Instructions:

1. Print pages 2-9
2. Cut into note cards
3. Optional: rewrite by hand on 4x6 cards for maximal learning

Important:

OpenAPI is a specification used to describe an HTTP API. The API description is composed of building blocks with different types of data. The note cards represent different building blocks. The outermost building block is called the **`Root`**. Not all of the building blocks are included (this should satisfy 99% of the cases). Each note card is for a given type (titled in the heading). Then, it has a property (left-side) and type (right-side) in columns. The properties that are **bold** are required. Some types are scalar (bottom-level building blocks), and others are referencing other types as building blocks. Note that the building blocks with "**`Maps`**" as a suffix are special and represent a map with keys and the corresponding value type is without the "Maps" suffix. For example, **`MediaTypesMap`** corresponds to the **`MediaType`** type. One exception is the **`WebhooksMap`** is a map of keys to **`PathItem`** objects. Similarly, in components the objects prefixed with "**`Named`**" are maps of names to the corresponding type of data (such as NamedParameters are maps of names to **`Parameter`** objects). In the following snippet, "`application/json`" is the map key in a **`MediaTypesMap`**, and everything that follows is a **`MediaType`** type.

```
content:
  application/json:
    schema:
      type: object
      properties:
        id:
          type: string
        createdAt:
          type: string
          format: date-time
```

The examples here and in the note cards are almost always incomplete and abbreviated. They serve as a reminder or hint to help unblock you.

Root

openapi	**`string`** (3.1.0)
info	**`Info`** object
servers	List of **`Server`** objects
security	List of **`SecurityRequirement`** objects
tags	List of **`Tag`** objects
externalDocs	**`ExternalDocs`** object
paths	**`Paths`** object
webhooks	**`WebhooksMap`** object
components	**`Components`** object
jsonSchemaDialect	**`string`**

Example:

```
openapi: 3.1.0
info: …
components: {}
```

Info

title	**`string`**
version	**`string`**
description	**`string`**
termsOfService	**`string`**
summary	**`string`**
contact	**`Contact`** object
license	**`License`** object

Example:

```
info:
  title: FooTube.net
  version: v1
```

License

name	**`string`**
url	**`string`**
identifier	**`string`**

Example:

```
license:
  name: MIT
```

Contact

name	**`string`**
url	**`string`**
email	**`string`**

Example:

```
contact:
  name: Adam
```

Server

url	**string**
description	**string**
variables	**ServerVariablesMap**

Example:

```
servers:
  - url: https://api.remockly.com/v1
    description: Mock server
```

ServerVariable

default	**string**
enum	**[string]**
description	**string**

Example:

```
servers:
  - url: https://{tenant}.remockly.com/v1
    description: Mock server
    variables:
      tenant:
        default: www
        description: Your tenant id
```

SecurityRequirement

[string]

The names of security requirements are defined in the `Components` `NamedSecuritySchemes`*.*

Example:

```
security:
  - JWT: []
```

Tag

name	**string**
description	**string**
externalDocs	**ExternalDocs** object

Example:

```
tags:
  - name: Feedback
    description: A **wonderful** example.
  - name: Approvals
```

ExternalDocs

url	**`string`**
description	**`string`**

Paths

starts with **'/'**	**`PathItem`** object

Example:

```
paths:
  /feedback:
    get:
      responses:
        200:
          description: OK
  /approvals:
    $ref: ./paths/approvals.yaml
```

PathItem

servers	List of **`Server`** objects
parameters	List of **`Parameter`** objects
summary	**`string`**
description	**`string`**
get	**`Operation`** object
put	**`Operation`** object
post	**`Operation`** object
delete	**`Operation`** object
options	**`Operation`** object
head	**`Operation`** object
patch	**`Operation`** object
trace	**`Operation`** object

Operation

responses	**`Responses`** object
tags	**`[string]`**
parameters	List of **`Parameter`** objects
summary	**`string`**
description	**`string`**
externalDocs	**`ExternalDocs`** object
operationId	**`string`**
security	List of **`SecurityRequirement`**
servers	List of **`Server`** objects
requestBody	**`RequestBody`** object
deprecated	**`boolean`**
callbacks	**`CallbacksMap`** object

Parameter

name	**`string`**
in	**`string`**
description	**`string`**
required	**`boolean`**
deprecated	**`boolean`**
allowEmptyValue	**`boolean`**
style	**`string`**
explode	**`boolean`**
allowReserved	**`boolean`**
schema	**`Schema`** object
example	**`any`**
examples	**`ExamplesMap`** object
content	**`MediaTypesMap`** object

RequestBody

content	**`MediaTypesMap`** object
required	**`boolean`**
description	**`string`**

MediaType

schema	**`Schema`** object
example	**`any`**
examples	**`ExamplesMap`** object
encoding	**`EncodingMap`** object

Example:

```
application/json:
  schema:
    type: object
    properties:
      id:
        type: string
      createdAt:
        type: string
        format: date-time
```

Example

value	**`any`**
summary	**`string`**
description	**`string`**
externalValue	**`string`**

Example:

```
value:
  id: abc123
  createdAt: 2023-01-29T10:59:21Z
summary: Case ABC
```

