

# How do I add a sub-property to a resource?

If you don't know what a sub-property is, then read the [What is a sub-property?](#) article first. **Adding a sub-property to a resource** is very similar to [adding a property](#). Instead of using the *AddProperty* service, we will use the **AddSubProperty** service. They share a nearly identical signature but the *AddSubProperty* service includes one additional argument to identify the parent property. Here is a simple example:

```
objResource      = HTTP_Resource_Services('GetObject')
If Error_Services('NoError') then
    HTTP_Resource_Services('AddSubProperty', objResource, 'mailing', 'address', 'PO Box 1234')
    HTTP_Resource_Services('AddSubProperty', objResource, 'mailing', 'city', 'New Orleans')
    HTTP_Resource_Services('AddSubProperty', objResource, 'mailing', 'county', 'Orleans')
    HTTP_Resource_Services('AddSubProperty', objResource, 'mailing', 'state', 'LA')
    HTTP_Resource_Services('AddSubProperty', objResource, 'mailing', 'zip', '70116')
    HTTP_Resource_Services('AddSubProperty', objResource, 'shipping', 'address', '6649 N Blue Gum St')
    HTTP_Resource_Services('AddSubProperty', objResource, 'shipping', 'city', 'New Orleans')
    HTTP_Resource_Services('AddSubProperty', objResource, 'shipping', 'county', 'Orleans')
    HTTP_Resource_Services('AddSubProperty', objResource, 'shipping', 'state', 'LA')
    HTTP_Resource_Services('AddSubProperty', objResource, 'shipping', 'zip', '70116')
    // Serialize the JSON object.
    jsonResource   = HTTP_Resource_Services('GetSerializedResource', objResource)
end
```

In the above code, the *GetObject* service is called without any arguments. This simply returns a handle to an empty resource object. It doesn't have to be completely empty, but for the sake of a simple demonstration we'll start with an empty resource object and use the *AddSubProperty* service to add our sub-properties. Note, the parent property does not have to be in the resource already. The *AddSubProperty* service will create this for us. We finish by calling the *GetSerializedResource* service so that we get a stringified JSON object that we can inspect or use in our API. Here is what our resource object looks like:

```
{
  "mailing":{
    "address":"PO Box 1234",
    "city":"New Orleans",
    "county":"Orleans",
    "state":"LA",
    "zip":"70116"
  },
  "shipping":{
    "address":"6649 N Blue Gum St",
    "city":"New Orleans",
    "county":"Orleans",
    "state":"LA",
    "zip":"70116"
  }
}
```

Another service that might add a little more convenience is **AddSubProperties**. It's a wrapper around the *AddSubProperty* service that allows the developer to pass in an @FM delimited list of sub-property names and values into a single call. For example:

```
objResource      = HTTP_Resource_Services('GetObject')
If Error_Services('NoError') then
    SubPropertyName = 'address'           : @FM : 'city'           : @FM : 'county' : @FM : 'state' : @FM :
    'zip'
    SubPropertyValue = 'PO Box 1234'      : @FM : 'New Orleans' : @FM : 'Orleans' : @FM : 'LA'       : @FM :
    '70116'
    HTTP_Resource_Services('AddSubProperties', objResource, 'mailing', SubPropertyName, SubPropertyValue)
    SubPropertyName = '6649 N Blue Gum St' : @FM : 'New Orleans' : @FM : 'Orleans' : @FM : 'LA'       : @FM :
    '70116'
    HTTP_Resource_Services('AddSubProperties', objResource, 'shipping', SubPropertyName, SubPropertyValue)
    // Serialize the JSON object.
    jsonResource   = HTTP_Resource_Services('GetSerializedResource', objResource)
end
```

Developers will need to decide if the *AddSubProperties* service provides enough convenience in lieu of multiple *AddSubProperty* calls. One possible beneficial use case is when the values are already stored in an @FM delimited array (such as a database row).