# Make an Existing Job Request

Using an *existing* Oracle Job makes your smart contract code more succinct. This page explains how to retrieve the current weather temperature (in Kelvin) for a defined city using an existing Oracle job.

## OpenWeather Consumer

In Make a GET Request, the example contract code declared which URL to use, where to find the data in the response, and how to convert it so that it can be represented on-chain.

In this example, we're using a job found on the Chainlink Market that is pre-configured to perform these tasks. This means that our contract doesn't need to specify additional parameters for various adapters, it only needs the Oracle address and the Job ID. The remaining adapters are configured by the external adapter, in particular weather_cl_ea.

This example uses the Alpha Chain Kovan Oracle, which runs the OpenWeather Data Job.

> Remember to fund your contract with LINK!
>
> Making a job request will fail unless your deployed contract has enough LINK to pay for it. **Learn how to Acquire testnet LINK and Fund your contract**.

Deploy this contract using Remix ↗

What is Remix?

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;

import "@chainlink/contracts/src/v0.8/ChainlinkClient.sol";

/**
 * THIS IS AN EXAMPLE CONTRACT WHICH USES HARDCODED VALUES FOR CLARITY.
 * PLEASE DO NOT USE THIS CODE IN PRODUCTION.
 */
contract OpenWeatherConsumer is ChainlinkClient {
    using Chainlink for Chainlink.Request;

    address private oracle;
    bytes32 private jobId;
    uint256 private fee;

    uint256 public result;
```

```
    /**
     * Network: Kovan
     * Oracle:
     *      Name:           Alpha Chain - Kovan
     *      Listing URL:    https://market.link/nodes/ef076e87-49f4-486b-9878-
     *      Address:        0xAA1DC356dc4B18f30C347798FD5379F3D77ABC5b
     * Job:
     *      Name:           OpenWeather Data
     *      Listing URL:    https://market.link/jobs/e10388e6-1a8a-4ff5-bad6-d
     *      ID:             235f8b1eeb364efc83c26d0bef2d0c01
     *      Fee:            0.1 LINK
     */
    constructor() {
        setPublicChainlinkToken();
        oracle = 0xAA1DC356dc4B18f30C347798FD5379F3D77ABC5b;
        jobId = "235f8b1eeb364efc83c26d0bef2d0c01";
        fee = 0.1 * 10 ** 18; // (Varies by network and job)
    }

    /**
     * Initial request
     */
    function requestWeatherTemperature(string memory _city) public {
        Chainlink.Request memory req = buildChainlinkRequest(jobId, address(th
        req.add("city", _city);
        sendChainlinkRequestTo(oracle, req, fee);
    }

    /**
     * Callback function
     */
    function fulfillWeatherTemperature(bytes32 _requestId, uint256 _result) pu
        result = _result;
    }

    // function withdrawLink() external {} - Implement a withdraw function to
}
```

For more information on finding existing jobs, see Find Existing Jobs.

## What's Next

- › Find Existing Jobs
- › API Reference

› Contract Addresses