# Introduction

It is possible to define the underlying data model without having to write java code. To do this, you must use an XML descriptor which describes the DataNodes and their relationships. An skeleton XML descriptor has the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<zkib-model>
   <datanode name="my-data">
      <finder name="title" type="title">
      </finder>
      <finder name="country" type="country">
         ...
      </finder>
   </datanode>
   <datanode name="title"/>
   <datanode name="country">
      <finder name="city" type="city">
         ...
      </finder>
   </datanode>

   <datanode name="city">
         ...
   </datanode>
</zkib-model>
```

Within the XML tag whose root is always zk-ib, you can specify one or more DataNodes. Each DataNode has a unique name. Within each DataNode, you can define multiple finders. Each finder specifies a name and a type. The name will be used to build xpaths, whlie the type identifies the type of DataNode this xpaths refers to.

Within each finder you can define multiple search handlers. They will be responsible for retrieving data from persistent storage, just like the find method on the finder interface. Additionally, you can define one or new instance handlers. They will be responsible for creating new business objects on user request.

Finally, each DataNode can have many persistence handlers. They will act just lik the doInsert, doUpdate and doDelete methods on DataNode class. Each type of handler can be executed

conditionally, depending on expressions to be evaluated at run time. These expressions can use the following predefined variables:

Additionally, EL expressions may refer to all variables defined within the DataSource. Those variables are accessed via JXPathContext.getVariables() method. To use of this type of data models, simply create a datamodel component on the ZUL page and assign the src attribute the path to the XML descriptor. The path can be a web component or a class path resource.

| Variable | Value |
| --- | --- |
| self | Current DataNode |
| instance | Business object wrapped into current DataNode |
| parent | Parent DataNode |
| parent.instance | Business object wrapped into parent DataNode |
| datasource | DataSoruce the current DataNode belongs to |

Revision #2
Created 1 June 2021 10:39:18
Updated 1 June 2021 10:40:23