How to Write API Documentation as a Technical Writer

BY JORDAN STANCHEV

JPDOCU SCHOOL OF TECHNICAL WRITING

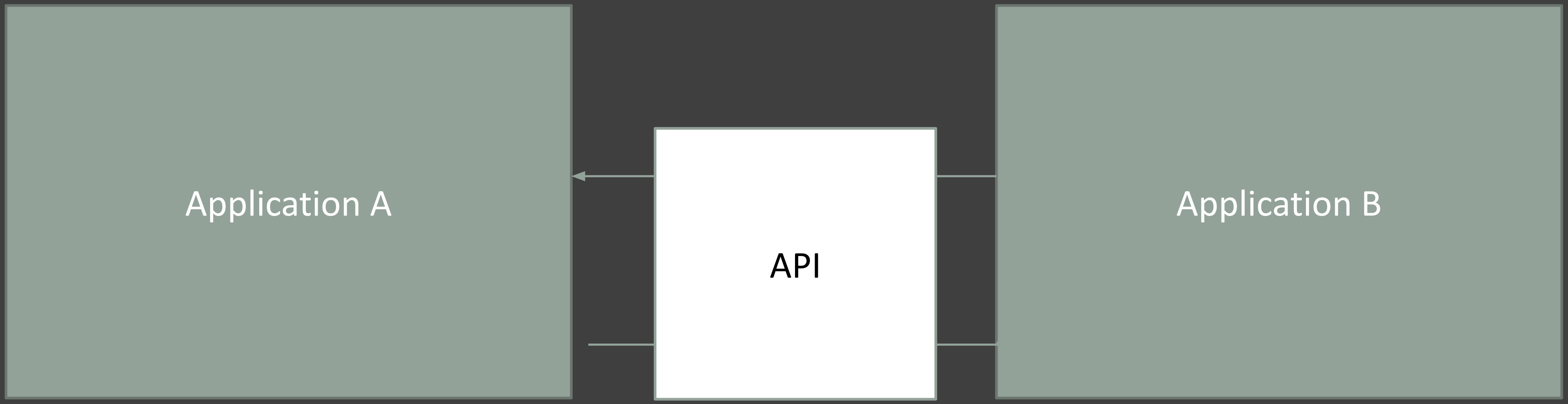
What will you learn?

- ✓What’s an API and what are the benefits from using APIs?
- ✓How does an API work?
- ✓Who and how creates the API and the API documentation?
- ✓Which tools and how to use them to try an API?
- ✓Which tools to use and how to automatically generate API documentation?

What’s an API?

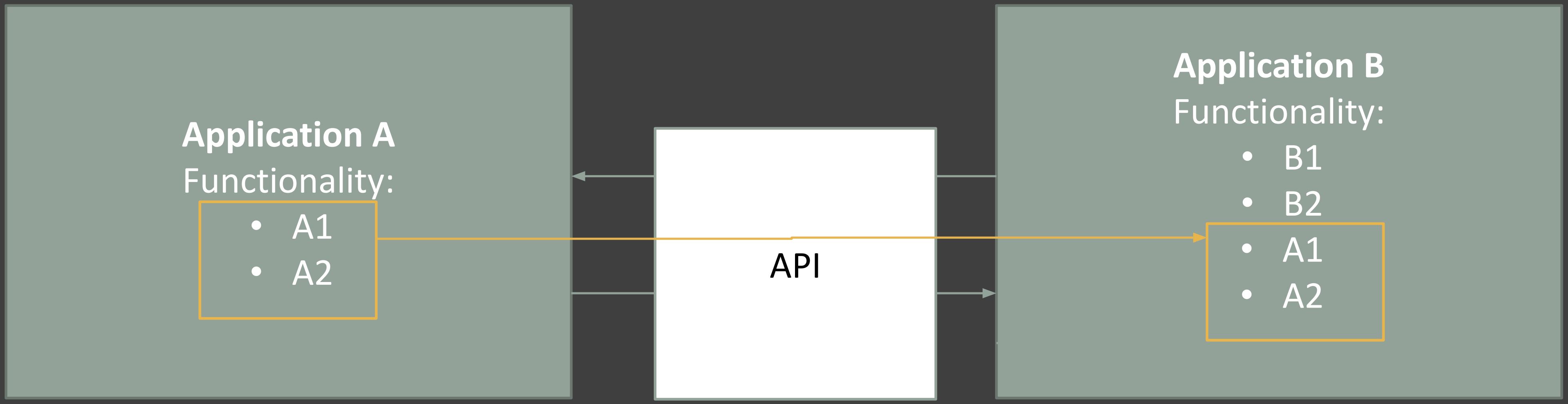
APIs stands for an *application programming interface*.

