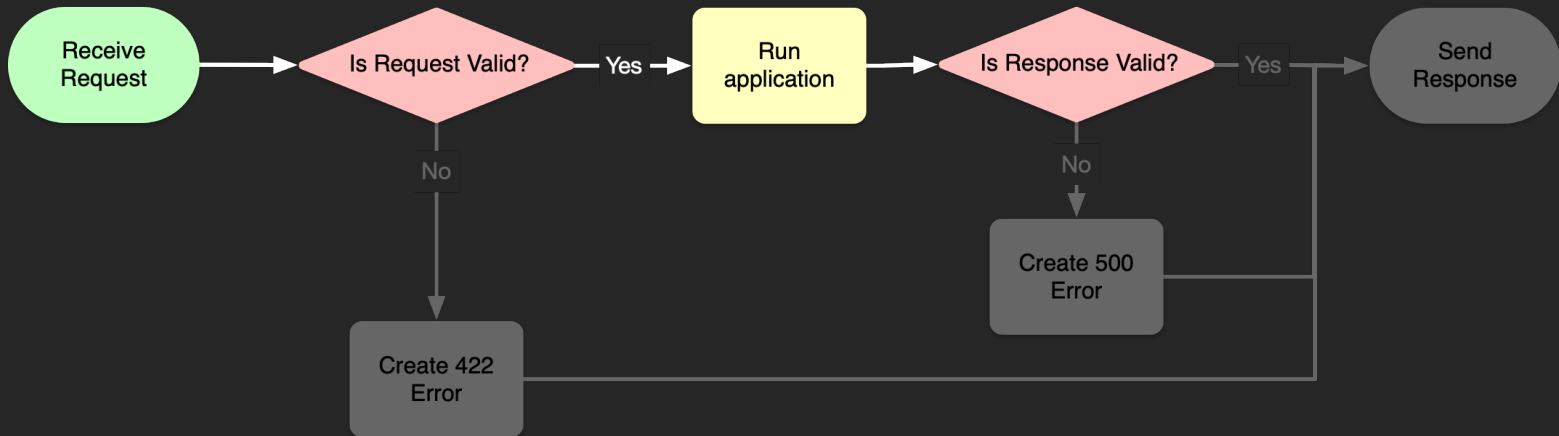
Response is invalid



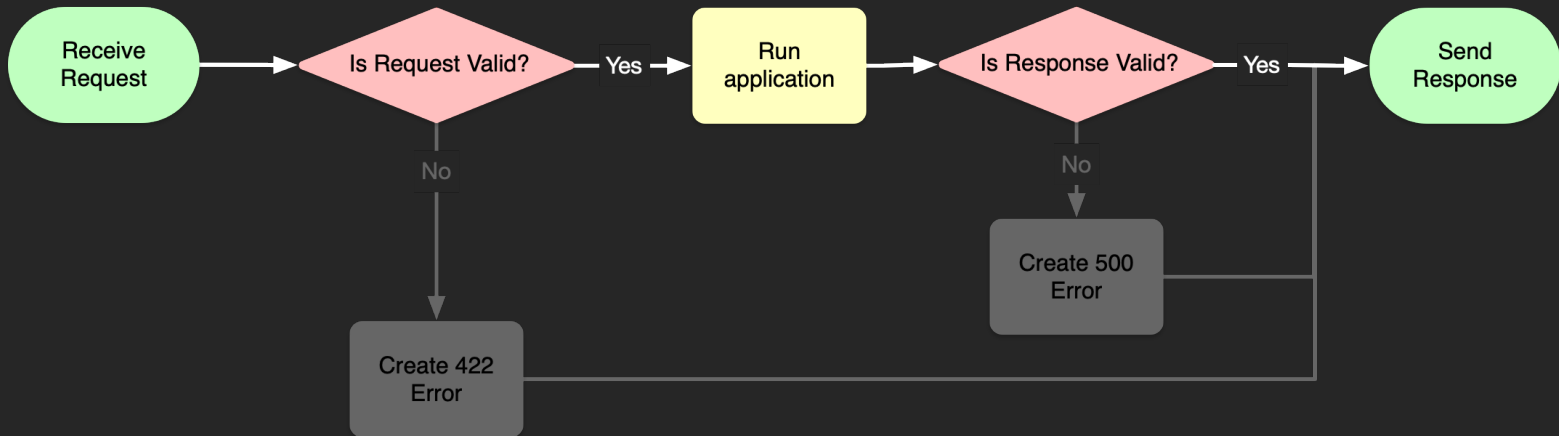
Response is invalid



Successful validation



Successful validation



Validation middleware

```
class OpenApiValidationMiddleware implements MiddlewareInterface
{
    public function __construct(string $oasFilename, Cache $cache)
    {
        $builder = new ValidatorBuilder();
        $builder->fromYamlFile($oasFilename);
        $builder->setCache($cache)->overrideCacheKey('openapi');

        $this->reqValidator = $builder->getServerRequestValidator();
        $this->respValidator = $builder->getResponseValidator();
    }

    public function process($request, $handler)
    {
        try {
            // validate request
            $match = $this->reqValidator->validate($request);
        } catch (ValidationFailed $e) {
            throw new HttpException($request, 422, $e);
        }

        // process
        $response = $handler->handle($request);

        try {
            // validate response
            $this->respValidator->validate($match, $response);
            return $response;
        } catch (ValidationFailed $e) {
            throw new HttpException($request, 500, $e);
        }
    }
}
```



Validation middleware

```
public function __construct(string $oasFilename, Cache $cache)
{
    $builder = new ValidatorBuilder();
    $builder->fromYamlFile($oasFilename);
    $builder->setCache($cache)->overrideCacheKey('openapi');

    $this->reqValidator = $builder->getServerRequestValidator();
    $this->respValidator = $builder->getResponseValidator();
}
```

Validation middleware

```
public function __construct(string $oasFilename, Cache $cache)
{
    $builder = new ValidatorBuilder();
    $builder->fromYamlFile($oasFilename);
    $builder->setCache($cache)->overrideCacheKey('openapi');

    $this->reqValidator = $builder->getServerRequestValidator();
    $this->respValidator = $builder->getResponseValidator();
}
```

Validation middleware

