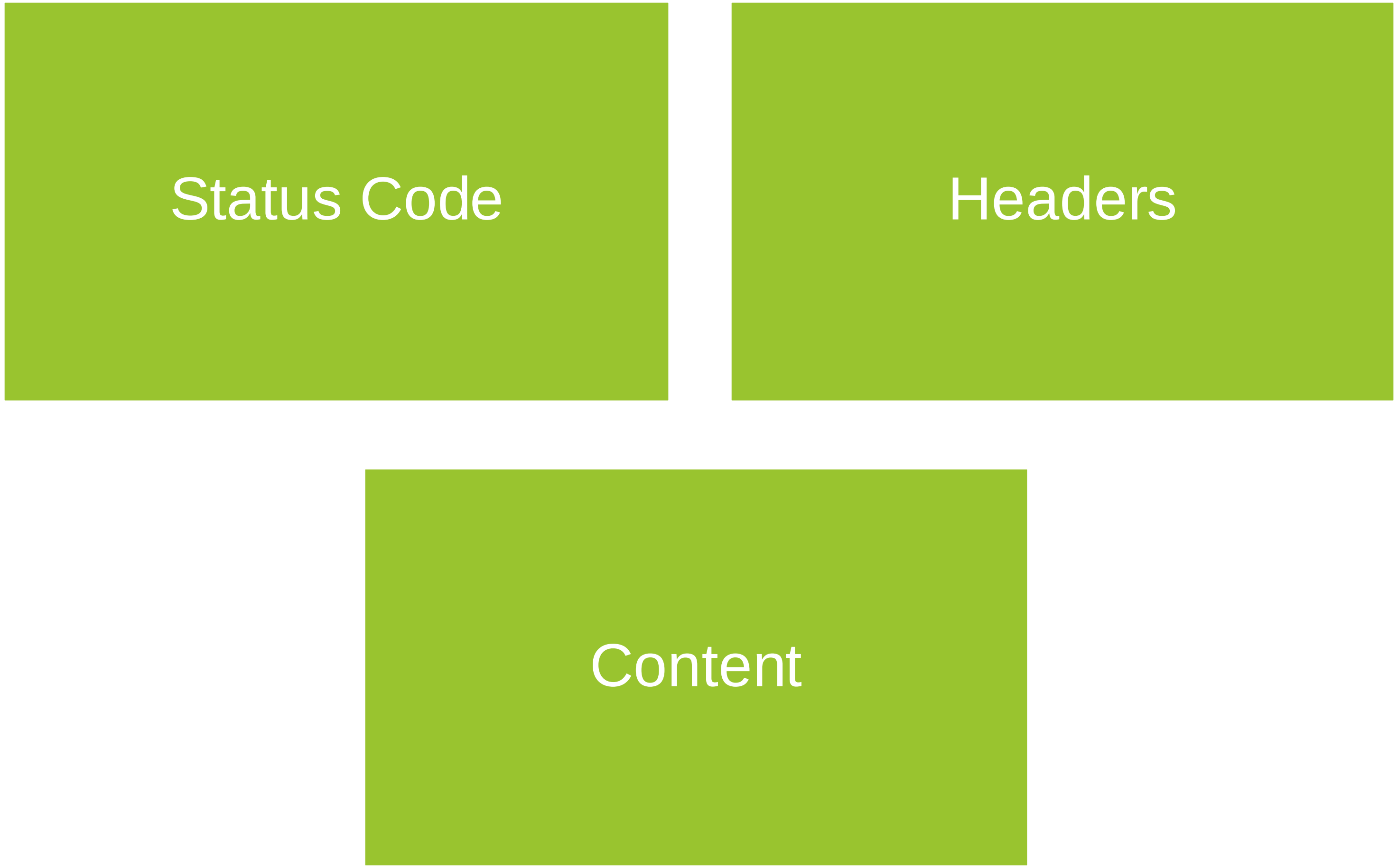
Designing API Responses

Introduction

Response Components



Status Code

Headers

Content

HTTP Status Codes

HTTP Status Code Class

Class	Summary	Description
1XX	Informational	The request is received. The process is continued further.
2XX	Success	The request was successfully received and executed.
3XX	Redirection	Redirected to handle further action.
4XX	Client-side Error	One or more errors in the request sent by the client.
5XX	Server-side Error	One or more errors while processing the request at the server.

