

## PHPCR - Task #3845

### Iterator behaviour - Java vs PHP

2009-07-07 22:39 - Karsten Dambekalns

<b>Status:</b>	Resolved	<b>Start date:</b>	2009-07-07
<b>Priority:</b>	Should have	<b>Due date:</b>	
<b>Assignee:</b>	Karsten Dambekalns	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Sprint:</b>			

#### Description

From David Buchmann:

iterators are a bit different between java and php. in java, next() advances the iterator one step and then returns the current element. advancing behind the end throws an exception. usually, you do

```
while(iterator.hasNext()) {
    item = iterator.next();
    ...
}
```

in php, next() only advances the pointer to the next element, its return type is void. getting the element is done using current(), it is an error to call current() if you are not at an element. to check if next() moved you past the end, there is the valid() method. iterators can be used in the foreach construct, that relies on this semantics. this typically looks like

```
foreach($iterator as $item) {
    ...
}
```

PHPCR\_IteratorInterface extends the php Iterator. in my opinion, it should follow the php iterator semantics, not the java ones. then it could be used with the foreach construct and it would be less confusing.

we should however define the nextNode() and the other nextXY to follow the java semantics, as they are actually discribed in the jcr specifications. so they would internally do

```
nextNode() {
    $this->next();
    return $this->current();
}
```

and, while criticizing: do we need append method? if the base php iterator does not have it, i do not really see why we should have it here. those iterators are typically used in a search result. we can not add anything to the jcr workspace there, as we do not know where to put it. at least, it should be optional and allowed to throw something like a NotImplemented exception.

#### Associated revisions

##### Revision d4ee2089 - 2009-10-28 13:17 - Karsten Dambekalns

[API] PHPCR: Removed IteratorInterface and thus hasNext(), remove() and append(). Use valid() instead of hasNext(). next() is now supposed to behave like in the PHP iterator interface (i.e. return void). Resolves #3845.

##### Revision 3359 - 2009-10-28 13:17 - Karsten Dambekalns

[API] PHPCR: Removed IteratorInterface and thus hasNext(), remove() and append(). Use valid() instead of hasNext(). next() is now supposed to behave like in the PHP iterator interface (i.e. return void). Resolves #3845.

##### Revision 3360 - 2009-10-28 13:18 - Karsten Dambekalns

[API] TYPO3CR: Removed hasNext(), remove() and append() from Rangeliterator. Use valid() instead of hasNext(). next() now returns void, use current() to fetch the current item. Relates to #3845.

## History

---

### #1 - 2009-07-07 23:17 - Karsten Dambekalns

No, from <http://php.net/next>: "next() behaves like current(), with one difference. It advances the internal array pointer one place forward before returning the element value. That means it returns the next array value and advances the internal array pointer by one."

The append() is not needed and can probably be dumped. I'll look up why we put that in...

### #2 - 2009-07-10 10:35 - Karsten Dambekalns

I mixed up next() (for arrays) and Iterator::next(). The latter is indeed not supposed to return anything. So we should follow suit and make our iterators PHPish in that respect.

As for append() I could not find anything aside from the comment in the sources: *append() is something we thought would be nice...* So we should drop it again.

### #3 - 2009-10-28 14:00 - Karsten Dambekalns

- Status changed from Accepted to Resolved

- % Done changed from 0 to 100

Applied in changeset r3359.