Build with NEXT

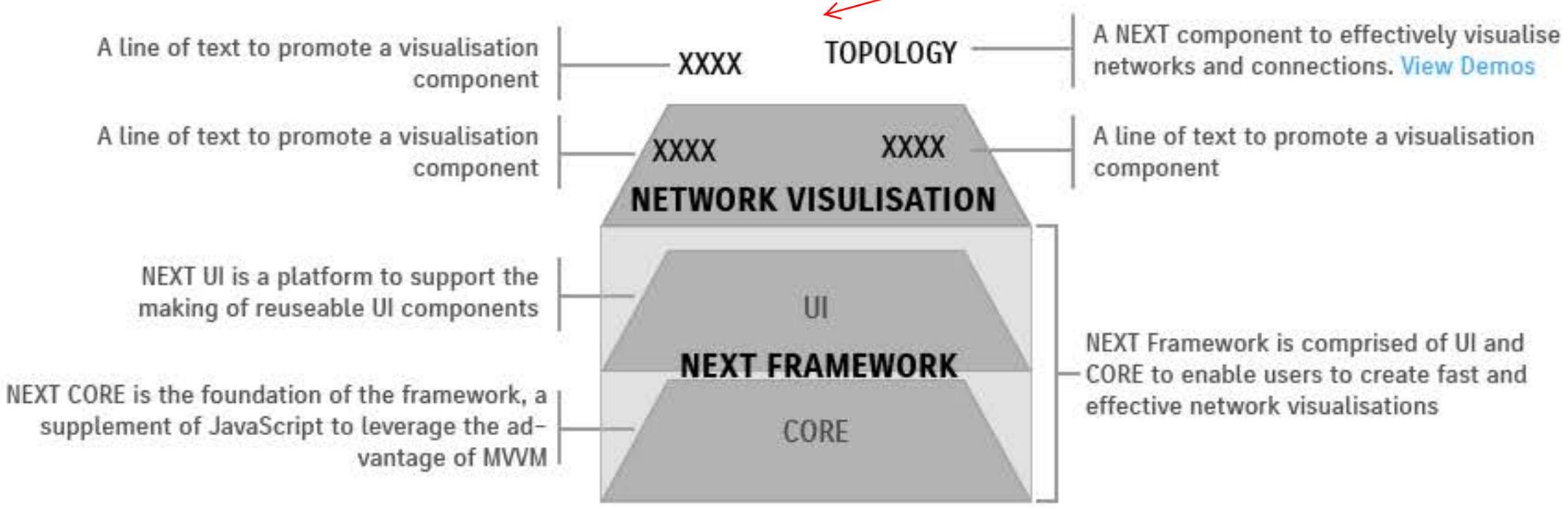
NEXT is a front-end framework that maximise the power of JSON and provides perhaps the only topology component to visualise simple and complex networks.

Download NEXT Learn how to use NEXT

Latest: v1.0.0_2015/03/12_23Mb

Still work in progress

NEXT is a effective framework to build Network Visualisation



Efficient Framework

NEXT is designed and structured base on the concept of data driven. The adoption of MVC and MVVM improves productivity and maintainability.

Effective Topolgoy

It is easy to generate sample topology with NEXT, and by investing minimum effort to follow the tutorials, users could leverage more advanced functionality like aggregation.

High Performance

NEXT framework is constructed with concise codes and accurate structures to achieve high performance.

Active Support

NEXT is under active development and the regular iterations and updates make the framework more stable and more reliable.

Rich APIs

NEXT provides a wide range of well-structured APIs to enable users to efficiently and quickly achieve functionalities, customised topology and build applications on the web.

Compatible with rich UI styles

NEXT is compatible with popular 3rd party CSS frameworks such as Bootstraps and this makes creating beautiful web apps effortless.

NEXT is a practical tool for real-world projects

"..NeXt has been a cornerstone of our SDN apps development efforts, first in the WAN Automation Engine (WAE) and more recently for those apps running over OpenDaylight .."

Chris Metz
Developer Lead, WAE Project

"...Next allowed us to deliver an innovative feature by giving customers the ability to see their network topologies, and it is a much faster way to understand networks and spot potential configuration errors than reading through individual device configurations..."

