

Detect Language API (Test Strategy & Test Scenario)

CONTENTS

- Introduction
- Test Objectives
- Test Methodology
- Test Scenario and Test Case Preparation
- Automation
- Reporting

1.Introduction

The purpose of this document is to explain the types of testing involved and the corresponding high-level activities (Like Analysis, Test case Preparation, Manual Execution, Automation Development, Reporting, Code Migration).

2.Test Objectives

The main objective is to make sure finding all the observations in Automation and Manual Execution. Most important is to deliver the good quality of the product.

3.Test Methodology

Automation Tool: Ready API (SmartBear)

Language: Groovy

Framework: TDD (Test Driven Development)

4.Test Scenario and Test Case Preparation

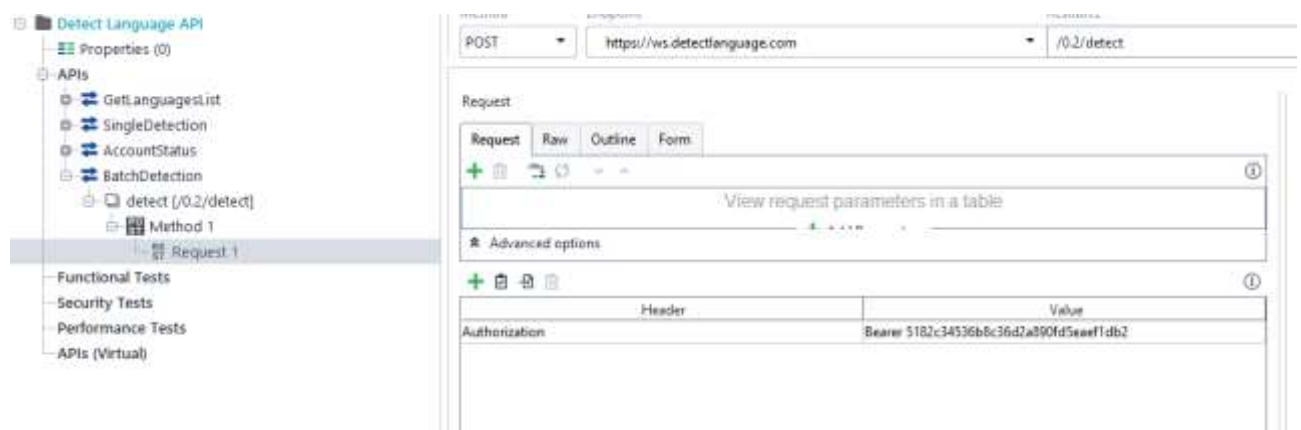
APIs:

Single Detection:

API endpoint: <https://ws.detectlanguage.com/0.2/detect>

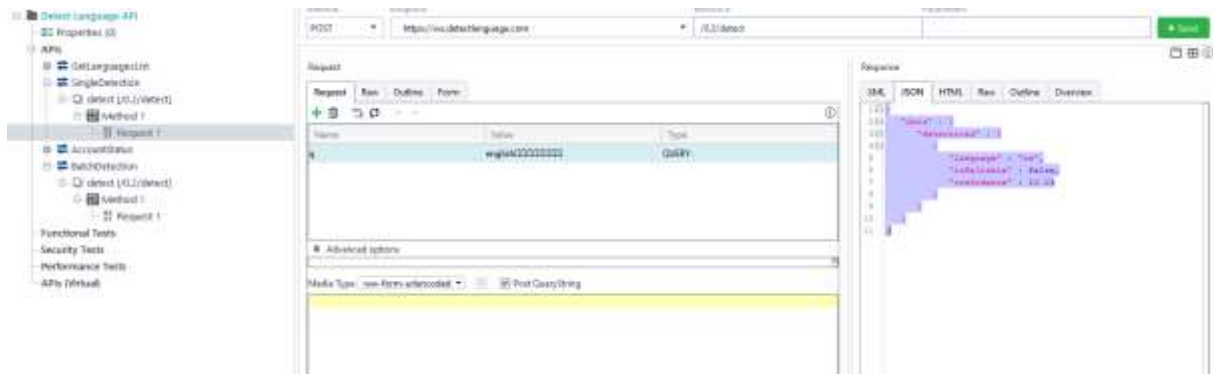
HTTP method: **POST**

Authentication: Under Headers section, added a new row as header key as "Authorization" and the value as "Bearer <your-api-key>" where <your-api-key> is replaced with actual API key.



Request parameters: q - text, mandatory (must be valid UTF-8 encoded string)

1. Under request tab, added a new row with name 'q' and value is the language text which we wanted to verify, and type is query.



Response format: Response contains array of language candidates.

Each result contains following values:

Language code

Confidence score

Is reliable - true/false.

Example Json response as below:

```
{
  "data": {
    "detections": [
      {
        "language": "te",
        "isReliable": false,
        "confidence": 10.24
      }
    ]
  }
}
```

Test Scenarios & Test Cases:

Test Case 1: Valid Request with a single language

Scenario: Sending a valid request with a string in same language.