The API allows two different software application to interact with each other. Using the API these applications can exchange data in a reliable way.



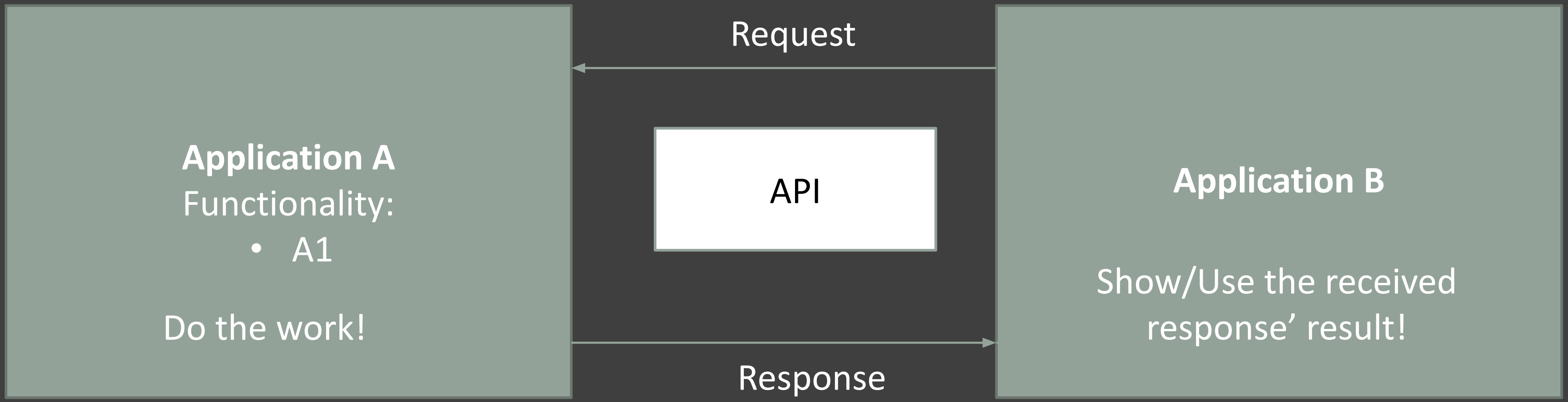
Benefits from Using an API

By using an API, the applications can gain immediate access to software modules and functionality that is already developed by another application. This way you can leverage and re-use existing functionality from another application in the context of your own app, allows you to offer more functionality to your customers.



How Does an API Work?

When using an API, an application first sends a **request** to use the functionality. The receiving application performs all the **work** and in the end, returns a **response**.



Application A
Functionality:
- A1

Do the work!

Request

API

Response

Application B

Show/Use the received response’ result!

Who Creates API and API Documentation?

Usually, you must be the developer who created the API to be able to explain how the API works.

Usually, the technical writers are not the developers, but are a part of the team of developers who creates the API.

As a consequence, as a technical writer would focus on writing documentation about using the API.

What’s an API Documentation?

A type of software documentation that explains:

- What the API can do for me? What functionality does it offer and what work can it do for my application?
- How to connect to the API?
- How to authenticate myself, once connected to the API?
- What are the types of requests I can send to this API? What are their parameters?
- What are the types of responses I can expect to receive, once the API request is being processed? What will be the result I can expect in the end?

Types of APIs

“Standard” API – direct connection protocol

Web API – transfers the request and response via HTTP (Hypertext Transfer Protocol) on the Web.

Who Reads API Documentation?

Documentation about the API is what I call "developers' documentation" - it is for a target audience of **software developers**. To properly use the API, developers need to know the technical details of the API.

Note, that the end-user would **not** need API documentation; for them you provide end-user guides where you aim to reduce technical details and implementation information.

