

# RESTful API based Online Compiler (TINKER)

Aryan Anmol

Department of Computer  
Science

G L Bajaj Institute of  
Technology and Management

Abhishek Singhal

Department of Computer  
Science

G L Bajaj Institute of  
Technology and Management

Abhishek Kumar

Department of Computer  
Science

Shambhunath Institute of  
Engineering and Technology

## ABSTRACT

The Project TINKER aims towards providing a common endpoint for compiling and execution of programs written in different languages as well as providing the results/outcomes of those programs. These programs can be in C, C++, Java, Python, and PHP. The result will be in JSON format containing the output of the program, its execution time, memory and errors (if any) consumed along with a unique ID. This ID can be used to access the information about previously submitted program at any point of time which will include the program itself, input, language, the same output when the program was first executed and a timestamp, since each and every submission is stored with Google Firebase database. Google Firebase was chosen over the traditional real time database due to its high reliability. These services can be utilized by any one by making HTTP requests on our project (TINKER) endpoints.

## General Terms

Online Compiler RESTful API REST and SOAP

## Keywords

Restful API, API Technology, Online Code Evaluator, Laravel Framework

## 1. INTRODUCTION

The project “TINKER” is a REST API based online code evaluation system would make it easier to compile and run code. It requires input parameters as a POST request which will return the status about the code. The status will include various responses about the code which can be further used to verify the code. This service could be seamlessly embedded with multiple user interface like mobile application or a web based application. This application provides security for code by using encryption and decryption algorithm. This API enables the user to create custom test on multiple programming languages.

## 2. RESTful API

Representational state transfer (REST) or RESTful web services is a way of providing interoperability between computer systems on the Internet. REST-compliant Web services allow requesting systems to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations REST is a technical specification about heterogeneous inter-communication for Web applications. Adoption of REST can lead to a simple, scalable, effective, safe and reliable architecture. REST is a lightweight RPC protocol built upon HTTP protocol, its simplicity and web friendliness makes it an unparalleled alternative to SOAP, a popular RPC solution for years. A well-structured Web application can be easily built when using Rest, Many developers successfully created simple and robust API base Restful web service. The term

representational state transfer was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation. Fielding used REST to design HTTP and Uniform Resource

Identifiers (URI). A resource is a kind of information that can be accessed, such as an application object, a database record, an algorithm, and so on. Each resource is identified by a unique URI (Universal Resource Identifier), REST represent URI in the form of "/user/name", and operations on HTTP methods GET, PUT, POST, DELETE, HEADER and OPTIONS, resulting in the next resource being transferred back to the caller. An important characteristic of REST is that the server side keeps stateless between multiple interactions, every server in the clusters can serve the client on every request.

## 3. SOFTWARE DESIGN

A remote machine is needed which would act as a server for hosting the web application and the compilers for the execution of programs. The machine would need to have the following dependencies installed

1. Compiler – GCC, Javac, Python, php
2. Apache2 web server
3. PHP v7.2.6 for running web application

Configure the remote machine to receive the API requests and handle them. To handle the requests, we are using Laravel Framework (based on PHP language). It accepts requests on the following two endpoints.

1. POST /API
2. GET /API/{id}

All the submissions sent through the POST request are stored with Google Firebase’s Database. Instead of using traditional relational database, Google Firebase database is chosen since it is highly reliable. A submission is retrieved when a GET request is sent to our application with a ID parameter in the URI. This ID is used to fetch the submission from the database and all the details relevant to it are retrieved. The response of both the requests are sent back in JSON format as its easily readable by human eye and can be parsed by any language on any platform.

## 3.1 Working Model

### 3.1.1 COMPILER:

To execute a particular program and generate the output we need to have compilers of the languages that we need to be installed on a machine which will act as our remote server. The most common compilers are GCC for C & C++ and javac for java. These compilers will be installed on our ubuntu based remote machine/server.

### 3.1.2 LARAVEL:

Laravel is a PHP framework that follows model-view-controller architectural pattern. When the client sends a

request to our web application, its handled by the Laravel based on the routing and endpoints defined on the basis of REST API.

Our app consists of two endpoints

POST /api

FIRST END POINT <https://www.tinker.tk/api>

GET /api/{id}

SECOND END POINT <https://www.tinker.tk/api/{id}>

### 3.2 JASON Response

JSON is used to display the response in a structured manner which needs to be used by another application. JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3<sup>rd</sup> Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others.