Encoding

contentType	**`string`**
headers	**`HeadersMap`** object
style	**`string`**
explode	**`boolean`**
allowReserved	**`boolean`**

Header

description	**`string`**
required	**`boolean`**
deprecated	**`boolean`**
allowEmptyValue	**`boolean`**
style	**`string`**
explode	**`boolean`**
allowReserved	**`boolean`**
schema	**`Schema`** object
example	**`any`**
examples	**`ExamplesMap`** object
content	**`MediaTypesMap`** object

*Mutually exclusive properties **schema** or **content** is required.*

Responses

default	**`Response`** object
100-600	**`Response`** object
1XX, 2XX, 3XX, 4XX, 5XX	

Example:

```
responses:
  default: {Response}
  200: {Response}
  4XX: {Response}
```

Response

description	**`string`**
headers	**`HeadersMap`** object
content	**`MediaTypesMap`** object
links	**`LinksMap`** object

Components

parameters	**NamedParameters**
schemas	**NamedSchemas**
responses	**NamedResponses**
examples	**NamedExamples**
requestBodies	**NamedRequestBodies**
headers	**NamedHeaders**
securitySchemes	**NamedSecuritySchemes**
links	**NamedLinks**
callbacks	**NamedCallbacks**

Example:

```
components:
  schemas:
    Feedback: {Schema}
    Followers: {Schema}
```

SecurityScheme

type	**string**
in	**string**
name	**string**
description	**string**
scheme	**string**
bearerFormat	**string**
flows	**OAuth2Flows**
openIdConnectUrl	**string**

Example:

```
components:
  securitySchemes:
    RedoclyKey:
      type: apiKey
      in: header
      name: api_key
```

OAuth2Flows

implicit	**ImplicitFlow** object
password	**PasswordFlow** object
clientCredentials	**ClientCredentials**
authorizationCode	**AuthorizationCode**

Example:

```
components:
  securitySchemes:
    RedoclyAuth:
      type: oauth2
      flows:
        authorizationCode:
          …
```

ImplicitFlow

refreshUrl	**string**
scopes	**object**
authorizationUrl	**string**

Example:

```
implicit:
  scopes:
    read: Read data.
    create: Create new data.
    update: Update data.
  authorizationUrl: https://example.com/au
```

PasswordFlow

refreshUrl	**string**
scopes	**object**
tokenUrl	**string**

Example:

```
password:
  scopes:
    read: Read data.
    create: Create new data.
    update: Update data.
  tokenUrl: https://example.com/token/
```

ClientCredentials

refreshUrl	**string**
scopes	**object**
tokenUrl	**string**

Example:

```
clientCredentials:
  scopes:
    read: Read data.
    create: Create new data.
    update: Update data.
  tokenUrl: https://example.com/token/
```

AuthorizationCode

refreshUrl	**string**
scopes	**object**
authorizationUrl	**string**
tokenUrl	**string**
x-usePkce	**boolean** | **XUsePkce**

Example:

```
authorizationCode:
  x-usePkce:
    disableManualConfiguration: false
    hideClientSecretInput: false
  authorizationUrl: 'https://gitlab.com/oauth/authorize'
  tokenUrl: 'https://gitlab.com/oauth/token'
  scopes:
    'api': Grants complete read/write access to the API.
    'read_user': Grants read-only access.
```

Discriminator

mapping	**DiscriminatorMapping**
propertyName	**string**

Example:

```
discriminator:
  propertyName: type
  mapping:
    sentiment: ./sentiment.yaml
    rating: ./rating.yaml
```

Schema (part 1 of 4)

externalDocs	**ExternalDocs**
discriminator	**Discriminator**
title	**string**
description	**string**
format	**string**
multipleOf	**number**
maximum	**number**
minimum	**number**
exclusiveMaximum	**number**
exclusiveMinimum	**number**
maxLength	**integer**
minLength	**integer**
pattern	**string**
maxItems	**integer**
minItems	**integer**

Schema (part 2 of 4)

uniqueItems	**boolean**
maxProperties	**integer**
minProperties	**integer**
required	**[string]**
enum	**[any]**
allOf	**[Schema]**
anyOf	**[Schema]**
oneOf	**[Schema]**
not	**Schema**
if	**Schema**
then	**Schema**
else	**Schema**
dependentSchemas	**[Schema]**
prefixItems	**[Schema]**
default	**any**

Schema (part 3 of 4)

examples	**[any]**
example	**any**
deprecated	**boolean**
properties	**SchemaProperties**
items	**Schema** | **boolean**
contains	**Schema**
minContains	**integer**
maxContains	**integer**
patternProperties	**object**
propertyNames	**Schema**
unevaluatedItems	**Schema**
unevaluatedProperties	**Schema** | **boolean**
summary	**string**
additionalProperties	**Schema** | **boolean**

Schema (part 4 of 4)

contentEncoding	**string**
contentMediaType	**string**
readOnly	**boolean**
writeOnly	**boolean**
xml	**Xml**
const	**any**
$comment	**string**
$id	**string**
$anchor	**string**
$defs	**NamedSchemas**
definitions	**NamedSchemas**
$vocabulary	**string**