

# Property Expression Language

Property Expression Language is used in the Property Models to access property values of object wrapped by model. Its syntax is very similar to the [Object Graph Navigation Language](#) (OGNL). (Wicket actually used to utilize the OGNL implementation until version 1.1.x. However, later it has been replaced by custom, better-performing, implementation of expression parser).

Current implementation of the property expression parser is represented by the class `wicket.util.lang.PropertyResolver`. This class supports the following expressions:

- `"property"`: This expression can be used to access a bean property with get and set method for a property named `property`. Alternatively, if a object wrapped by model is an instance of Map and there is no get property for given name, the expression will be used as a key of the given map.
- `"property1.property2"`: Both properties are looked up in the same way as mentioned above. If the `property1` evaluates to null and there exists a setter with name `property1` on the model object and the class representing the class representing `property1` has a default constructor then a new instance will be constructed and assigned on the model object via the `property1` and the `property2` will be set on this new object.
- `"property.index"`: If the property is a List or Array then the second property can be used as a index on that list/array like this: `'mylist.0'`. This kind of expression will be mapped on the `getProperty(index)` or `setProperty(index, value)` methods. If the object represents a List and the index is greater than its size of this list, the list will be expanded.

Index or map properties can be alternatively written as: `"property[index]"` or `"property[key]"`.

For example, if we have a following class:

## Person.java

```
public static class Person {
    private String name;

    private Person parent;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Person getParent() {
        return parent;
    }

    public void setParent(Person parent) {
        this.parent = parent;
    }
}
```

Then we can use the following expressions:

- `name` can be used to access getters and setter for the `name` property
- `parent.name` can be used to retrieve or set the name of a parent. If there is no parent set (i. e. it is null), a new instance of `Person` is created and set on the model object. Thereafter a `name` property is get/set on this new instance.

### From the mailing list

We started out with OGNL in the past but:

- 1) OGNL at one point took about 30% processor time of the whole request. We simplified and optimized and wrote OGNL out.
- 2) We feel it's not the recommended way of programming to rely on property expressions beyond simple navigations
- 3) By overriding `wicket.model.AbstractPropertyModel#onGetObject(wicket.Component)` and `AbstractPropertyModel#onSetObject(Component, Object)` users can provide their own resolving if [necessary].

Eelco Hillenius