



# THE LOANPRO API

NESTED ENTITIES

[loanprosoftware.simnang.com](https://loanprosoftware.simnang.com)

# API SAMPLE

Fork Expand Nested Entities — <https://plnkr.co/edit/56jFDv>

## **CAUTIONARY NOTE**

The provided API samples store API credentials in the browser

Use the API samples to explore the API and not as your integration

# WHAT ARE ENTITIES?

Entities are a collection of data in OData

Think of them as "objects"

# WHAT ARE NESTED ENTITIES?

Nested entities are groups of data inside another data group

Think of them as objects contained in objects

```
class Object1{  
    int data;  
};  
  
class Object2{  
    Object1 nested;  
};
```

Object1 is nested in Object2

# WHY NESTED ENTITIES?

Nested entities allow describing data by composition

- Loans are composed of escrow, collateral, etc.

They could also be viewed as an entity having or possessing other entities

- Customers have phones, addresses, etc.

# REFERENES TO NESTED ENTITIES

GET requests to an entity will return references to nested entities

GET [https://loanpro.simmang.com/api/public/api/1/odata.svc/Customers\(2\)](https://loanpro.simmang.com/api/public/api/1/odata.svc/Customers(2))

```
{
  "d": {
    "__metadata": {
      "uri": "http://loanpro.simmang.com/api/public/api/1/odata.svc/Customers(id=2)",
      "type": "Entity.Customer"
    },
    "PrimaryAddress": {
      "__deferred": {
        "uri": "Customers(id=2)/PrimaryAddress"
      }
    },
    "MailAddress": {
      "__deferred": {
        "uri": "Customers(id=2)/MailAddress"
      }
    },
    "id": 2,
    "customerType": "customer.type.individual",
    "status": "Active",
    "firstName": "John",
    "lastName": "Doe",
    "birthDate": "1970-01-01"
  }
}
```

Instead of returning Primary address and Mail Address we get referenes to those entities

# GRABBING NESTED ENTITIES

- Expand nested entities by sending the `$expand` URL parameter
- Value is a comma-separated list of nested entities to expand

GET [https://loanpro.simnang.com/api/public/api/1/odata.svc/Customers\(2\)?\\$expand=MailAddress](https://loanpro.simnang.com/api/public/api/1/odata.svc/Customers(2)?$expand=MailAddress)

```
{
  "d": {
    "__metadata": {
      "uri": "http://loanpro.simnang.com/api/public/api/1/odata.svc/Customers(id=2)"
      "type": "Entity.Customer"
    },
    "PrimaryAddress": {
      "__deferred": {
        "uri": "Customers(id=2)/PrimaryAddress"
      }
    },
    "MailAddress": {
      "__metadata": {
        "uri": "http://loanpro.simnang.com/api/public/api/1/odata.svc/Address(id=)"
        "type": "Entity.Address"
      },
      "id": 4,
      "address1": "123 Oak Lane",
      "address2": null,
      "city": "Benicia",
      "state": "geo.state.CA",
      "zip": "94510"
    }
  }
}
```



# COLLECTIONS OF NESTED ENTITIES

- Customers can often have multiple phones
- When expanding, follow the same steps
- Server will just return an array of objects

GET [https://loanpro.simnang.com/api/public/api/1/odata.svc/Customers\(2\)?\\$expand=MailAddress](https://loanpro.simnang.com/api/public/api/1/odata.svc/Customers(2)?$expand=MailAddress)

```
{
  "Phones": {
    "results": [
      {
        "__metadata": {
          "uri": "http://loanpro.simnang.com/api/public/api/1/odata.svc/Phones(id=2)",
          "type": "Entity.Phone"
        },
        "id": 2,
        "entityId": 2,
        "entityType": "Entity.Customer",
        "phone": "5555555555",
        "type": "customer.phoneType.cell",
        "active": 1
      }
    ]
  }
  ...
}
```

# CREATING NESTED ENTITIES

- Send a PUT request to the parent entity
- Tell the parent entity to update by including `__id` and `__update` properties
- Include the information for the nested entity

# CREATING SAMPLE

PUT [https://loanpro.simnang.com/api/public/api/1/odata.svc/Customers\(2\)](https://loanpro.simnang.com/api/public/api/1/odata.svc/Customers(2))

```
{
  "Phones":{
    "results":
      [
        {
          "phone":"7777777777777777",
          "isPrimary":"0",
          "isSecondary":"1",
          "type":"customer.phoneType.cell"
        }
      ]
  },
  "__id":2,
  "__update":true
}
```

# UPDATING NESTED ENTITIES

- Send a PUT request to the parent entity
- Tell the parent entity to update by including **\_\_id** and **\_\_update** properties for the parent entity
- Tell the nested entities to update by including their **\_\_id** and **\_\_update** properties for each nested entity

# UPDATING SAMPLE

PUT [https://loanpro.simnang.com/api/public/api/1/odata.svc/Customers\(2\)](https://loanpro.simnang.com/api/public/api/1/odata.svc/Customers(2))

```
{
  "Phones":{
    "results":
      [
        {
          "phone":"15555555555",
          "isPrimary":"1",
          "isSecondary":"0",
          "type":"customer.phoneType.cell",
          "__id":2,
          "__update":true
        }
      ]
  },
  "MailAddress":{
    "country":"company.country.usa",
    "city":"San Diego",
    "state":"geo.state.NY",
    "address1":"170011 N 250 W",
    "zipcode":"92105",
    "__id":4,
    "__update":true
  }
}
```

# QUESTIONS?

Feel free to ask in our [Forums](#)

Or, email us at [support@simnang.com](mailto:support@simnang.com)