Input: q: " Hello world"

Expected Output: Response should contain a language code, a confidence score, and a reliability indicator.

Test Case 2: Valid Request with multiple languages

Scenario: Sending a valid request with multiple strings in different languages.

Input: q: "English 喜欢 తెలుగు"

Expected Output: Response should contain language candidates for each text.

Language codes, confidence scores, and reliability indicators should present for each text.

Test Case 3: Empty String in the request parameter

Scenario: Sending an empty string as a request parameter.

Input: q: ""

Expected Output: API should throw an error, response indicating that the input is invalid, or input is missing.

Test Case 4: Invalid Parameter Type (Sending Special characters or numbers)

Scenario: Sending an invalid parameter type (e.g., an integer instead of a string or special characters like +, -, / etc.).

Input: q: 123 or +/-/#

Expected Output: API should throw an error with response indicating that the input is other than text.

Test Case 5: No Parameter Provided

Scenario: Sending the request without the mandatory 'q' parameter.

Input: No 'q' parameter provided in the request.

Expected Output: API should throw an error, response indicating that the 'q' parameter is missing.

Test Case 6: Authentication is missing or wrong Authentication API Key.

Scenario: Sending the request without Authentication or wrong Authentication API Key.

Input: No Authentication or wrong API Key

Expected Output: API should throw an error, response indicating that the Authentication is missing or Wrong Authentication API Key

Batch Detection

API endpoint: <https://ws.detectlanguage.com/0.2/detect>

HTTP method: **POST**

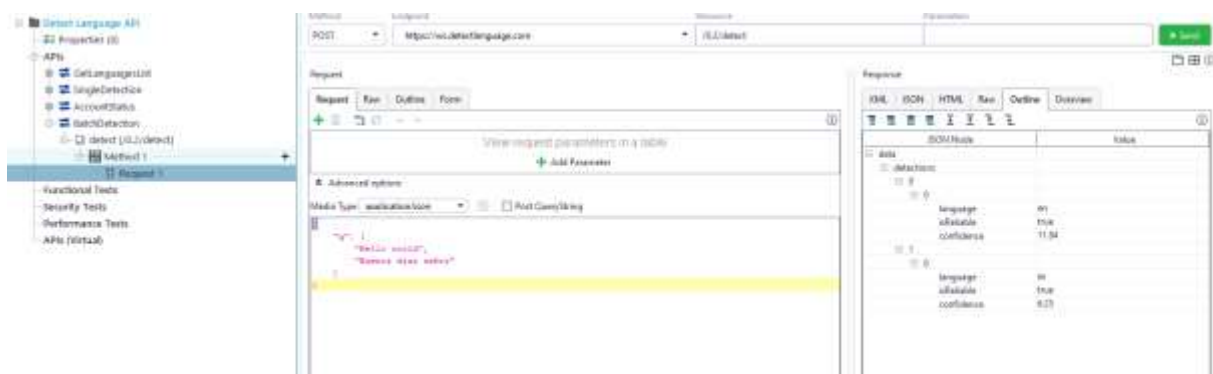
Batch mode allows to detect language of several texts using one query. It saves network bandwidth and significantly increases performance.

Authentication: Under Headers section, added a new row as header key as "Authorization" and the value as "Bearer <your-api-key>" where <your-api-key> is replaced with actual API key.

Request parameters: q - text, mandatory (must be valid UTF-8 encoded string)

1. Passing the request as Json as shown below with array of multiple languages

```
{  
  "q": [  
    "Hello world",  
    "Buenos dias señor"  
  ]  
}
```



Response format: Response contains array of languages based on the request.

Example **Json** response as below:

```
{
  "data": {
    "detections": [
      [
        {
          "language": "en",
          "isReliable": true,
          "confidence": 11.94
        }
      ],
      [
        {
          "language": "es",
          "isReliable": true,
          "confidence": 6.25
        }
      ]
    ]
  }
}
```

Test Scenarios & Test Cases:

Test Case 1: Valid Request with multiple valid languages (Batch)

Scenario: Sending a valid request with multiple strings in different languages (batch).

Input:

```
{
  "q": [
    "Hello world",
    "Buenos dias señor"
  ]
}
```

Expected Output: Response should contain language candidates for each text.

Language codes, confidence scores, and reliability indicators should present for each text.

```
{
  "data": {
    "detections": [
      [
        {
          "language": "en",
          "isReliable": true,
          "confidence": 11.94
        }
      ],
      [
        {
          "language": "es",
          "isReliable": true,
          "confidence": 6.25
        }
      ]
    ]
  }
}
```


Test Case 2: Valid request with languages and numeric values

Scenario: Sending a Valid request with different languages and numeric values

Input:

```
{
  "q": [
    "Hello world",
    "Buenos dias señor",
    "121232"
  ]
}
```

Expected Output: Response should contain language candidates for each text.

Language codes, confidence scores, and reliability indicators should present for each text.