API Documentation - Examples

Stripe is cited frequently as one the best examples of a high-quality API documentation. https://stripe.com/docs/api

SAP’s API documentation sample: https://api.sap.com/api/storage_manager/resource

Apple’s API documentation: https://developer.apple.com/documentation/

Facebook’s API documentation: https://developers.facebook.com/docs/

Google API documentation: https://developers.google.com/workspace/products

API Documentation – Framework

Where to start – first step is to automatically generate documentation from the API design annotations themselves.

Use tools such as https://swagger.io/solutions/api-documentation/

What's a tech writer to do next?! Write the rest of the documentation!

What's the information you must provide in your API documentation to complement the automatically generated one make it a great user experience:

1. How to access the API?
2. What type of requests the API accepts?
3. What type of responses the API returns?

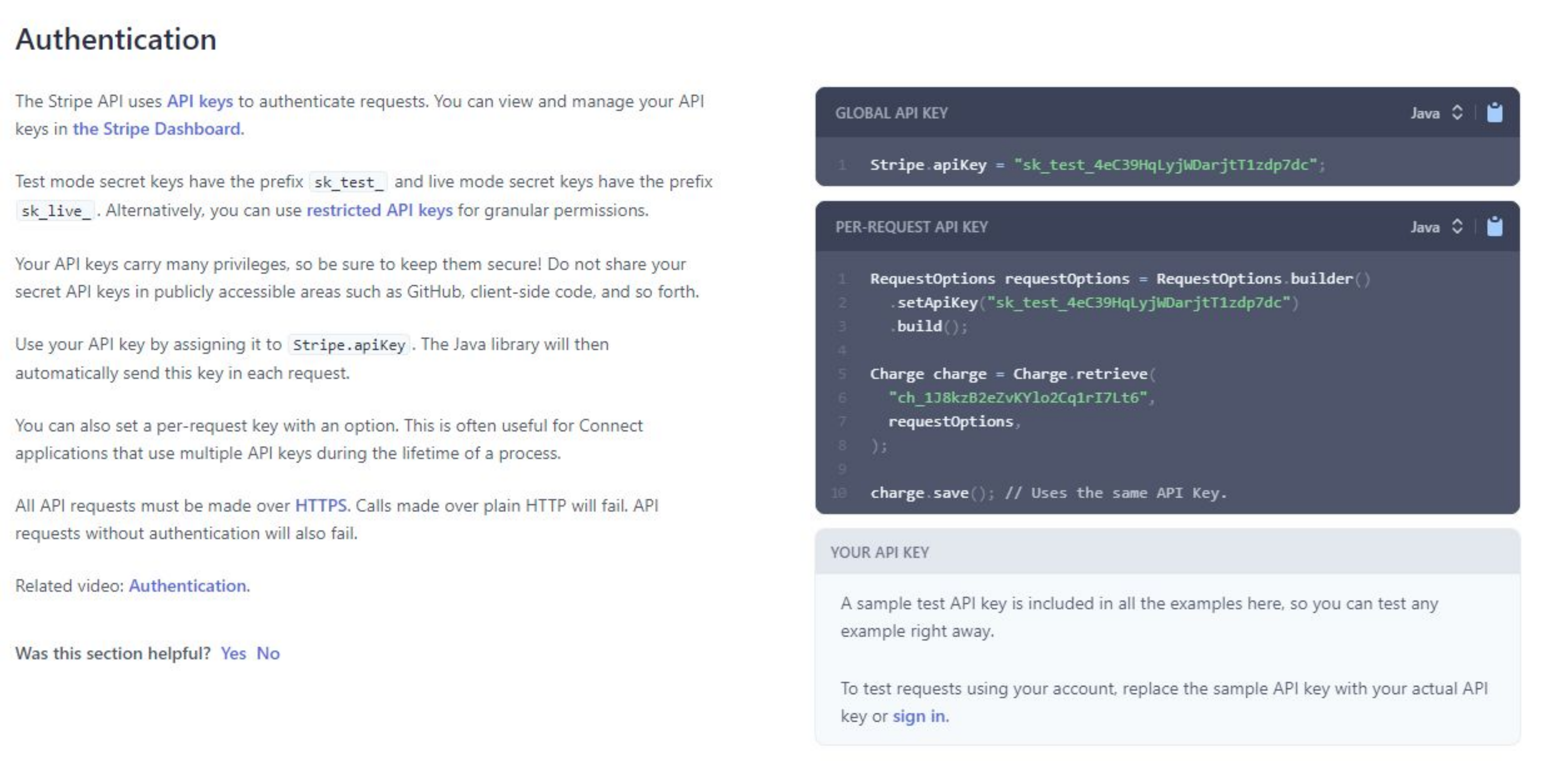
API Documentation – Structure

What’s the information you must provide in your API documentation to make it good?

