

# C++:

## Example 1:

```
restful_API_Call > C++ rest.cpp > main()
1  #include "splashkit.h"
2  using std::to_string;
3
4  int main()
5  {
6      http_response get_data;
7      string response;
8
9      // Get a single JSON web resource
10     get_data = http_get("https://jsonplaceholder.typicode.com/posts/1", 443);
11     response = http_response_to_string(get_data);
12     free_response(get_data);
13     // Output the response
14     json json_response = json_from_string(response);
15     write_line("UserID => " + std::to_string(json_read_number_as_int(json_response, "userId")));
16     write_line("ID    => " + std::to_string(json_read_number_as_int(json_response, "id")));
17     write_line("Title  => " + json_read_string(json_response, "title"));
18     write_line("Body   => " + json_read_string(json_response, "body"));
19     write_line("=====");
20
21     return 0;
22 }
```

PROBLEMS OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE

```
• breezy breezy ~/restful_API_Call main basic g++ rest.cpp -l SplashKit -o rest
• breezy breezy ~/restful_API_Call main basic ./rest
UserID => 1
ID    => 1
Title  => sunt aut facere repellat provident occaecati excepturi optio reprehenderit
Body   => quia et suscipit
suscipit recusandae consequuntur expedita et cum
reprehenderit molestiae ut ut quas totam
nostrum rerum est autem sunt rem eveniet architecto
=====
• breezy breezy ~/restful_API_Call main basic
```

## Example 2:

```
restful_API_Call > C++ rest-ex2.cpp > main()
1  #include "splashkit.h"
2
3  int main()
4  {
5      http_response get_data;
6      string response;
7
8      // Get a List of a JSON web resource
9      get_data = http_get("https://jsonplaceholder.typicode.com/posts/", 443);
10     response = http_response_to_string(get_data);
11     free_response(get_data);
12     // To access each JSON key value pair the string should be split to an vector<json>
13     // objects for simplicity sake we output just the string here.
14     write_line(response);
15
16     return 0;
17 }
```

PROBLEMS OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE

```
• breezy breezy ~/restful_API_Call main basic g++ rest-ex2.cpp -l SplashKit -o rest-ex2
• breezy breezy ~/restful_API_Call main basic ./rest-ex2
[
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
  },
  {
    "userId": 1,
    "id": 2,
    "title": "qui est esse",
    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
  },
  {
    "userId": 1,
    "id": 3,
    "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",

```

### Example 3:

```
restful_API_Call > C++ rest-ex3.cpp > main()
1  #include "splashkit.h"
2
3  int main()
4  {
5      http_response get_data;
6      string response;
7
8      // Create a JSON Web resource
9      // Create the JSON body for the http post call
10     json_body = create_json();
11     json_set_string(body, "title", "foo");
12     json_set_string(body, "body", "test entry");
13     json_set_number(body, "userId", 1);
14     // Create the header for the http post call
15     vector<string> headers;
16     headers.push_back("Content-type: application/json; charset=UTF-8");
17     // Send the request
18     get_data = http_post("https://jsonplaceholder.typicode.com/posts", 443, json_to_string(body), headers);
19     response = http_response_to_string(get_data);
20     free_response(get_data);
21     // Output the response
22     json json_response = json_from_string(response);
23     write_line("Following resource has been created");
24     write_line("UserID => " + std::to_string(json_read_number_as_int(json_response, "userId")));
25     write_line("ID      => " + std::to_string(json_read_number_as_int(json_response, "id")));
26     write_line("Title   => " + json_read_string(json_response, "title"));
27     write_line("Body    => " + json_read_string(json_response, "body"));
28     write_line("=====");
29
30     return 0;

```

PROBLEMS OUTPUT **TERMINAL** PORTS COMMENTS DEBUG CONSOLE

```
• breezy breezy ~/restful_API_Call main o basic g++ rest-ex3.cpp -l SplashKit -o rest-ex3
• breezy breezy ~/restful_API_Call main o basic ./rest-ex3
Following
resource has been created
UserID => 1
ID      => 101
Title   => foo
Body    => test entry
=====
o breezy breezy ~/restful_API_Call main o basic
```