```
public function __construct(string $oasFilename, Cache $cache)
{
    $builder = new ValidatorBuilder();
    $builder->fromYamlFile($oasFilename);
    $builder->setCache($cache)->overrideCacheKey('openapi');

    $this->reqValidator = $builder->getServerRequestValidator();
    $this->respValidator = $builder->getResponseValidator();
}
```



Validation middleware

```
public function __construct(string $oasFilename, Cache $cache)
{
    $builder = new ValidatorBuilder();
    $builder->fromYamlFile($oasFilename);
    $builder->setCache($cache)->overrideCacheKey('openapi');

    $this->reqValidator = $builder->getServerRequestValidator();
    $this->respValidator = $builder->getResponseValidator();
}
```



Validation middleware

```
public function process($request, $handler)
{
    try {
        // validate request
        $match = $this->reqValidator->validate($request);
    } catch (ValidationFailed $e) {
        throw new HttpException($request, 422, $e);
    }
}
```

Validation middleware

```
public function process($request, $handler)
{
    try {
        // validate request
        $match = $this->reqValidator->validate($request);
    } catch (ValidationFailed $e) {
        throw new HttpException($request, 422, $e);
    }
}
```



Validation middleware

```
public function process($request, $handler)
{
    try {
        // validate request
        $match = $this->reqValidator->validate($request);
    } catch (ValidationFailed $e) {
        throw new HttpException($request, 422, $e);
    }
}
```



Validation middleware

```
public function process($request, $handler)
{
    ...

    // process
    $response = $handler->handle($request);
```



Validation middleware

```
public function process($request, $handler)
{
    ...

    try {
        // validate response
        $this->respValidator->validate($match, $response);
        return $response;
    } catch (ValidationFailed $e) {
        throw new HttpException($request, 500, $e);
    }
}
```