1. How to access the API?
2. What type of requests the API accepts?
3. What type of responses the API returns?

Step 1: Accessing the API

Describe how to authenticate and access the API (programmatically!) in the supported languages:



Authentication

The Stripe API uses API keys to authenticate requests. You can view and manage your API keys in the Stripe Dashboard.

Test mode secret keys have the prefix `sk_test_` and live mode secret keys have the prefix `sk_live_`. Alternatively, you can use restricted API keys for granular permissions.

Your API keys carry many privileges, so be sure to keep them secure! Do not share your secret API keys in publicly accessible areas such as GitHub, client-side code, and so forth.

Use your API key by assigning it to `Stripe.apiKey`. The Java library will then automatically send this key in each request.

You can also set a per-request key with an option. This is often useful for Connect applications that use multiple API keys during the lifetime of a process.

All API requests must be made over HTTPS. Calls made over plain HTTP will fail. API requests without authentication will also fail.

Related video: Authentication.

Was this section helpful? Yes No

GLOBAL API KEY

```java
Stripe.apiKey = "sk_test_4eC39HqLyjWDarjtT1zdp7dc";
```

PER-REQUEST API KEY

```java
RequestOptions requestOptions = RequestOptions.builder()
  .setApiKey("sk_test_4eC39HqLyjWDarjtT1zdp7dc")
  .build();

Charge charge = Charge.retrieve(
  "ch_1J8kzB2eZvKYlo2Cq1rI7Lt6",
  requestOptions,
);

charge.save(); // Uses the same API Key.
```

YOUR API KEY

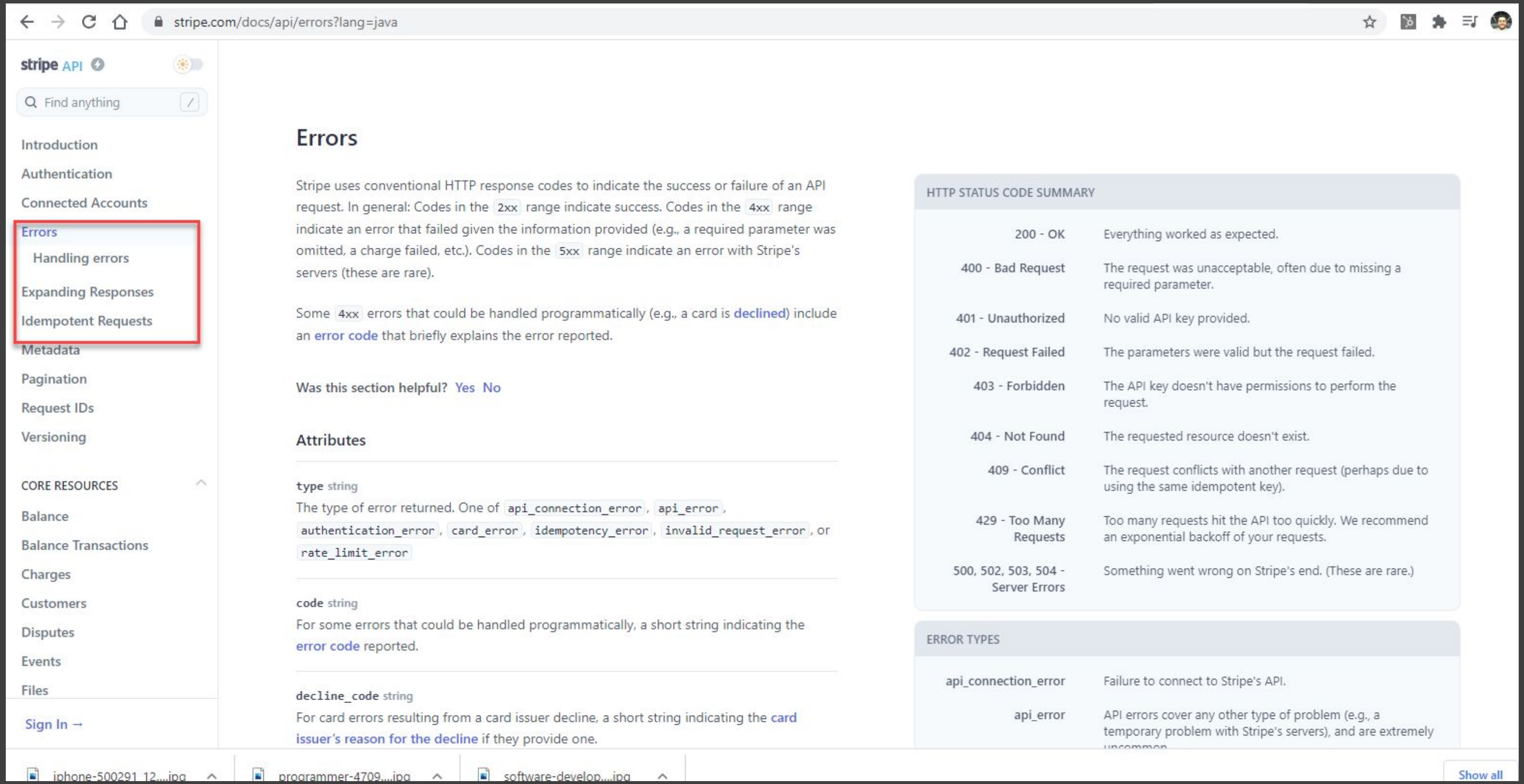
A sample test API key is included in all the examples here, so you can test any example right away.