## Python:

### Example 1:

```
restful_API_Call > Python rest-ex1.py > ...
1  from splashkit import *
2
3  # Get a single JSON web resource
4  get_data = http_get("https://jsonplaceholder.typicode.com/posts/1", 443)
5  response = http_response_to_string(get_data)
6  free_response(get_data)
7
8  # Parse the response as JSON
9  json_response = json_from_string(response)
10
11 # Output the response
12 print("UserID => " + str(json_read_number_as_int(json_response, "userId")))
13 print("ID      => " + str(json_read_number_as_int(json_response, "id")))
14 print("Title   => " + json_read_string(json_response, "title"))
15 print("Body    => " + json_read_string(json_response, "body"))
16 print("=====")
17
18 # Free the JSON object after use
19 free_json(json_response)
20

```

PROBLEMS 30 OUTPUT **TERMINAL** PORTS COMMENTS DEBUG CONSOLE

```
• breezy breezy ~/restful_API_Call main o basic skm python3 rest-ex1.py
UserID => 1
ID      => 1
Title   => sunt aut facere repellat provident occaecati excepturi optio reprehenderit
Body    => quia et suscipit
suscipit recusandae consequuntur expedita et cum
reprehenderit molestiae ut ut quas totam
nostrum rerum est autem sunt rem eveniet architecto
=====
🚀 python3 command run successfully 🚀
• breezy breezy ~/restful_API_Call main o basic
```

## Example 2:

```
restful_API_Call > 🐡 rest-ex2.py > ...
1  from splashkit import *
2
3  # Get a list of a JSON web resource
4  get_data = http_get("https://jsonplaceholder.typicode.com/posts/", 443)
5  response = http_response_to_string(get_data)
6  free_response(get_data)
7
8  # Output the response
9  print(response)
10
```

PROBLEMS 24 OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE

🐡 breezy breezy ~/restful\_API\_Call main ○ basic skm python3 rest-ex2.py

```
[
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum re
rum est autem sunt rem eveniet architecto"
  },
  {
    "userId": 1,
    "id": 2,
    "title": "qui est esse",
    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro
```

## Example 3:

```
restful_API_Call > 🐡 rest-ex3.py > ...
1  from splashkit import *
2
3  # Create a JSON Web resource
4  # Create the JSON body for the HTTP POST call
5  body = create_json()
6  json_set_string(body, "title", "foo")
7  json_set_string(body, "body", "test entry")
8  json_set_number_integer(body, "userId", 1)
9
10 # Create the header for the HTTP POST call
11 headers = ["Content-type: application/json; charset=UTF-8"]
12
13 # Send the request
14 get_data = http_post_with_headers("https://jsonplaceholder.typicode.com/posts", 443, json_to_string(body), headers)
15 response = http_response_to_string(get_data)
16 free_response(get_data)
17
18 # Output the response
19 json_response = json_from_string(response)
20 print("Following resource has been created")
21 print("UserID => " + str(json_read_number_as_int(json_response, "userId")))
22 print("ID      => " + str(json_read_number_as_int(json_response, "id")))
23 print("Title  => " + json_read_string(json_response, "title"))
24 print("Body   => " + json_read_string(json_response, "body"))
25 print("=====")
```

PROBLEMS 28 OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE

🐡 breezy breezy ~/restful\_API\_Call main ○ basic skm python3 rest-ex3.py

```
Following resource has been created
UserID => 1
ID      => 101
Title  => foo
Body   => test entry
=====
🐡 python3 command run successfully 🐡
```

