**Scenario 1 :** You will find a API named Company API under Enrichment API in the following link: [https://clearbit.com/docs#enrichment-api-company-api](https://clearbit.com/docs)

1. What is the purpose of this API? What is your strategy to test it?
2. Prepare the sample input test dataset

Brief Answers to above questions: -

An API (Application Programming Interface) is a set of functions that allows applications to access data and interact with external software components, operating systems, or microservices.

1. The Purpose of the company API is to allow a user to access or look into the company data. Company data comprises of the Elements named as Attributes that are stored in Database by their respective description and data type.

To simplify, Company API delivers a user request and sends the system’s response back to the user. When you search by Domain name, API returns the respective attribute that is present (gives 200 as success response) or else if any attribute is not present, system gives a null value for that attribute.

API lets a developer make a specific “call” or “request” in order to send or receive information. This communication is done using a programming language called “JSON.”  It can also be used to make a defined action such as updating or deleting data. There are four basic request methods that can be made with API:

1. GET – Gathers information (Pulling all Coupon Codes)
2. PUT – Updates pieces of data (Updating Product pricing)
3. POST – Creates (Creating a new Product Category)
4. DELETE – (Deleting a blog post)

**Testing Strategy: -**

The first part of API testing involves setting up a testing environment, with the required set of parameters around the API. This involves configuring the database and server for the application’s requirements. In the above scenario, making sure that all the attributes are present in Database configurations with their respective data types/description.

Types of API testing to be performed on company API:

1. Functionality Testing – API works as per configurations and returns an expected value based on provided input
2. Reliability Testing – Making sure that API is consistently connected and returning consistent results
3. Security testing – Make sure that API has defined security requirements including authentication, permissions and access controls.
4. Negative Testing – Make sure that API returns incorrect responses for incorrect inputs. For example, API should return any one the below Reponses for incorrectly provided inputs
   1. 202 – Asynchronously looking for the company
   2. 404 – Company not found
   3. 422 – Domain name is invalid
5. **Sample Input test data sheet**

|  |  |
| --- | --- |
| **API Call** | **Action** |
| GET /Company | 1. return a 200-status containing the record’s attributes. 2. return a 202 status containing null status |

|  |  |  |
| --- | --- | --- |
| Test Scenario Category | Test Action Category | Test Action Description |
| Basic positive tests (happy paths) | Validate status code: |  |
| Execute API call with valid required parameters |  | 1. All requests should return 2XX HTTP status code |
|  |
| 2. Returned status code is according to spec: |
| – 200 OK for GET requests |
| – 202 null response |
| 404 for company not found 422 for domain name invalid |

|  |
| --- |
|  |