To test requests using your account, replace the sample API key with your actual API key or sign in.

https://stripe.com/docs/api/authentication?lang=java

Step 2: API Responses

Describe what will the API response look like:



Errors

Stripe uses conventional HTTP response codes to indicate the success or failure of an API request. In general: Codes in the `2xx` range indicate success. Codes in the `4xx` range indicate an error that failed given the information provided (e.g., a required parameter was omitted, a charge failed, etc.). Codes in the `5xx` range indicate an error with Stripe's servers (these are rare).

Some `4xx` errors that could be handled programmatically (e.g., a card is declined) include an error code that briefly explains the error reported.

Was this section helpful? Yes No

Attributes

type string
The type of error returned. One of `api_connection_error`, `api_error`, `authentication_error`, `card_error`, `idempotency_error`, `invalid_request_error`, or `rate_limit_error`

code string
For some errors that could be handled programmatically, a short string indicating the error code reported.

decline_code string
For card errors resulting from a card issuer decline, a short string indicating the card issuer's reason for the decline if they provide one.

HTTP STATUS CODE SUMMARY	
200 - OK	Everything worked as expected.
400 - Bad Request	The request was unacceptable, often due to missing a required parameter.
401 - Unauthorized	No valid API key provided.
402 - Request Failed	The parameters were valid but the request failed.
403 - Forbidden	The API key doesn't have permissions to perform the request.
404 - Not Found	The requested resource doesn't exist.
409 - Conflict	The request conflicts with another request (perhaps due to using the same idempotent key).
429 - Too Many Requests	Too many requests hit the API too quickly. We recommend an exponential backoff of your requests.
500, 502, 503, 504 - Server Errors	Something went wrong on Stripe's end. (These are rare.)

ERROR TYPES	
api_connection_error	Failure to connect to Stripe's API.
api_error	API errors cover any other type of problem (e.g., a temporary problem with Stripe's servers), and are extremely uncommon.