Also, for numeric values it should give a message like 'other than alphabets or not part of any language.'

Test Case 3: Invalid Parameter Type (Sending Special characters or numbers)

Scenario: Sending an invalid parameter type (e.g., an integer instead of a string or special characters like +, -, / etc.).

Input:

```
{
  "q": [
    "12334",
    "#%&"
  ]
}
```

Expected Output: API should throw an error with response indicating that the input is other than text.

5. Automation

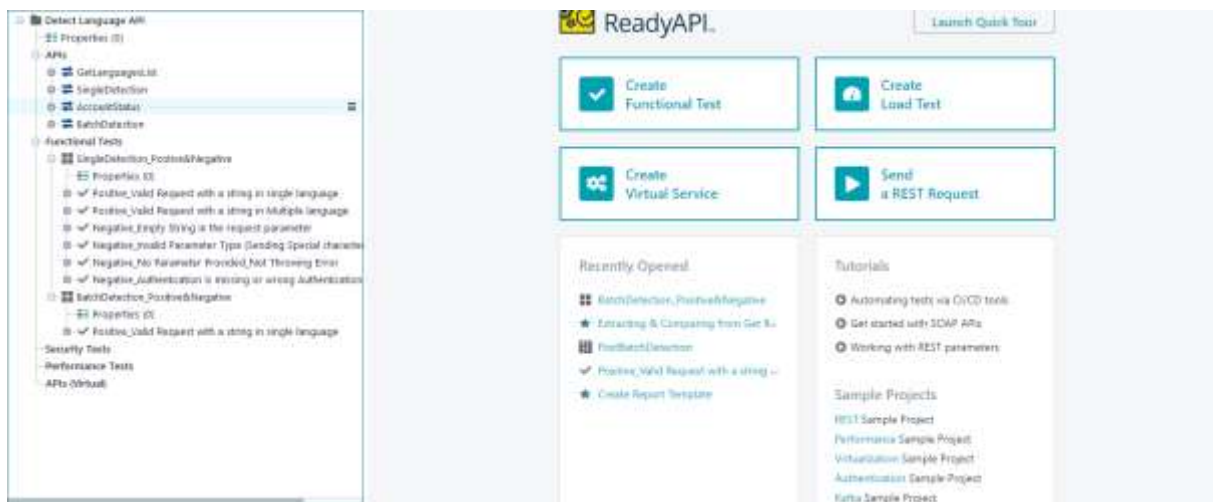
I have used Ready API with Groovy Scripting for Automation. I have used TDD framework as we wanted to test for different test data for the same APIs.

Steps followed in Automation:

1. Configure the given APIs in Ready API under APIs section.



2. Create Test Suite and Test Cases under Functional Tests



3. Now under each test, we have followed below steps:
 - a. Create Template excel report which is used to see the output after the execution. Report can be modified based on request.
 - b. Data Source, which is used to provide test data, test data can be provided in different forms like grid or excel providing with path.
 - c. We can use groovy scripting, or we can fetch test data from before step and pass it to the service.
 - d. Post or Get service which we need to hit for the given test data.
 - e. Here is the final step and most important step, fetching all the data from test data and response of the service and comparing the HTTP codes, data, values etc.

6. Reporting

Test Results will be uploaded for different scenarios and Test cases. Below are the high-level observations.

Single Detection:

Green Color is working as expected and Red color is observations

Test Case 1: Valid Request with a single language -- Getting correct response code and expected response with Language code, Confidence score, Is reliable.

Test Case 2: Valid Request with multiple languages -- Getting correct response code and expected response with Language code, Confidence score, Is reliable.

Test Case 3: Empty String in the request parameter – Getting 200Ok response without having Language code, Confidence score, Is reliable. As mentioned above under detailed test scenarios, it should throw an error indicating that the input is invalid, or input is missing.

Test Case 4: Invalid Parameter Type (Sending Special characters or numbers) -- Getting 200Ok response without having Language code, Confidence score, Is reliable. As mentioned above under detailed test scenarios, it should throw an error indicating that the input is other than text.

Test Case 5: No Parameter Provided -- Getting 200Ok response without having Language code, Confidence score, Is reliable. As mentioned above under detailed test scenarios, it should throw an error indicating that the 'q' parameter is missing.

Test Case 6: Authentication is missing or wrong Authentication API Key. – Getting 401 Unauthorized with message invalid API Key.

Batch Detection:

Green Color is working as expected and red color is observations.

Test Case 1: Valid Request with multiple valid languages (Batch) -- Getting correct response code and expected response with Language code, Confidence score, Is reliable.

Test Case 2: Valid request with languages and numeric values -- Getting correct response code and expected response with Language code, Confidence score, is reliable but for numeric value response doesn't have Language code, Confidence score, Is reliable.

Test Case 3: Invalid Parameter Type (Sending Special characters or numbers) -- Getting 200Ok response without having Language code, Confidence score, Is reliable. As mentioned above under detailed test scenarios, it should throw an error indicating that the input is other than text.