HTTP-based Service

A simple HTTP POST request is used to send the input graph and retrieve its the layouted version. You can just perform your own request to the correct URL and pass all required information. However, as we might change internal APIs we recommend you to use the (very small) wrapper implementations we provide for a specific language. We do not provide a wrapper for each language, feel free to write your own and contact us so we can add it to the list below.



You might want to have a look at the *Live* section of our exemplary server at http://layout.rtsys.informatik.uni-kiel.de:9444 to play around.

Language Bindings

Each zip archive contains the wrapper class you can use to call our service as well as a small example illustrating the usage pattern for the specific language.

- JavaScript
- Java
- C#
- Python

Examples

The graph is passed in serialized form of either graph format. In this example we use the JSON format (for JavaScript also native JSON can be passed).

graph

```
"{id:\"root\",children:[{id:\"n1\",labels:[{text:\"n1\"}],width:100,height:100},"
                       + "{id:\"n2\",labels:[{text:\"n2\"}],width:100,height:50,children:[{id:\"n3\","
                       + "labels:[{text:\"n3\"}],width:20,height:20},{id:\"n4\",labels:[{text:\"n4\"}],width:
20,"
                       + "height:20}],edges:[{id:\"e4\",source:\"n3\",target:\"n4\"}]}],"
                       + "edges:[{id:\"e1\",source:\"n1\",target:\"n2\"}]}";
```

options

```
spacing: 100
algorithm: "de.cau.cs.kieler.klay.layered"
edgeRouting: ORTHOGONAL
```

JavaScript

```
var graph = "[graph]";
var options = { opt1: val1 };
$.kielerLayout({graph: graph, options: options,
    iFormat: 'org.json', oFormat: 'org.json',
    success : function (data) {
        console.log(data);
    }
});
```

Java

```
String graph = "[graph]";
Map<String, Object> opts = new HashMap<String, Object>();
String layouted = KIELERLayout.layout(server, "org.json", "org.json", options, graph);
```

C#

```
String graph = "[graph]";
Dictionary<String, Object> options = new Dictionary<String, Object>();
String layouted = KIELER.KIELERLayout.layout(server, "org.json", "org.json", options, graph);
```

Available Layout Algorithms and Options

The available layout algorithms along with the available options and some further information can be requested under the following URL. A JSON object is returned, the structure and contained data equals the *ServiceData* class in the diagram shown below. Furthermore, several algorithms offer a *preview image* illustration a possible result. You can retrieve this image as a png using the second URL. Optionally the image can be retrieved base64 encoded.

```
http://layout.rtsys.informatik.uni-kiel.de:9444/layout/serviceData

http://layout.rtsys.informatik.uni-kiel.de:9444/layout/previewImage/[imgPath][?base64=true]
```