Simon Knight
Developer, VIRL Project

"..The NeXt framework enables us to quickly and easily provide a modern and visually appealing HTML5 Web Application with minimal design knowledge required. We appreciate the UX thought and design that has gone into the framework..."

Simon Knight
Developer, VIRL Project

View Showcase

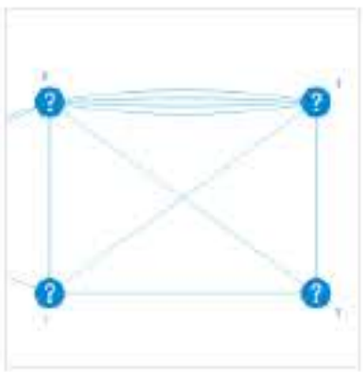
Example to show what Topology could do

Picture or GIF to show what the functionality is and click to bring out the popup panel for the function

Example

Topology

Basic



Base

Auto Layout

Highlight Link and Node

Topology Icons

Set node's lable and icon

Disable node or link

Topology theme

Customize Node and Link style

NodeSet

NodeSet

Hierarchy NodeSet

Hierarchy NodeSet

Group

Add Group

Layer

Define customize scene

Interaction with Layers

Topology theme

Customize Node and Link style

NodeSet

NodeSet

Hierarchy NodeSet

Scene

Define customize scene

Path

Add path

Map

Add multiple path

Add traffic path

US Map

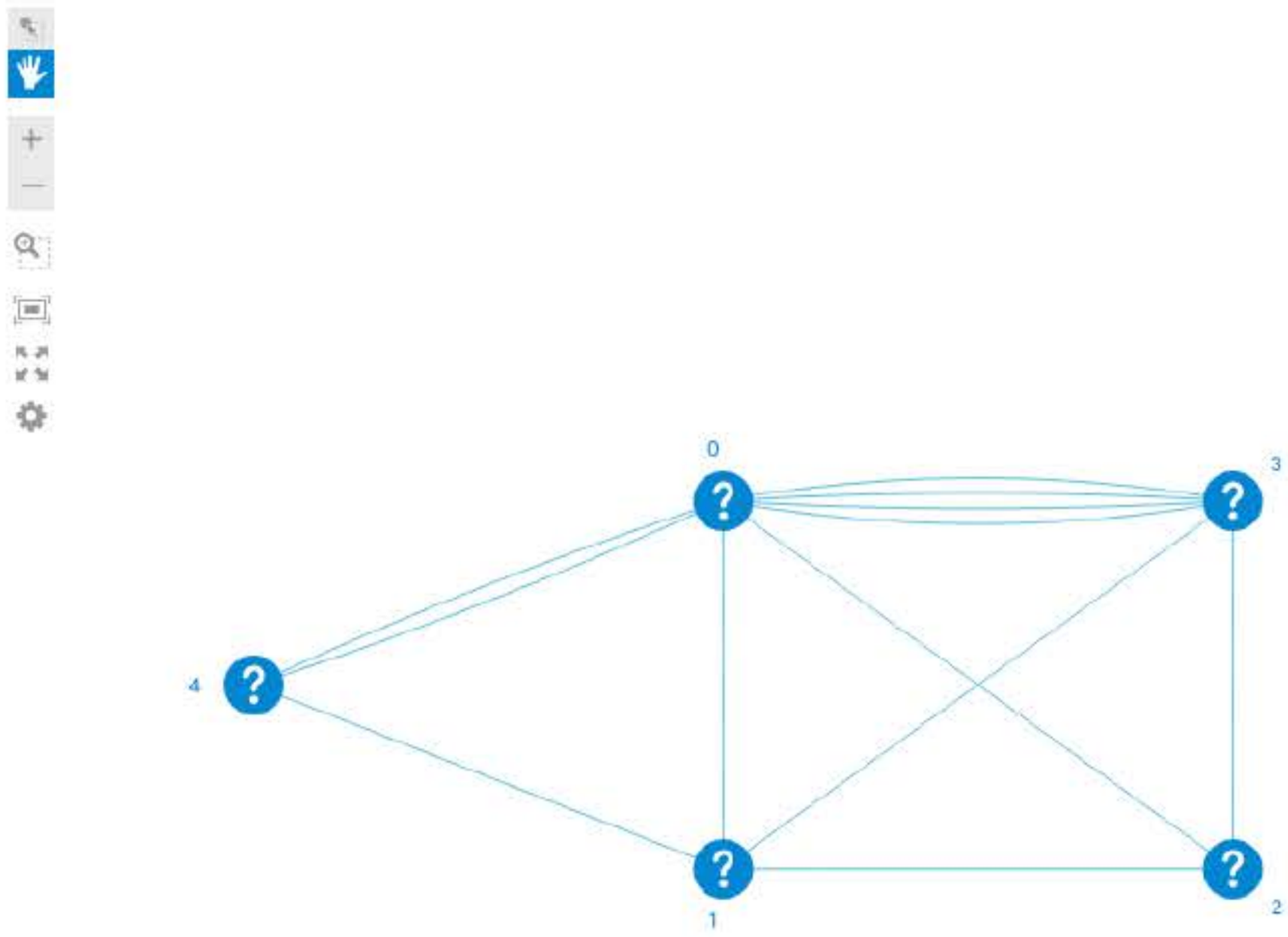
World Map

Example

Topology Basic_Base

Herer are a few line of text to inform user what this exmaple is all about.Herer would a few line of text to inform user what this exmaple is all about.Herer would a few line of text to inform user what this exmaple is all about.Herer would a few line of text to inform user what this exmaple is all about.

Popup panel to show a specific function



Dynamic panel allow users to engage with the example if interested

HTML CSS JavaScript Edit in JSFiddle

```
<div class="media">
  <div class="media-left">
    <a href="#">
      <img class="media-object" src="..." alt="...">
    </a>
  </div>
  <div class="media-body">
    <h4 class="media-heading">Media heading</h4>
    ...
```

Layer

Define customize scene

Interaction with Layers

Topology theme

Customize Node and Link style

NodeSet

NodeSet

Hierarchy NodeSet

Scene

Define customize scene

Path

Add path

Map

Add multiple path

Add traffic path

US Map

World Map

Showcase

Suggest for Showcase

Have you seen or used NEXT for a project?

Let us know, and we could add your project for Showcase.

