

```
ros2 pkg create --build-type ament_cmake --license Apache-2.0 cpp_srvcli --dependencies
rclcpp example_interfaces
```

Part	Meaning
ros2 pkg create	Invokes the ROS 2 command-line tool to create a new package.
--build-type ament_cmake	Specifies that the package uses ament_cmake as its build system (for C++).
--license Apache-2.0	Declares that the package is licensed under the Apache 2.0 license. This info goes into the package.xml.
cpp_srvcli	This is the <b>name of your new package</b> .
--dependencies rclcpp example_interfaces	Automatically adds <depend> entries in package.xml and find_package() lines in CMakeLists.txt for rclcpp (ROS 2 C++ client library) and example_interfaces (contains common message and service types like AddTwoInts).

## 서비스 서버 코드

```
#include "rclcpp/rclcpp.hpp"
#include "example_interfaces/srv/add_two_ints.hpp"

#include <memory>

void add(const std::shared_ptr<example_interfaces::srv::AddTwoInts::Request> request,
         std::shared_ptr<example_interfaces::srv::AddTwoInts::Response> response)
{
    response->sum = request->a + request->b;
    RCLCPP_INFO(rclcpp::get_logger("rclcpp"), "Incoming request\na: %ld" " b: %ld",
                request->a, request->b);
    RCLCPP_INFO(rclcpp::get_logger("rclcpp"), "sending back response: [%ld]", (long int)response->sum);
}

int main(int argc, char **argv)
{
    rclcpp::init(argc, argv);

    std::shared_ptr<rclcpp::Node> node = rclcpp::Node::make_shared("add_two_ints_server");

    rclcpp::Service<example_interfaces::srv::AddTwoInts>::SharedPtr service =
        node->create_service<example_interfaces::srv::AddTwoInts>("add_two_ints", &add);

    RCLCPP_INFO(rclcpp::get_logger("rclcpp"), "Ready to add two ints.");

    rclcpp::spin(node);
    rclcpp::shutdown();
}
```

# 서비스 클라이언트 코드

```
#include "rclcpp/rclcpp.hpp"
#include "example_interfaces/srv/add_two_ints.hpp"

#include <chrono>
#include <cstdlib>
#include <memory>

using namespace std::chrono_literals;

int main(int argc, char **argv)
{
    rclcpp::init(argc, argv);

    if (argc != 3) {
        RCLCPP_INFO(rclcpp::get_logger("rclcpp"), "usage: add_two_ints_client X Y");
        return 1;
    }

    std::shared_ptr<rclcpp::Node> node = rclcpp::Node::make_shared("add_two_ints_client");
    rclcpp::Client<example_interfaces::srv::AddTwoInts>::SharedPtr client =
        node->create_client<example_interfaces::srv::AddTwoInts>("add_two_ints");

    auto request = std::make_shared<example_interfaces::srv::AddTwoInts::Request>();
    request->a = atoll(argv[1]);
    request->b = atoll(argv[2]);

    while (!client->wait_for_service(1s)) {
        if (!rclcpp::ok()) {
            RCLCPP_ERROR(rclcpp::get_logger("rclcpp"), "Interrupted while waiting for the service. Exiting.");
            return 0;
        }
        RCLCPP_INFO(rclcpp::get_logger("rclcpp"), "service not available, waiting again...");
    }

    auto result = client->async_send_request(request);
    // Wait for the result.
    if (rclcpp::spin_until_future_complete(node, result) ==
        rclcpp::FutureReturnCode::SUCCESS)
    {
        RCLCPP_INFO(rclcpp::get_logger("rclcpp"), "Sum: %ld", result.get()->sum);
    } else {
        RCLCPP_ERROR(rclcpp::get_logger("rclcpp"), "Failed to call service add_two_ints");
    }

    rclcpp::shutdown();
    return 0;
}
```

executable 코드

```
cmake_minimum_required(VERSION 3.5)
project(cpp_srvcli)
```

```
find_package(ament_cmake REQUIRED)
find_package(rclcpp REQUIRED)
find_package(example_interfaces REQUIRED)
```

```
add_executable(server src/add_two_ints_server.cpp)
ament_target_dependencies(server rclcpp example_interfaces)
```

```
add_executable(client src/add_two_ints_client.cpp)
ament_target_dependencies(client rclcpp example_interfaces)
```

```
install(TARGETS
  server
  client
  DESTINATION lib/${PROJECT_NAME})
```

```
ament_package()
```

## 결과

```
F2_tools
inynxml2_vendor
inynxml_vendor
lsf
lsf_cpp
opic_monitor
racetools
trajectory_msgs
urtlesim
urtlesim_msgs
ncrustify_vendor
nique_identifier_msgs
ndf
ndf_parser_plugin
sualization_msgs
aml_cpp_vendor
std_vendor
saungkim@saungkim-Vivobook-Go-E1504FA-E1504FA:~/ros2_ws$ ros2 run cpp_srvcli ser
[INFO] [1750732581.441443849] [rclcpp]: Ready to add two ints.
[INFO] [1750732632.841840533] [rclcpp]: Incoming request
: 2 b: 3
[INFO] [1750732632.841905450] [rclcpp]: sending back response: [5]
```

```
saungkim@saungkim-Vivobook-Go-E1504FA-E1504FA:~$ 

```
<span></sp
an><span class="gp">$ </span>ros2<span class="w"> </span>run<span class="w"> </s
pan>cpp_srvcli<span class="w"> </span>client<span class="w"> </span><span class=
"m">2</span><span class="w"> </span><span class="m">3</span>
<span class="go">[INFO] [rclcpp]: Sum: 5</span>
</pre>^C
saungkim@saungkim-Vivobook-Go-E1504FA-E1504FA:~$ ros2 run cpp_srvcli client 2 3
Package 'cpp_srvcli' not found
saungkim@saungkim-Vivobook-Go-E1504FA-E1504FA:~$ cd ros2_ws
saungkim@saungkim-Vivobook-Go-E1504FA-E1504FA:~/ros2_ws$ ros2 run cpp_srvcli cli
ent 2 3
Package 'cpp_srvcli' not found
saungkim@saungkim-Vivobook-Go-E1504FA-E1504FA:~/ros2_ws$ source install/setup.ba
sh
saungkim@saungkim-Vivobook-Go-E1504FA-E1504FA:~/ros2_ws$ ros2 run cpp_srvcli cli
ent 2 3
[INFO] [1750732632.842136198] [rclcpp]: Sum: 5
saungkim@saungkim-Vivobook-Go-E1504FA-E1504FA:~/ros2_ws$ □
```


```