2XX Successful Status Codes

200 OK

201 Created

204 No Content

4XX Client Error Status Codes

400 Bad Request

401 Unauthorized

403 Forbidden

404 Not Found

405 Method Not Allowed

5XX Server Error Status Codes

500 Internal Server Error

501 Not Implemented

Designing Status Codes in Responses

HTTP Status Codes for GET Operation

HTTP Status Codes for GET Operation

Successful Operation

200 OK

Resource can’t be found

404 Not Found

HTTP Status Codes for POST Operation

Success
(new resource created)

201 Created

Invalid input

400 Bad Request

Success
(no new resource)

200 OK

Success
(no result exists)

204 No Content

HTTP Status Codes for PUT Operation

Successful update
(no new resource)

200 OK

Success
(new resource created)

201 Created

Success
(no result exists)

204 No Content

HTTP Status Codes for DELETE Operation

Successful deletion

204 No Content

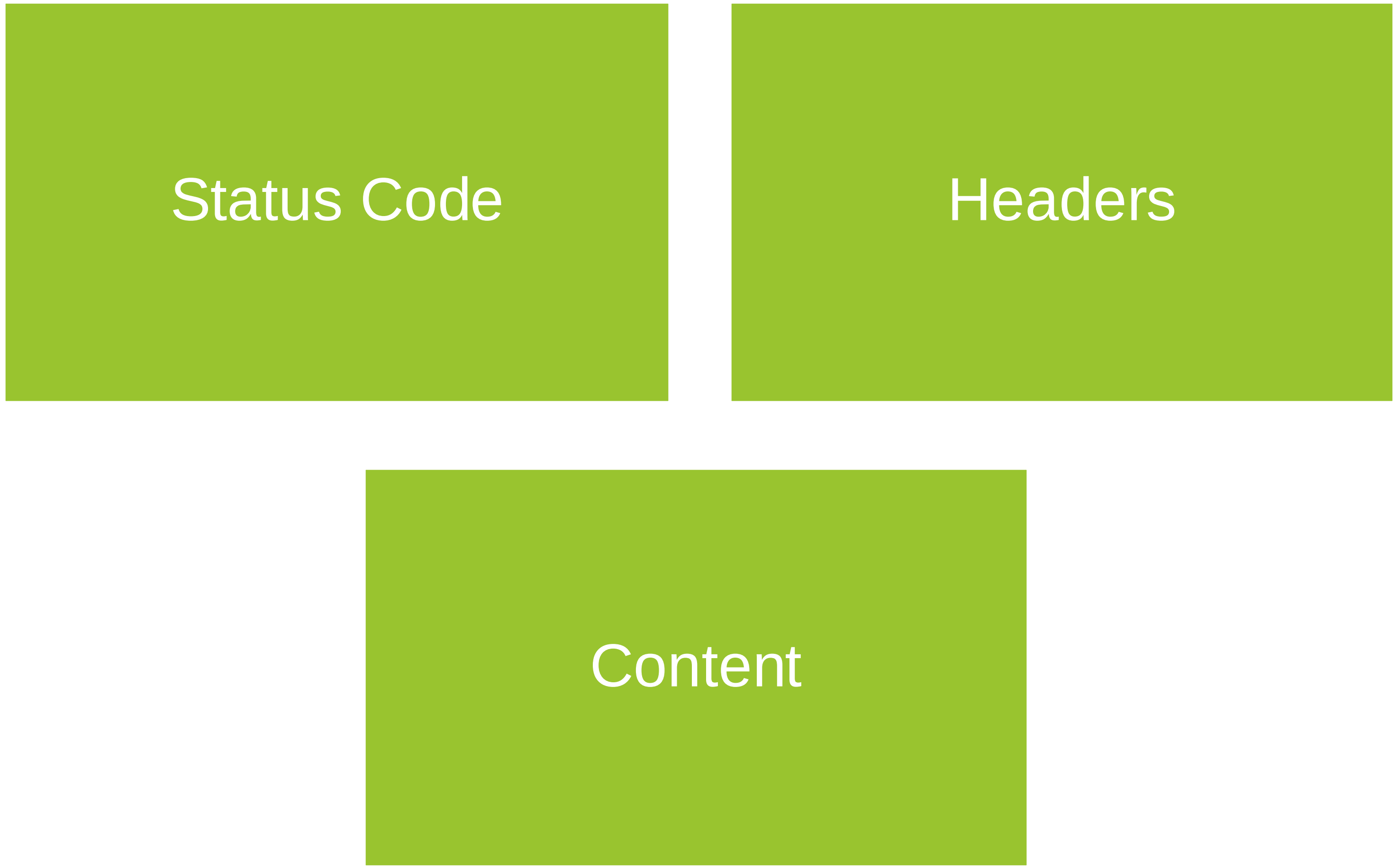
Resource can’t be found

404 Not Found

STEP 14: Identify the HTTP Status Codes for the Operation

Choosing the Format of Responses

Response Components



Status Code

Headers

Content

Choosing the Format of Responses

Format of data

Can support more than 1 format

Available formats

Example Formats

application

- application/json