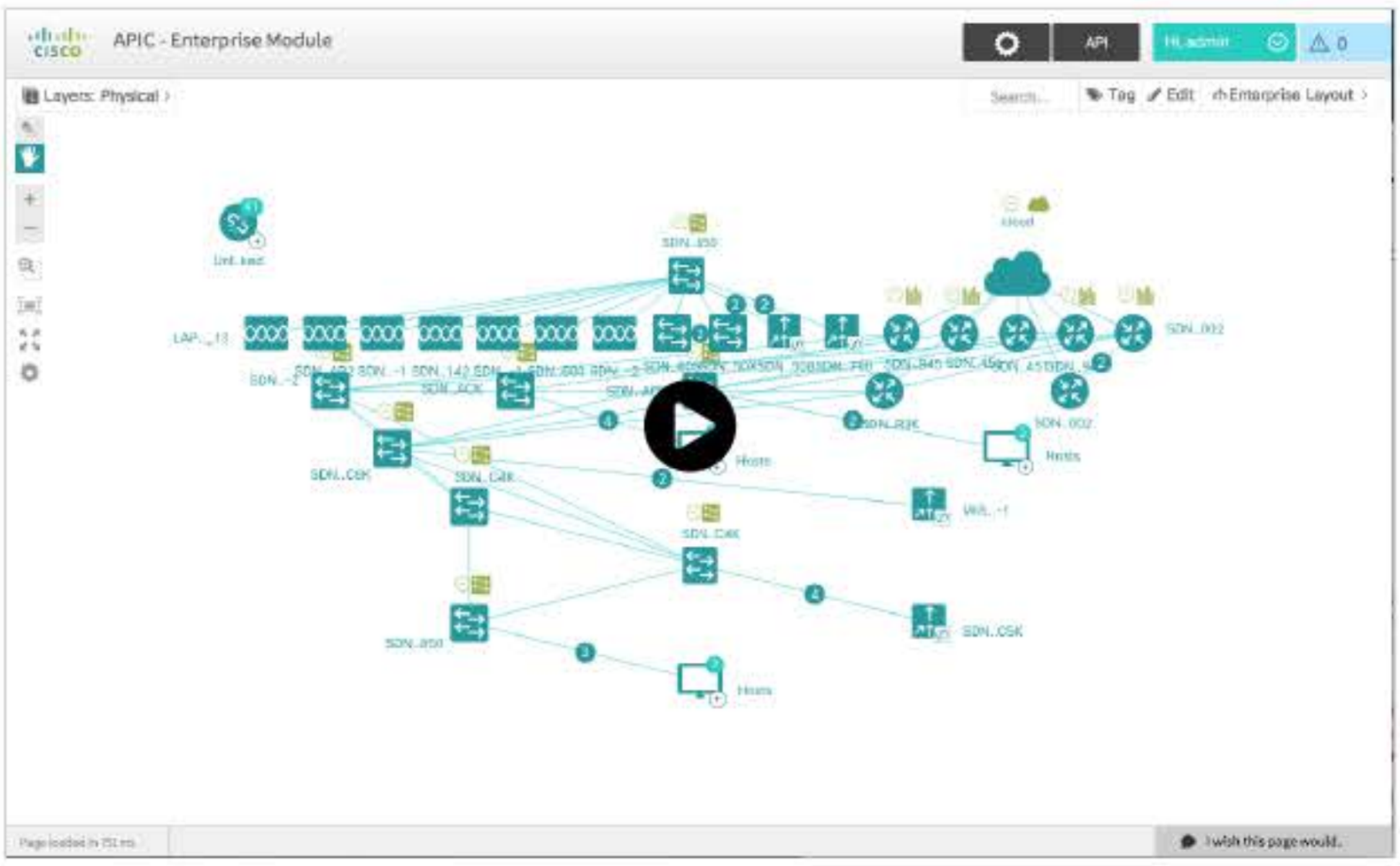
Submit suggestion

NEXT has been used in the development and making of many real-world projects. Specially the topology component has been a very popular and valuable tool for developers and teams across CISCO to create and build web based network topology. Here are some projects that has been used NEXT framework.

The showcase panel to show how next is used in real projects

APIC-EM

APIC-EM is an open network modeling & reporting toolkit that provides Fast & Flexible, object-based network modeling for networks. NEXT was used by the project team to create network topolgies.