## 4 RESTful API Documentation

TINKER API is a free API which compiles the source code provided by the user in the cloud and returns the output. Currently this API supports 5 programming languages.

No authorization key is required to access this API. Anyone can directly access the API at the endpoints as given in the documentation. Go through the documentation given below to get familiar with the API. All URLs referenced in the documentation have the following base:

<https://www.tinker.tk/api>

### 4.1 Functionality

Tinker API allows you to do the following actions:

- Upload source code.
- Run the program with input data on server side in 5 programming languages.
- Download results of the execution at any point of time.

Supported Programing languages are

**Table 1**

Programing Language	Compiler
C language	GCC
C++	CPP
Python 2	Python2
Python 3	Python3
JAVA	Javac
PHP	php

### 4.1.1 Query Parameters

**Table 2**

Parameter	Description
Source	Source code need to be compiled
Lang	Language identifier. It can be retrieved from the above table
Stdin	Data that will be given to the program on stdin

### 4.1.2 Return Value

**Table 3**

Parameter	Description
Key	Unique Key is generated on each submission to track response later
Status	Submission current status
Result	Submission current result
Any_Cmperr	Weather any compiler error or not(True/False)
Stdout	Output produced by the program
Cmperr	Compilation error information for the program
Time	Timestamp of the execution
Memory	Memory used by the program in kB
Stderr	Error produced by the program

### 4.1.3 Status

**Table 4**

Value	Meaning
0	Done-the program has finished execution
1	Compilation-the program is being compiled
2	Program Execution

### 4.1.4 Result

**Table 5**

Value	Meaning
11	Compilation error-the program could not be executed due to compilation error
12	Runtime error
13	Time limit exceeded

15	Success
17	Memory limit exceeded
19	Unauthorized system call
20	Internal Error

```
{
  "-LBHhXondR1n0EtcE7Jq":
  {
    "any_cmperr":false,
    "cmperr": "",
    "language": "c",
    "memory": "4416",
    "result": "15",
    "sourceCode": "#include<stdio.h>\n\nint main() {\n\tint a, b,c;\n\tscanf("%d%d%d\\",\n\t\t&a,&b,&c);\n\tprintf("%d\\", a+b+c);\n\treturn 0;\n}",
    "status": "0",
    "stderr": "",
    "stdin": "2\n3\n5",
    "stdout": "10",
    "time": "0",
    "timestamp": "2018-04-29 21:20:04"
  }
}
```

Fig 1 POST response

```
{
  "key": "-LBHhXondR1n0EtcE7Jq",
  "status": "0",
  "result": "15",
  "any_cmperr": false,
  "stdout": "15",
  "cmperr": "",
  "stderr": "",
  "time": "0",
  "memory": "4416"
}
```

Fig 2 GET Response

#### 4.2.1 Get old Submission

HTTP Request

GET api/{key}

Key –It is a unique string that is return on every new submission

#### 4.2.2 Return Value

Table 6

Parameter	Description
-----------	-------------

Any_cmprr	Weather there is a compiler error or not (True /False)
Cmperr	Compilation error information
Language	Language Identifier
Memory	Memory used by the program in kB
Result	Submission result
Source code	Source code to be compiled
Status	Submission current status
Stdin	Data to be given to the program
Stdout	Output produced by the program
Time	Execution time in seconds
Time Stamp	Time at which submission is made

## 4. CONCLUSION

The RESTful API based online cloud compiler would enable teachers to built customized testing platform as per the need and requirement. This API could also be used in organizing hackathons or developing a mobile application that teaches programming to anyone who wants to learn and play with computer code.

## 5. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

## 6. REFERENCES

- [1]"Web Services Architecture". World Wide Web Consortium. 11 February 2004. 3.1.3 Relationship to the World Wide Web and REST Architectures. Retrieved 29 September 2016.
- [2] Cesare Pautasso, Olaf Zimmermann, Frank Leymann, "RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision" WWW 2008, April 21–25, 2008, Beijing, China. ACM 978-1-60558-085-2/08/04.
- [3] Fielding, Roy Thomas (2000). "Chapter 5: Representational State Transfer (REST)".
- [4] "Fielding discussing the definition of the REST term". groups.yahoo.com. Retrieved 2017-08-08.