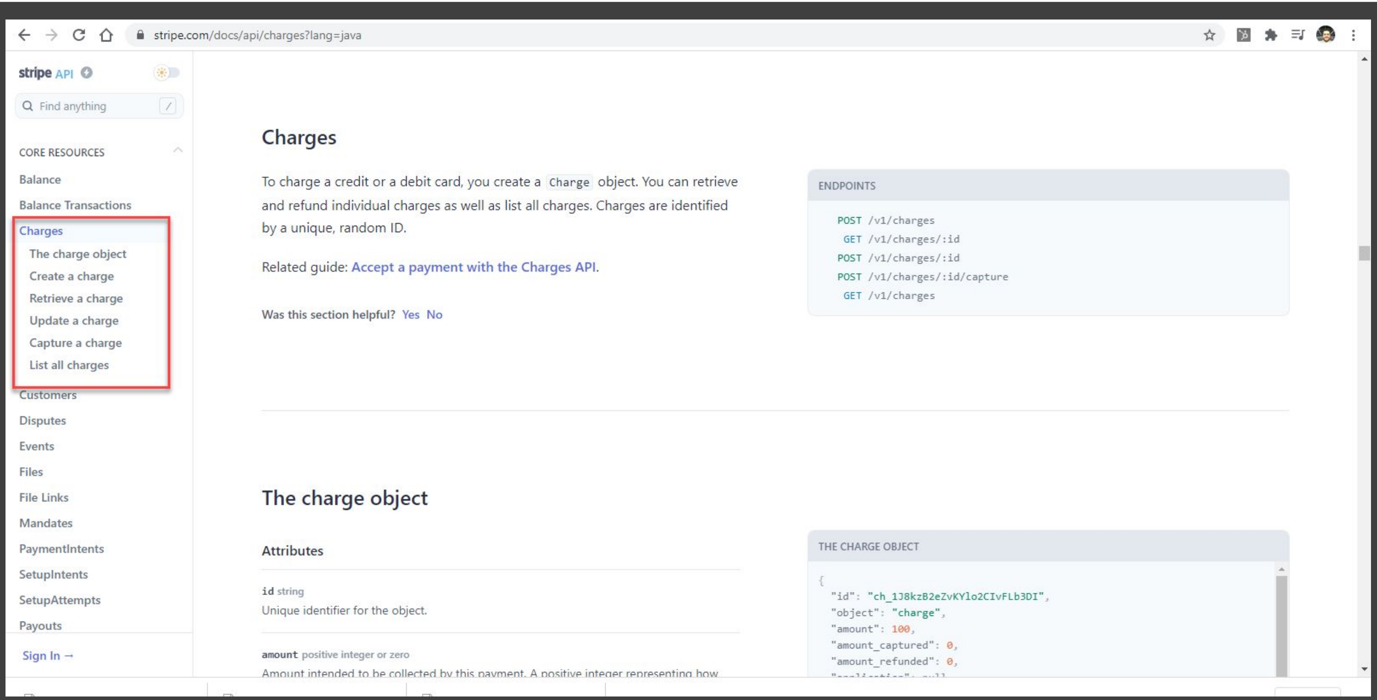
https://stripe.com/docs/api/errors?lang=java

Step 3: Requests

Provide here information about:

Functionality and modules provided via the API

Details about the requests to access the functionality – parameters, syntax, samples.



stripe.com/docs/api/charges?lang=java

stripe API

Find anything

CORE RESOURCES

- Balance
- Balance Transactions
- Charges
  - The charge object
  - Create a charge
  - Retrieve a charge
  - Update a charge
  - Capture a charge
  - List all charges
- Customers
- Disputes
- Events
- Files
- File Links
- Mandates
- PaymentIntents
- SetupIntents
- SetupAttempts
- Payouts

Sign In →

Charges

To charge a credit or a debit card, you create a `Charge` object. You can retrieve and refund individual charges as well as list all charges. Charges are identified by a unique, random ID.

Related guide: Accept a payment with the Charges API.

Was this section helpful? Yes No

ENDPOINTS

```
 POST /v1/charges
  GET /v1/charges/:id
 POST /v1/charges/:id
 POST /v1/charges/:id/capture
  GET /v1/charges
```

The charge object

Attributes

id string
Unique identifier for the object.

amount positive integer or zero
Amount intended to be collected by this payment. A positive integer representing how