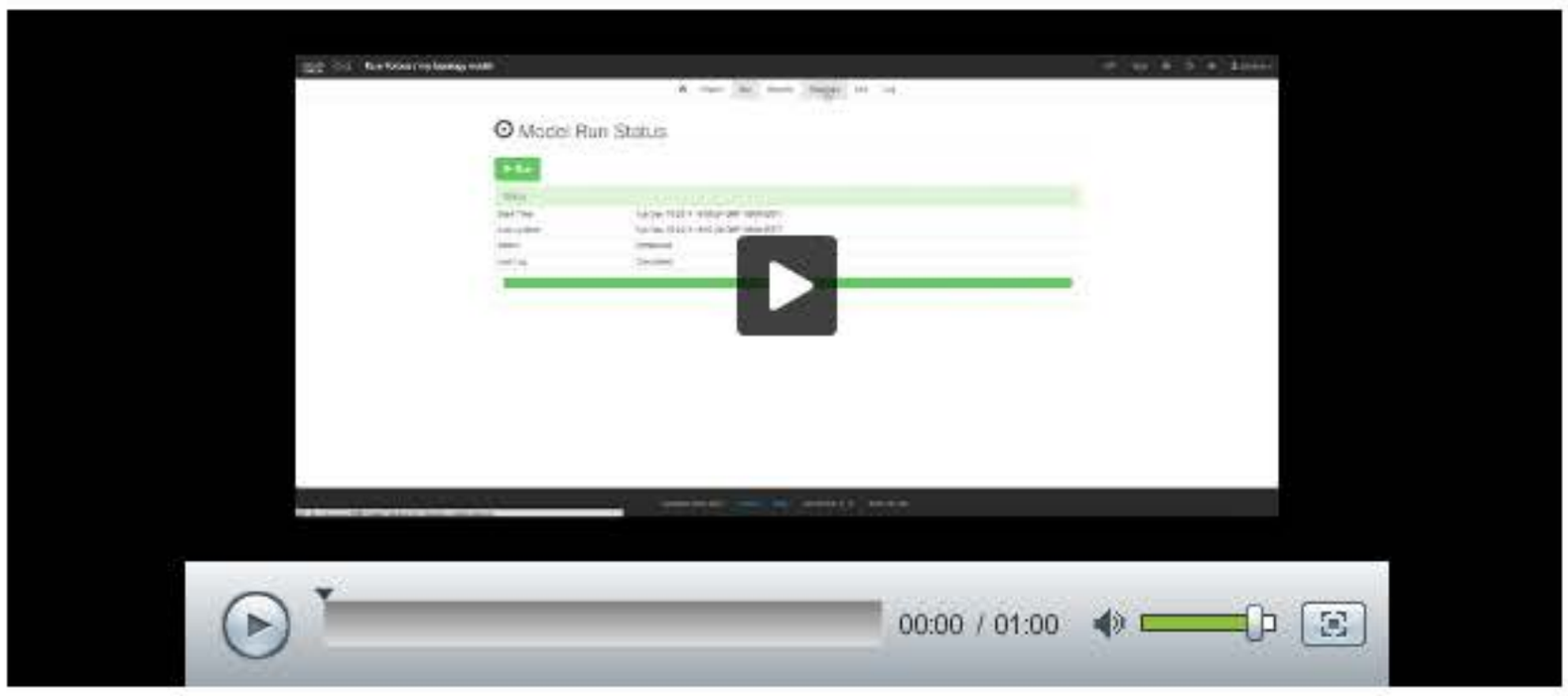


ODL APP

This app is an open network modeling & reporting toolkit that provides Fast & Flexible, object-based network modeling for networks. NEXT was used by the project team to create network topolgies.

LOKI_http://iwe.cisco.com/web/loki/home

This app is an open network modeling & reporting toolkit that provides Fast & Flexible, object-based network modeling for networks. NEXT was used by the project team to create network topolgies.



Tutorial uses the updated work that Abu and Ronak have done.

Define a class

To define a class in NeXT framework, use the following syntax to declare a class.

Syntax : nx.define(className,classBody)

Example : nx.define(Test,{})

Explanation : The above example defines the Test class with the empty body. User can initialize the instance of the Test class using the following code :

Usage : var obj = new Test()

Sample code on JSFiddle. Give it a Try!

Define a class with the class body

Class body consists of mainly two parts : properties and methods. Properties are similar to the instance variables and are specific to the instance of the class. They are basically used to record the state of an instance. Methods are used to alter the values of the properties or perform some logical operation.

Syntax :

```
nx.define(#className#,{
    properties : {
        #propertyName1# : #propertyValue1# ,
        #propertyName2# : #propertyValue2#
            },
    methods : {
        #method1#: function(params) {
                },
        #method2#: function(params) {
                }
    }
        })
```

Example :

```
nx.define('Demo', {
            properties: {
                propertyOne : 1,
                propertyTwo : 2
            }
})
```

Explanation : This defines the class Test with two properties(propertyOne with value 1 and propertyTwo with value 2). By default NeXT implicitely provides the getters and setters for the properties. So each property(in this case propertyOne and propertyTwo) has their getters and setters. In the usage section below, when the d.propertyOne() line executes, it automatically invokes the getter of the propertyOne which is provided by NeXT framework. Similar is the concept with d.propertyTwo(). But when d.propertyOne(9) executes, it calls the setter of the propertyOne and sets the value of propertyOne to 9. Similar is the concept with d.propertyTwo('Name').

Usage :

```
var d = new Demo()
console.log(d.propertyOne()) // Output : 1
console.log(d.propertyTwo()) // Output : 2
d.propertyOne(9)
d.propertyTwo('Name')
console.log(d.propertyOne()) // Output : 9
console.log(d.propertyTwo()) // Output : Name
```

Sample code on JSFiddle. Give it a Try!

Define a class with the class body

Class body consists of mainly two parts : properties and methods. Properties are similar to the instance variables and are specific to the instance of the class. They are basically used to record the state of an instance. Methods are used to alter the values of the properties or perform some logical operation.