- application/xml
- application/octet-stream

text

- text/plain
- text/csv
- text/html

image

- image/png
- image/jpeg

Most popular format used in RESTful APIs

application/json

STEP 15: Identify the Content Type of Response for the Operation

Designing Response Body

Designing Response Body

Follow the naming convention

Don’t expose database details

Array of items for collections

Consider supporting pagination for collections

STEP 16: Identify the Response Body for the Operation

Designing Error Handling

Classification of Errors

Client-side Errors

Server-side Errors

Types of Error

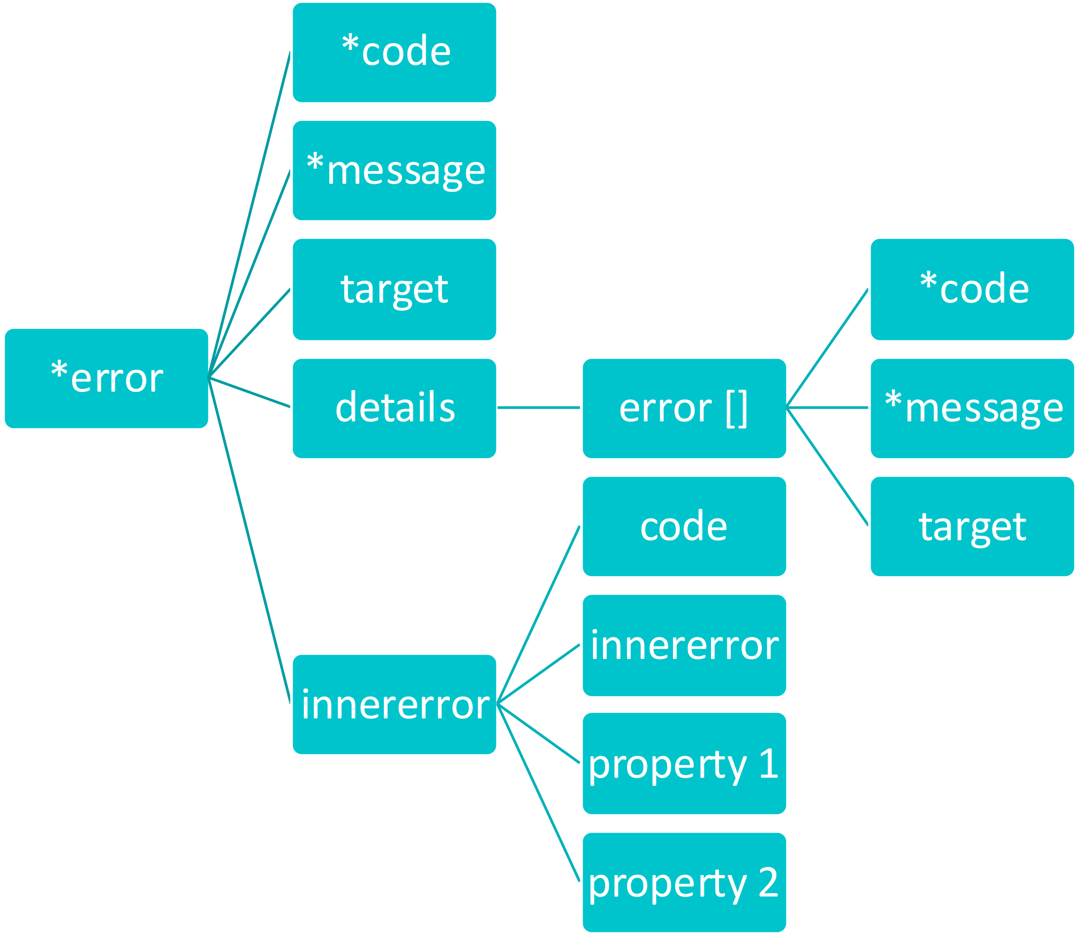
Client-side Errors

- Invalid credentials
- Incorrect parameters
- Invalid IDs

Server-side Errors

- Exceptions

Error Message Format



Error Message Format - Example

```
{
  "error": {
    "code": "INVALID_INPUT",
    "message": "One or more input arguments are invalid",
    "target": "CollegeInfo",
    "details": [
      {
       "code": "INCORRECT_FORMAT",
       "target": "zipcode"
       "message": "Zipcode doesn't follow correct format",
      }
    ]
    "innererror": {
      "message": "Input string wasn't in a correct format",
    }
  }
}
```

STEP 17: Handle Errors for the Operation

Designing API Responses

Summary