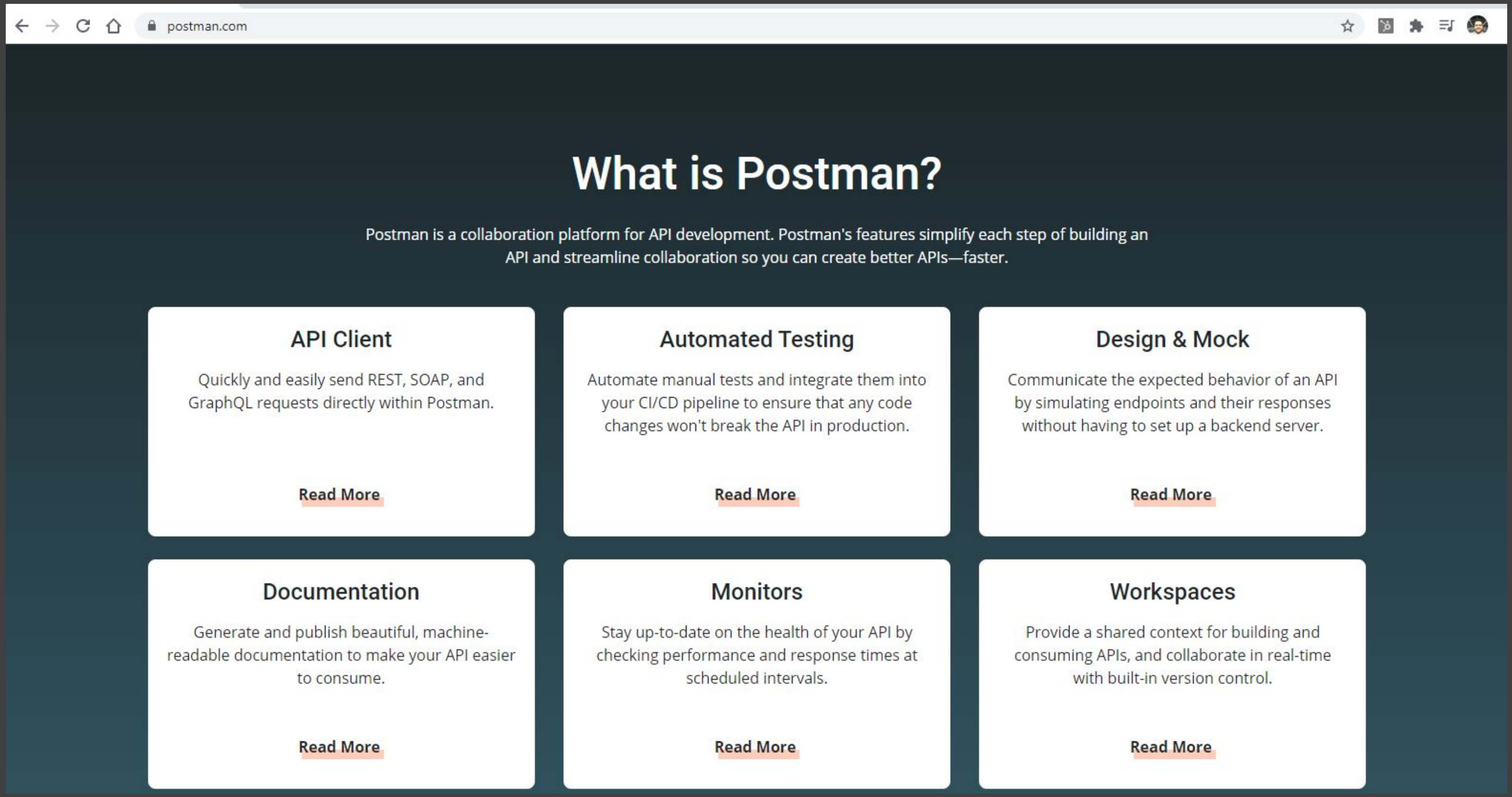
THE CHARGE OBJECT

```
{
  "id": "ch_1J8kzB2eZvKYlo2CIvFLb3DI",
  "object": "charge",
  "amount": 100,
  "amount_captured": 0,
  "amount_refunded": 0,
```

https://stripe.com/docs/api/charges?lang=java

Tools for Using APIs

Which tools to use to **test** what our API calls actually do?



What is Postman?

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster.

API Client

Quickly and easily send REST, SOAP, and GraphQL requests directly within Postman.

Read More

Automated Testing

Automate manual tests and integrate them into your CI/CD pipeline to ensure that any code changes won't break the API in production.

Read More

Design & Mock

Communicate the expected behavior of an API by simulating endpoints and their responses without having to set up a backend server.

Read More

Documentation

Generate and publish beautiful, machine-readable documentation to make your API easier to consume.

Read More

Monitors

Stay up-to-date on the health of your API by checking performance and response times at scheduled intervals.