Syntax :

```
nx.define(#className#,{
    properties : {
        #propertyName1# : #propertyValue1# ,
        #propertyName2# : #propertyValue2#
            },
    methods : {
        #method1#: function(params) {
                },
        #method2#: function(params) {
                }
    }
        })
```

Example :

```
nx.define('Demo', {
            properties: {
                propertyOne : 1,
                propertyTwo : 2
            }
})
```

Explanation : This defines the class Test with two properties(propertyOne with value 1 and propertyTwo with value 2). By default NeXT implicitely provides the getters and setters for the properties. So each property(in this case propertyOne and propertyTwo) has their getters and setters. In the usage section below, when the d.propertyOne() line executes, it automatically invokes the getter of the propertyOne which is provided by NeXT framework. Similar is the concept with d.propertyTwo(). But when d.propertyOne(9) executes, it calls the setter of the propertyOne and sets the value of propertyOne to 9. Similar is the concept with d.propertyTwo('Name').

Usage :

```
var d = new Demo()
console.log(d.propertyOne()) // Output : 1
console.log(d.propertyTwo()) // Output : 2
d.propertyOne(9)
d.propertyTwo('Name')
console.log(d.propertyOne()) // Output : 9
console.log(d.propertyTwo()) // Output : Name
```

Sample code on JSFiddle. Give it a Try!

APIs

Classes | Modules

Type to filter APIs

nx.graphic.Topology Class

Show: ☑ Inherited ☐ Protected ☐ Private ☐ Deprecated

Uses nx.graphic.Topology.Config, nx.graphic.Topology.Projection, nx.graphic.Topology.Graph, nx.graphic.Topology.Event, nx.graphic.Topology.StageMixin, nx.graphic.Topology.NodeMixin, nx.graphic.Topology.LinkMixin, nx.graphic.Topology.LayerMixin, nx.graphic.Topology.TooltipMixin, nx.graphic.Topology.SceneMixin
Extends nx.ui.Component
Defined in: dest/next-topology/js/next-topology.js:10141
Module: nx.graphic.Topology

Topology base class

```
var topologyData = {
    nodes: [
        {"id": 0, "x": 410, "y": 100, "name": "12K-1"},
        {"id": 1, "x": 410, "y": 280, "name": "12K-2"},
        {"id": 2, "x": 660, "y": 280, "name": "0f-9k-03"},
        {"id": 3, "x": 660, "y": 100, "name": "0f-9k-02"},
        {"id": 4, "x": 180, "y": 190, "name": "0f-9k-01"}
    ],
    links: [
        {"source": 0, "target": 1},
        {"source": 1, "target": 2},
        {"source": 1, "target": 3},
        {"source": 4, "target": 1},
        {"source": 2, "target": 3},
        {"source": 2, "target": 0},
        {"source": 3, "target": 0},
        {"source": 3, "target": 0},
        {"source": 3, "target": 0},
        {"source": 0, "target": 4},
        {"source": 0, "target": 4},
        {"source": 0, "target": 3}
    ]
};
nx.define('MyTopology', nx.ui.Component, {
    view: {
        content: {
            type: 'nx.graphic.Topology',
            props: {
                width: 800,
                height: 800,
                nodeConfig: {
                    label: 'model.id'
                },
                showIcon: true,
                data: topologyData
            }
        }
    }
```

Use YUI for API section

Get NEXT

Feedback

NEXT framework is currently under active development, and the NEXT team constantly invests efforts to improving the functionality and stability of the framework, if you have feedback about everything, feel free to let us know, thank you for using NEXT.

Submit

Download NEXT now, and enjoy building web apps with this fantastic framework.

Download NEXT

The Latest Release: v1.0.0
2015/03/12_23Mb

0.7.3 Release note

Topology

- Support Firefox (30+)
- Added extend link & linkSet support

Known Issues

- [Core] Temporary don't support IE

0.7.3 Release note

Topology

- Support Firefox (30+)
- Added extend link & linkSet support

Known Issues

- [Core] Temporary don't support IE

0.7.3 Release note

Topology

- Support Firefox (30+)
- Added extend link & linkSet support

Known Issues

- [Core] Temporary don't support IE

0.7.3 Release note

Topology

- Support Firefox (30+)
- Added extend link & linkSet support

Known Issues

- [Core] Temporary don't support IE

0.7.3 Release note

Topology

- Support Firefox (30+)
- Added extend link & linkSet support

Known Issues

- [Core] Temporary don't support IE

0.7.3 Release note

Topology

- Support Firefox (30+)
- Added extend link & linkSet support

Known Issues

- [Core] Temporary don't support IE