Validation middleware

```
public function process($request, $handler)
{
    ...

    try {
        // validate response
        $this->respValidator->validate($match, $response);
        return $response;
    } catch (ValidationFailed $e) {
        throw new HttpException($request, 500, $e);
    }
}
```

Validation middleware

```
public function process($request, $handler)
{
    ...

    try {
        // validate response
        $this->respValidator->validate($match, $response);
        return $response;
    } catch (ValidationFailed $e) {
        throw new HttpException($request, 500, $e);
    }
}
```

Compliance Testing

Schemathesis reads your OpenAPI spec and tests your API against it

```
pip install schemathesis
```

```
schemathesis run --stateful=links --checks all \  
  --base-url=http://localhost:8888 \  
  doc/rps-openapi.yaml
```

Compliance Testing

```
doc -- doc -- -bash -- 119x26
===== Schemathesis test session starts =====
platform Darwin -- Python 3.9.7, schemathesis-3.12.3, hypothesis-6.36.0, hypothesis_jsonschema-0.22.0, jsonschema-4.4.0
rootdir: /Users/rob/Projects/slimng/slim4-rps-api/doc
hypothesis profile 'default' -> database=DirectoryBasedExampleDatabase('/Users/rob/Projects/slimng/slim4-rps-api/doc/.hypothesis/examples')
Schema location: file:///Users/rob/Projects/slimng/slim4-rps-api/doc/rps-openapi.yaml
Base URL: http://localhost:8888
Specification version: Open API 3.0.3
Workers: 1
Collected API operations: 4

GET /games . [ 25%]
POST /games . [ 50%]
POST /games/{game_id}/moves . [ 75%]
GET /games/{game_id}/judgement . [100%]

===== SUMMARY =====

Performed checks:
not_a_server_error          306 / 306 passed PASSED
status_code_conformance    306 / 306 passed PASSED
content_type_conformance    306 / 306 passed PASSED
response_headers_conformance 306 / 306 passed PASSED
response_schema_conformance 306 / 306 passed PASSED

===== 4 passed in 85.55s =====
```



Other Interesting Tools

- *Optic*: BC Break Detection
- *php-openapi-faker*: Create fake data from OpenAPI spec
- *Response2Schema*: Generate OpenAPI spec from JSON object
- *Laravel OpenAPI*: Generate OpenAPI spec from a Laravel app

Many more at <https://openapi.tools>

A dramatic photograph of a space shuttle launch at dusk or dawn. The shuttle is ascending vertically, leaving a massive, billowing plume of white and orange smoke and fire. Three tall, slender service towers stand on either side of the launch pad, their silhouettes visible against the bright light of the engine exhaust. The sky is a deep, dark blue, and the ground in the foreground is dark and silhouetted.

To sum up

Resources

- <https://www.openapis.org>
- <https://openapi.tools>
- <https://github.com/thephpleague/openapi-psr7-validator>
- <https://github.com/akrabat/slim4-rps-api>



Wandering Woodsman 🔥🚴🌳

@philsturgeon



If you've not got a test suite, YOU NEED A TEST SUITE.

If you've not got OpenAPI, why are you making every step of the API lifecycle worse, slower, and more manual.

API Design-First: <https://apisyouwonthate.com/blog/api-design-first-vs-code-first>

Or, play catchup: <https://apisyouwonthate.com/blog/creating-openapi-from-http-traffic>

Either way, go get OpenAPI.

11:49 AM · Feb 5, 2022 · Twitter Web App



Thank you!

<https://joind.in/talk/dbda1>

Photo credits

- Scaffolding: <https://www.flickr.com/photos/pagedooley/49683539647>
- Writing: <https://www.flickr.com/photos/throughkikslens/14516757158>
- Books: <https://www.flickr.com/photos/eternaletulf/41166888495>
- Computer code: <https://www.flickr.com/photos/n3wjack/3856456237>
- Rocket launch: <https://www.flickr.com/photos/gsfcr/16495356966>
- Stars: <https://www.flickr.com/photos/gsfcr/19125041621>