# REQUIREMENTS

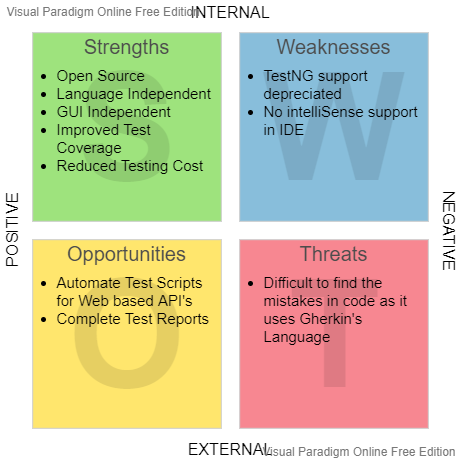
# Introduction

* The Project – Test Automation Framework Development for Webservice API using KARATE Framework
* Karate is the only open-source tool to combine API test-automation, mocks, performance-testing and even UI automation into a **single**, unifiedframework.
* The BDD syntax popularized by Cucumber is language-neutral, and easy for even non-programmers. Assertions and HTML reports are built-in, and you can run tests in parallel for speed.
* There's also a cross-platform stand-alone executable for teams not comfortable with Java. You don't have to compile code. Just write tests in a simple, readable syntax - carefully designed for HTTP, JSON, Graph QL and XML. And you can mix API and UI-test automation within the same test script.
* A JAVA API also exists for those who prefer to programmatically integrate Karate's rich automation and data-assertion capabilities.

1. **PROJECT SCOPE/OBJECTIVE**

* Created Test Automation Framework Development for Web service API using Karate Framework using Gherkin's language.
  + Open source
  + Parallel Execution
  + BDD Style, Native JSON/XML support
  + Readable Syntax
  + Reporting
  + Performance Testing
  + Mix API and UI test-automation
* Created the Test Report of all the Scenario’s

# SWOT ANALYSIS



# 4W's and 1'H

# Who: To all API’s for the automated Test Scripts.

# What: It is a Karate Framework which consists of Automated Test Scripts for testing the API’s

**When**: During API testing

**Where**: For Automated Test Scripts to test Web based API’s

**How**: It uses Gherkin’s language which is supported by the Karate Framework and all the Feature files was run using Junit for the API testing in different scenarios.

1. **AGING**

With Technologies rapidly increasing, Features of API testing has been improved as now Parallel Execution of multiple Feature file is feasible as well as Performance testing can be tested for multiple user threads.

1. **RADING COST**

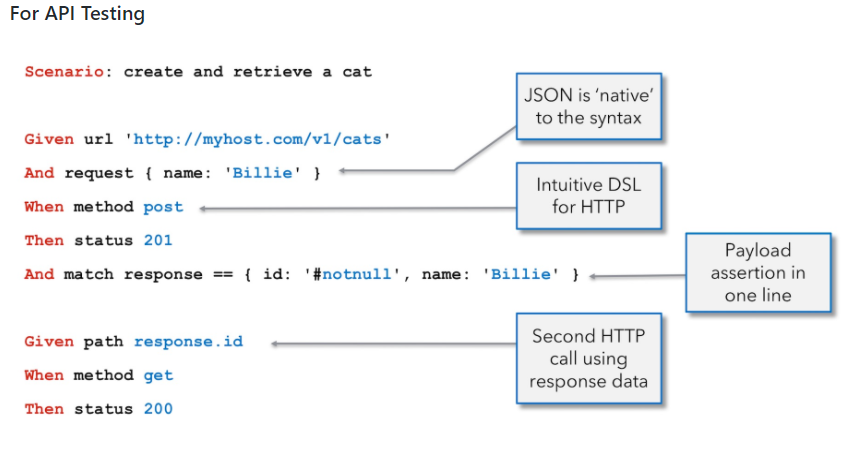
**Cost and Features**

1. Coding and Complexity

2. Maintenance and Resources

3. Database maintenance for web based API’s

1. **Defining the System**



1. **HIGH- AND LOW-LEVEL REQUIREMENTS**

