Tecnológico de Monterrey Campus Santa Fe

API

Gabriel Rodríguez De Los Reyes A01027384

Mariel Gómez Gutiérrez A01275607

Pablo Banzo Prida A01782031

Santiago Rodriguez Palomo A01025232

Grupo 501

Final Assessment

June 6th, 2025

This document provides comprehensive information about the Application Programming Interface (API) implemented for the Compilo platform. Each endpoint serves a specific function and is detailed in the sections below.

signUp

Service name	signUp()
Description	Creates a user after receiving the user data.
HTTP Method	Post
Input	{"username": "test%dusername", "password": "testpassword", "email": "test%d@email.com"}
Output	{"message":"User created successfully","token": "..." }

logIn

Service name	logIn()
Description	Logs the user into the platform.
HTTP Method	Post
Input	{"username": "testuser", "password": "testpassword"}
Output	{"message":"Login successful","token": "..." }

problems

Service name	problems()
Description	Used to get the problems, and show all of its attributes
HTTP Method	Get
Input	No request body (requires authentication token in header)
Output	[{ "id": "problem_id", "title": "Problem Title", "description": "Problem description with examples", "difficulty": "easy|medium|hard", "test_cases": [

	{
	"input": "test input",
	"output": "expected output"
	}
	],
	"function_name": "functionName",
	"arguments": [
	{
	"name": "param1",
	"type": "int|string|float"
	}
	],
	"created_at": "2024-01-01T00:00:00Z",
	"updated_at": "2024-01-01T00:00:00Z"
	}
	]

problems/{id}

Service name	problems/{id}
Description	Get specific problems and show all of its attributes.
HTTP Method	Get
Input	No request body (requires authentication token in header)

Output	{
	"id": "problem_id",
	"title": "Problem Title",
	"description": "Problem description with examples",
	"difficulty": "easy|medium|hard",
	"test_cases": [
	{
	"input": "test input",
	"output": "expected output"
	}
	],
	"function_name": "functionName",
	"arguments": [
	{
	"name": "param1",
	"type": "int|string|float"
	}
	],
	"created_at": "2024-01-01T00:00:00Z",

	"updated_at": "2024-01-01T00:00:00Z" }

compile

Service name	compile()
Description	Compiles and executes user-submitted code against predefined test cases for a specific programming problem.
HTTP Method	Post
Input	{"problemId": "problem_id", "code": "function code here"}

Output	{
	  "result": [
	    {
	      "status": "Success|Failed",
	      "output": [123],
	      "expectedOutput": [123]
	    }
	  ],
	  "error": "compilation error message if any",
	  "status": "Success|Error",
	  "line": 0,
	  "column": 0
	}

logs

Service name	logs()
Description	Gets the user’s logs.
HTTP Method	Get
Input	No request body (requires authentication token in header)

Output	[
	  {
	    "id": "log_id",
	    "user_id": "user_id",
	    "method": "POST|GET",
	    "path": "/endpoint/path",
	    "status": 200,

	"duration": "500ms", "body": "request body", "response_body": "response body", "case": "problem_id", "ip": "127.0.0.1", "created_at": "2024-01-01T00:00:00Z" }]

problems/{id}/solutions

Service name	problems/{id}/solutions
Description	Gets an specific solution for a given problem submitted by the user.
HTTP Method	Get
Input	No request body (requires authentication token in header)

Output	{
	"success": true,
	"solutions": [
	{
	"id": "solution_id",
	"problemId": "problem_id",
	"userId": "user_id",
	"code": "submitted code",
	"status": "passed|failed|partial",
	"submittedAt": "2024-01-01T00:00:00Z",
	"executionTime": "100ms"
	}
	],
	"totalSolutions": 5
	}

allsolutions

Service name	allsolutions()
Description	Gets all of the solutions submitted by the user for a given problem.
HTTP Method	Get
Input	No request body (requires authentication token in header)
Output	{

	"success": true,
	"solutions": [
	{
	"id": "solution_id",
	"problemId": "problem_id",
	"userId": "user_id",
	"code": "submitted code",
	"status": "passed|failed|partial",
	"submittedAt": "2024-01-01T00:00:00Z",
	"executionTime": "100ms"
	}
	],
	"totalSolutions": 15
	}

runCompile

Service name	runCompile()
Description	Executes a complete program with provided input values and returns the program's output along with execution status.
HTTP Method	Post

Input	{
	"program": "void main(void) {\nint x;int y;\nx = input(); y = input();\nwhile (x < 5) {\noutput(x * y);\nx = x + 1;\n}\n}",
	"inputs": ["0", "2"]
	}

Output	{
	"column": -1,
	"error": "",
	"line": -1,
	"message": "Program executed successfully",
	"outputs": [
	0,
	2,
	4,
	6,
	8
	]
	}

performTestCases

Service name	performTestCases()
Description	Executes a specific function within a program against multiple test cases and returns individual results for each test.
HTTP Method	Post

Input	{
	  "program": "int addTwo(int a, int b) {\nint y;\ny = a + b;\nreturn y;\n\n}",
	  "funName": "addTwo",
	  "testCases": [
	    [
	      5,
	      3
	    ],
	    [
	      10,
	      20
	    ],
	    [
	      -5,
	      15
	    ]
	  ]
	}

Output	{
	  "results": [
	    {
	      "column": -1,
	      "error": "",
	      "line": -1,
	      "output": 8
	    },
	    {
	      "column": -1,
	      "error": "",
	      "line": -1,
	      "output": 30
	    },
	    {
	      "column": -1,
	      "error": "",
	      "line": -1,
	      "output": 10
	    }
	  ]

	}

checkSyntax

Service name	checkSyntax()
Description	Validates the syntax of the submitted program.
HTTP Method	Post

Input	{
	"program": "void main(void) {\nint x;\nx = 5;\noutput(x);\n}"
	}

Output	{
	"column": -1,
	"error": "",
	"isSyntaxCorrect": true,
	"line": -1,
	"message": ""
	}