Read More

Workspaces

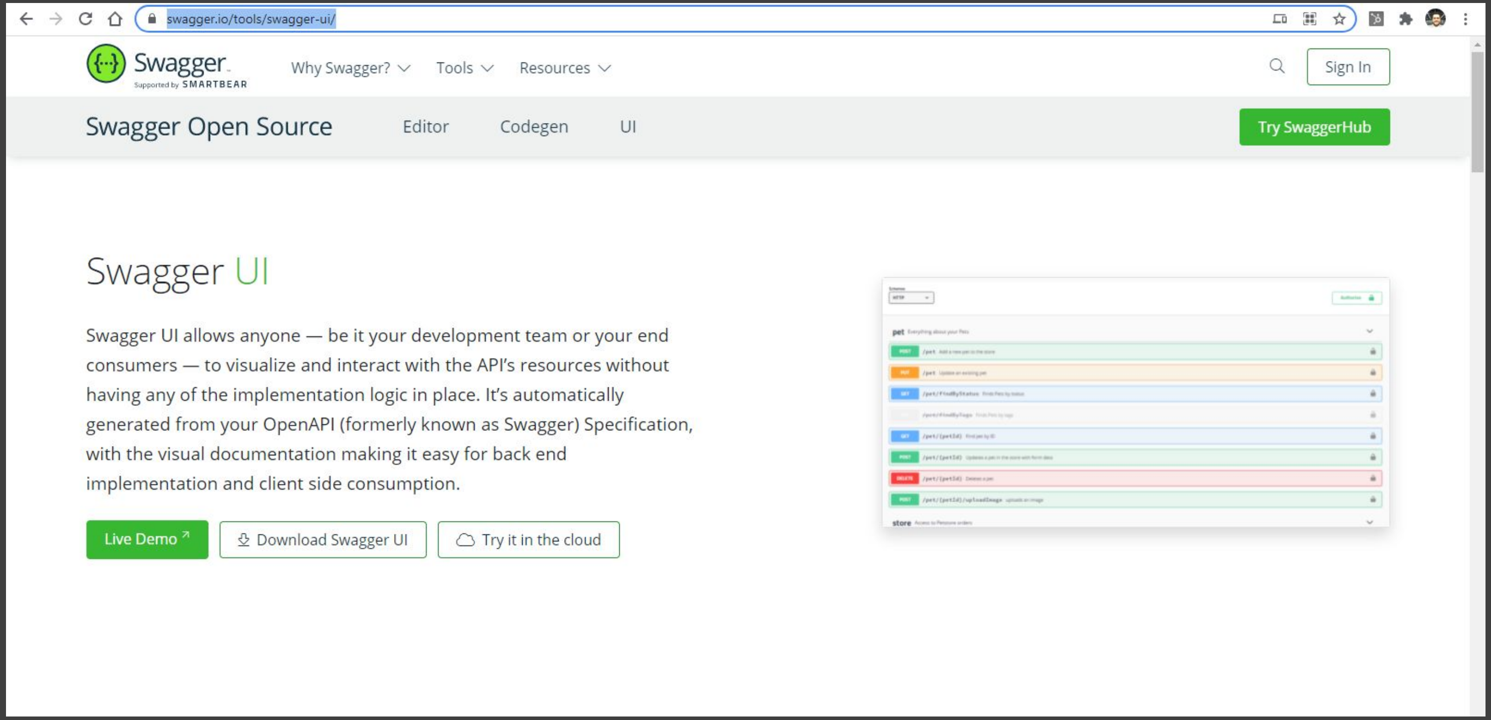
Provide a shared context for building and consuming APIs, and collaborate in real-time with built-in version control.

Read More

Tools for Automated Generation of API Documentation

Which tools to use to **automatically generate** API documentation?

https://app.swaggerhub.com/apis/JPDocu/JPDocu_sample1/1.0.0



Swagger UI

Swagger UI allows anyone — be it your development team or your end consumers — to visualize and interact with the API's resources without having any of the implementation logic in place. It's automatically generated from your OpenAPI (formerly known as Swagger) Specification, with the visual documentation making it easy for back end implementation and client side consumption.

Thank you!

More courses from JPDocu School of Technical Writing:

UX Writing for Technical Writers:

- A Quick Start to Software Documentation
- How to Write Software Documentation
- UX and Information Architecture Basics for the Technical Writer
- Project Management for Technical Writers
- Graphics For Software Documentation
- How to Create Instructional Videos

DITA XML Writing for Technical Writers:

- A Quick Start to Technical Writing with DITA
- Common DITA XML Map and Topic Elements
- How to Write Using DITA XML
- How to Reuse Content in DITA XML
- Linking in DITA XML
- How to Build a DITA XML Technical Writing Portfolio