🐡 breezy breezy ~/restful\_API\_Call main ○ basic

# C#:

## Example 1:

```
restful_API_Call > ex1 > C# Program.cs
1  using SplashKitSDK;
2
3  // Get a single JSON web resource
4  HttpResponseMessage getData = SplashKit.HttpGet("https://jsonplaceholder.typicode.com/posts/1", 443);
5  string response = SplashKit.HttpResponseToString(getData);
6  SplashKit.FreeResponse(getData);
7
8  // Output the response
9  Json jsonResponse = SplashKit.JsonFromString(response);
10 SplashKit.WriteLine("UserID => " + SplashKit.JsonReadNumberAsInt(jsonResponse, "userId"));
11 SplashKit.WriteLine("ID => " + SplashKit.JsonReadNumberAsInt(jsonResponse, "id"));
12 SplashKit.WriteLine("Title => " + SplashKit.JsonReadString(jsonResponse, "title"));
13 SplashKit.WriteLine("Body => " + SplashKit.JsonReadString(jsonResponse, "body"));
14 SplashKit.WriteLine("=====");
15
16 | SplashKit.FreeJson(jsonResponse);
```

PROBLEMS OUTPUT **TERMINAL** PORTS COMMENTS DEBUG CONSOLE

```
• breezy breezy ~/restful_API_Call/ex1 main basic dotnet run build
UserID => 1
ID => 1
Title => sunt aut facere repellat provident occaecati excepturi optio reprehenderit
Body => quia et suscipit
suscipit recusandae consequuntur expedita et cum
reprehenderit molestiae ut ut quas totam
nostrum rerum est autem sunt rem eveniet architecto
=====
```

## Example 2:

```
restful_API_Call > ex2 > C# Program.cs
1  using SplashKitSDK;
2
3  // Get a list of a JSON web resource
4  HttpResponseMessage getData = SplashKit.HttpGet("https://jsonplaceholder.typicode.com/posts/", 443);
5  string response = SplashKit.HttpResponseToString(getData);
6  SplashKit.FreeResponse(getData);
7
8  // Output the response
9  | SplashKit.WriteLine(response);
```

PROBLEMS OUTPUT **TERMINAL** PORTS COMMENTS DEBUG CONSOLE

```
• breezy breezy ~/restful_API_Call/ex2 main basic dotnet run build
[
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
  },
  {
    "userId": 1,
    "id": 2,
    "title": "qui est esse",
    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis volup
tate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
  }
],
```

### Example 3:

```
restful_API_Call > ex3 > C# Program.cs
1  using SplashKitSDK;
2
3  // Create a JSON Web resource
4  // Create the JSON body for the http post call
5  Json body = SplashKit.CreateJson();
6  SplashKit.JsonSetString(body, "title", "foo");
7  SplashKit.JsonSetString(body, "body", "test entry");
8  SplashKit.JsonSetNumber(body, "userId", 1);
9
10 // Create the headers for the HTTP POST call
11 List<string> headers = new List<string>
12 {
13     "Content-type: application/json; charset=UTF-8"
14 };
15
16 // Send the request
17 HttpResponseMessage getData = SplashKit.HttpPost("https://jsonplaceholder.typicode.com/posts", 443, SplashKit.JsonToString(body), headers);
18 string response = SplashKit.HttpResponseToString(getData);
19 SplashKit.FreeResponse(getData);
20
21 // Output the response
22 Json jsonResponse = SplashKit.JsonFromString(response);
23 SplashKit.WriteLine("Following resource has been created");
24 SplashKit.WriteLine("UserID => " + SplashKit.JsonReadNumberAsInt(jsonResponse, "userId"));
25 SplashKit.WriteLine("ID    => " + SplashKit.JsonReadNumberAsInt(jsonResponse, "id"));
26 SplashKit.WriteLine("Title  => " + SplashKit.JsonReadString(jsonResponse, "title"));
27 SplashKit.WriteLine("Body   => " + SplashKit.JsonReadString(jsonResponse, "body"));
28 SplashKit.WriteLine("=====");
```

PROBLEMS 14 OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE

• breezy breezy .../restful\_API\_Call/ex3 main basic dotnet run build

Following resource has been created  
UserID => 1  
ID => 101  
Title => foo  
Body => test entry  
=====